

Jornadas de Atualização em Informática 2016

Organização

José Carlos Maldonado

José Viterbo

Marcio Eduardo Delamaro

Sabrina Marczak



Jornadas de Atualização em Informática 2016

Editores

José Carlos Maldonado
José Viterbo
Marcio Eduardo Delamaro
Sabrina Marczak

Sociedade Brasileira de Computação (SBC)



Dados Internacionais de Catalogação na Publicação (CIP)

C749 Congresso da Sociedade Brasileira de Computação (36. : 4 – 7 jul. 2016 : Porto Alegre)
Jornada de Atualização em Informática 2016 [recurso eletrônico] / organização: José Carlos Maldonado ... [et al.].
Dados eletrônicos. – Porto Alegre: Sociedade Brasileira de Computação, 2016.
268 p. : il. : PDF ; 30.2MB

Modo de acesso: World Wide Web.
Inclui bibliografia
ISBN 978-85-7669-326-0 (e-book)

1. Computação – Brasil – Congressos. 2. Informática. I. Maldonado, José Carlos. II. Viterbo, José. III. Delamaro, Marcio Eduardo. IV. Marczak, Sabrina. V. Sociedade Brasileira de Computação. VI. Título.

CDU 004(063)

Ficha catalográfica elaborada por Jéssica Paola Macedo Müller – CRB-10/2662

Biblioteca Digital da SBC – SBC OpenLib

Índices para catálogo sistemático:

1. Ciência e tecnologia informáticas : Computação : Processamento de dados – Publicação de conferências, congressos, simpósios e etc. ... 004(063)

Editores

José Carlos Maldonado (Universidade de São Paulo)

José Viterbo (Universidade Federal Fluminense)

Marcio Eduardo Delamaro (Universidade de São Paulo)

Sabrina Marczak (Pontifícia Universidade Católica do Rio Grande do Sul)

Título - Jornadas de Atualização em Informática 2016

Local - Porto Alegre/RS

Ano de Publicação - 2016

Edição - 1^a

Editora - Sociedade Brasileira de Computação - SBC

Organizadores - José Carlos Maldonado (ICMC-USP), José Viterbo (UFF),
Marcio Eduardo Delamaro (ICMC-USP) e Sabrina Marczak (PUCRS)

ISBN: 978-85-7669-326-0



© Sociedade Brasileira de Computação, SBC

Organização

Coordenação Geral

José Carlos Maldonado (ICMC-USP)

José Viterbo Filho (UFF)

Marcio Eduardo Delamaro (ICMC-USP)

Coordenação Local

Sabrina Marczak (PUCRS)

Comitê de Programa

Adenilso Simão (ICMC-USP)

Alessandro Garcia (PUC-Rio)

Altigran Soares da Silva (UFAM)

Bernadette Loscio (UFPE)

Daltro Nunes (UFRGS)

Fátima Nunes (EACH-USP)

Guilherme Travassos (COPPE/UFRJ)

Marco Aurélio Gerosa (IME-USP)

Mariza Bigonha (UFMG)

Mirella Moro (UFMG)

Nelson Fonseca (UNICAMP)

Raimundo Barreto (UFAM)

Rodolfo Azevedo (UNICAMP)

Routo Terada (IME-USP)

Sandra Fabbri (UFSCar)

Thais Vasconcelos Batista (UFRN)

Índice

Prefácio.....	7
Apresentação dos Autores	9
Cidades Inteligentes: Tecnologias, Aplicações, Iniciativas e Desafios (<i>Fabio Kon e Eduardo Zambom</i>).....	13
Computação Urbana: Técnicas para o Estudo de Sociedades com Redes de Sensoriamento Participativo (<i>Thiago H. Silva e Antonio A. F. Loureiro</i>)	61
Teoria da Computação: uma Introdução à Complexidade e à Lógica Computacional (<i>Celina de Figueiredo e Luis Lamb</i>).....	119
Agentes Inteligentes Conversacionais: Conceitos Básicos e Desenvolvimento (<i>Patrícia Tedesco e Flavia Barros</i>)	169
Interfaces Cérebro-Computador (<i>João Luis Garcia Rosa</i>).....	219

Prefácio

A Jornada de Atualização em Informática (JAI), promovida pela Sociedade Brasileira de Computação, é um dos mais importantes eventos acadêmicos de atualização científica e tecnológica da comunidade de Computação do Brasil e compreende trabalhos de pesquisadores sêniores da nossa comunidade, oferecendo uma oportunidade única para acadêmicos e profissionais atualizarem-se em temas diversos, interagindo com líderes das mais diversas áreas de pesquisa no Brasil.

Neste ano de 2016, temos a 35ª edição da Jornada de Atualização de Informática. A JAI é tradicionalmente oferecida no escopo do Congresso da Sociedade Brasileira de Computação (CSBC). O CSBC ocorre na cidade Porto Alegre, no Estado do Rio Grande do Sul, de 4 a 7 de julho, na PUCRS.

Primeiramente, gostaríamos de agradecer imensamente aos autores que submeteram as propostas para este ano de 2016 e aos membros do Comitê Científico, os quais – autores e avaliadores – contribuíram para a seleção de um conjunto de textos que tratam de temas avançados e de relevância técnico-científica, com amplas possibilidades de contribuição para soluções de alto impacto social. Agradecemos também a atenção e o apoio da organização geral do CSBC 2016 e da Diretoria da SBC que propiciaram todas as condições para a realização desta edição da JAI.

É importante destacar que a JAI favorece a participação de pesquisadores renomados, nacional e internacionalmente, com a colaboração de talentos emergentes, propiciando um ambiente rico para a evolução e disseminação do conhecimento no âmbito do CSBC. Gostaríamos inclusive de sugerir fortemente a oferta desses cursos das JAIs em outros eventos científicos, oportunizando a jovens e profissionais nas diversas regiões do Brasil, e eventualmente na América Latina, o contato com esses excelentes autores e pesquisadores. Salientamos que, por decisão do Conselho da SBC, os textos da JAI 2016 passarão a ser de acesso aberto a partir de 6 meses de sua edição.

Nesta edição da JAI, são apresentados cinco textos, sendo dois convidados da 34ª edição. Os cinco textos abordam um amplo espectro de temas relevantes para a computação e suas aplicações e contribuem certamente para soluções complexas e multidisciplinares relacionadas aos Grandes Desafios da Computação.

Os textos cobrem desde aspectos básicos de teoria de computação, passando por aspectos de comunicação usuário computador a aspectos de integração de soluções computacionais na perspectiva de sistemas de sistemas.

O texto “Teoria da Computação: uma Introdução à Complexidade e à Lógica Computacional” apresenta uma introdução às duas subáreas fundamentais da teoria da computação: complexidade e lógica computacional. Segundo os autores, essas duas áreas

contribuem significativamente para o desenvolvimento da Ciência da Computação, não somente do ponto de vista fundamental, mas também sob uma perspectiva de aplicações.

Por outro lado, os textos “Cidades Inteligentes: Tecnologias, Aplicações, Iniciativas e Desafios” e “Computação Urbana: Técnicas para o Estudo de Sociedades com Redes de Sensoriamento Participativo” apresentam um cenário decorrente do crescimento da população urbana e como os seus efeitos negativos poderiam ser mitigados pela Tecnologia da Informação e Comunicação (TIC), no sentido de tornar as cidades mais inteligentes e poder ajudar a melhorar os serviços urbanos, as questões relacionadas com a dinâmica de cidades e o comportamento social urbano, conseqüentemente, aprimorando a qualidade de vida de seus cidadãos.

Os outros dois textos trazem contribuições significativas que abrem horizontes de desenvolvimento de soluções e aplicações que certamente impactarão a dinâmica das relações sociais e econômicas. O texto “Agentes Conversacionais Incorporados: Desenvolvimento e Aplicações” apresenta um panorama sobre Agentes Inteligentes Conversacionais que visam a facilitar a interação com o usuário a ponto de que o interlocutor tenha a ilusão de estar conversando com outro ser humano. O texto “Interfaces Cérebro-Computador” traz uma perspectiva de indivíduos conectarem-se a dispositivos externos usando sinais cerebrais.

O conjunto de textos apresentado nesta edição evidencia a relevância de investimentos para o avanço da computação, e de TICs de uma forma geral, para a concepção e implementação de sistemas (sistemas de sistemas) computacionais que promovam a sustentabilidade, a inclusão e o acesso a uma ampla gama de cidadãos almejando uma sociedade mais justa e equilibrada.

Desejamos a todos os participantes do CSBC uma excelente Jornada de Atualização de Informática e reiteramos nossos imensos agradecimentos à comunidade. Esperamos que a compilação desses ricos textos contribua para motivar soluções elegantes e comprometidas com a evolução da riqueza social. Inovar e empreender é preciso!

Coordenação Geral

José Carlos Maldonado (ICMC-USP)

José Viterbo Filho (UFF)

Marcio Eduardo Delamaro (ICMC-USP)

Coordenação Local

Sabrina Marczak (PUCRS)

Apresentação dos Autores

Antonio Alfredo Ferreira Loureiro é Bolsista de Produtividade em Pesquisa do CNPq Nível 1A. Possui graduação em Ciência da Computação pela Universidade Federal de Minas Gerais (1983), mestrado em Ciência da Computação pela Universidade Federal de Minas Gerais (1987) e doutorado em Ciência da Computação pela University of British Columbia, Canadá (1995). Atualmente é Professor Titular da Universidade Federal de Minas Gerais. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas Distribuídos, atuando principalmente nos seguintes temas: algoritmos distribuídos, computação móvel/ubíqua, computação urbana, comunicação sem fio, gerenciamento de redes, redes de computadores, redes de sensores sem fio.

Celina Miraglia Herrera de Figueiredo obteve bacharelado (1982) e mestrado (1984) em Matemática na PUC-Rio, mestrado (1987) em Matemática no UMIST (UK), doutorado (1991) em Engenharia de Sistemas e Computação na COPPE/UFRJ com período sanduíche na University of Waterloo, Canadá. Fez carreira docente na UFRJ, onde ingressou no Instituto de Matemática em 1989, e na COPPE em 1991. Fez pós-doutorado em 1995 na University of Waterloo, Canadá. Atualmente é professora titular do Programa de Engenharia de Sistemas e Computação da COPPE, onde coordena o Núcleo de Excelência em Algoritmos Randomizados, Quânticos e Aproximativos: Projeto, Análise e Implementação de Soluções Eficientes para problemas Combinatórios Fundamentais. É pesquisadora na área de Ciência da Computação, com ênfase em Teoria da Computação, e lidera o grupo de algoritmos e combinatória da COPPE, atuando principalmente nos seguintes temas: teoria dos grafos, algoritmos e complexidade computacional. Tem bolsa de produtividade em pesquisa do CNPq desde 1992, estando atualmente no nível 1A. É desde 2005 Cientista do Nosso Estado FAPERJ. Recebeu em 2006 o Prêmio Giulio Massarani de Mérito Acadêmico da COPPE. Recebeu em 2013 homenagem na solenidade comemorativa dos 50 anos da COPPE.

Eduardo Felipe Zambom Santana é bacharel (2007) e mestre (2010) em Ciência da Computação pela Universidade Federal de São Carlos e atualmente é aluno de doutorado em Ciência da Computação no Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP) com tese de doutorado na área de Cidades Inteligentes. Tem mais de 10 anos de experiência como analista e arquiteto de sistemas em empresas de pequeno e grande porte e mais de 4 anos como professor de graduação em universidades privadas. Nos últimos 2 anos tem realizado pesquisas na área de cidades inteligentes principalmente em simulações de cenários de mobilidade urbana e na identificação de requisitos funcionais e não-funcionais para o desenvolvimento de uma plataforma de software para cidades inteligentes.

Fabio Kon é Professor Titular do Departamento de Ciência da Computação do IME-USP. Graduiu-se em Ciência da Computação pela USP (1990) e em Música pela UNESP (1992). Obteve seu doutoramento em Ciência da Computação na Universidade de Illinois em Urbana-Champaign, EUA (2000), onde também atuou como pesquisador da Startup VOSAIC, a primeira empresa de streaming de vídeo da história da Internet, uma spinoff do grupo de pesquisa onde realizava o doutorado. Atua em Ciência da Computação e sua principal área de pesquisa é Sistemas Distribuídos e Middleware, tendo recebido o prêmio “10-Year Best Paper Award” na International Middleware Conference em 2010. Realiza também pesquisas em Computação Musical, Software Livre, Desenvolvimento Ágil de Software e nas interfaces entre a Computação e a áreas de Música e Saúde. Foi membro da diretoria internacional da Open Source Initiative (OSI), é vice-diretor do Centro de Competência em Software Livre do IME/USP, Conselheiro da SBC e Editor-Chefe do periódico de acesso aberto Springer *Journal of Internet Services and Applications*. Possui mais de 120 artigos completos publicados em periódicos e congressos com revisão/seleção por pares, sempre na área de tecnologia e software.

Flávia de Almeida Barros é professora do Centro de Informática da Universidade Federal de Pernambuco desde 1996, atualmente no nível Professor Associado 3. Obteve o título de PhD em Ciência da Computação, com foco em Processamento de Linguagem Natural (PLN), pela University of Essex (1995). Recentemente, realizou estágio de Pós-doutorado na University of Bristol, com foco em Agentes Conversacionais Incorporados (2015). Atua na área de PLN há mais de 20 anos, já tendo oferecido um curso na JAI sobre o tema (1996). Tem atuado como membro de comitês de programa e como revisora de artigos na área de Inteligência Artificial (IA) e PLN. Possui publicações em fóruns nacionais e internacionais nas suas áreas de atuação. Já orientou e orienta alunos de graduação, mestrado e doutorado em temas correlatos. Destaca-se aqui um trabalho de doutorado que resultou em uma linguagem para construção de *chatterbots* com intencionalidade baseada na Teoria da Conversação, denominada de iAIML.

João Luís Garcia Rosa é professor livre docente do Departamento de Ciências da Computação, na Universidade de São Paulo (USP) — Instituto de Ciências Matemáticas e de Computação (ICMC) — em São Carlos, onde ministra aulas na graduação e pós-graduação de Redes Neurais, Inteligência Artificial, Teoria da Computação e Linguagens Formais. É graduado em Engenharia Elétrica-Eletrônica e Automação (1983), mestre em Engenharia de Computação (Inteligência Artificial - Processamento de Línguas Naturais) (1993) e doutor em Lingüística (Computacional) (1999), todos pela Universidade Estadual de Campinas (Unicamp). Realizou estágio pós-doutoral em Neurociência Computacional, na área de Modelos de cérebro no Computational Neurodynamics Laboratory (CNL), na University of Memphis, Estados Unidos, em 2013. Atua como orientador pleno de mestrado e doutorado no Programa de Pós-graduação em Ciência da Computação e Matemática Computacional do ICMC-USP. Está vinculado ao Laboratório de Computação Bioinspirada (BioCom). Tem experiência na área de Ciência da Computação, com ênfase em Inteligência Artificial, atuando principalmente nos seguintes temas: interfaces cérebro-computador, neurodinâmica computacional, e redes neurais artificiais biologicamente plausíveis.

Luís C. Lamb é PhD in Computer Science pela Imperial College London, University of London (2000), Mestre pela UFRGS (1995) e Bacharel em Ciências de Computação pela UFRGS (1992). Realizou estágio pós-doutoral no *Group of Logic, Language and Computation*, King's College London (2005-2006). Em 2010, concluiu o MIT Executive Program in Strategy and Innovation e em 2014 concluiu o Executive Program in Management and Leadership. Atualmente é Professor Titular do Departamento de Informática Teórica e Diretor do Instituto de Informática da UFRGS (2011 a 2015). É *Honorary Visiting Fellow* na City University London. É autor de dois livros científicos: “Neural-Symbolic Cognitive Reasoning”, Springer (2009) e “Compiled Labelled Deductive Systems” sobre lógicas não-clássicas (IoP, 2004), além de diversos artigos em conferências e periódicos de referência. Tem experiência nas áreas de Ciência da Computação e Lógica Aplicada. É co-editor-chefe da Revista de Informática Teórica e Aplicada, membro do *Editorial Board* do *Logic Journal of the Interest Group in Pure and Applied Logics*, na subárea de *Algorithms and Neural Networks* e membro do *Editorial Board* do *Journal of the Brazilian Computer Society*. Atualmente é membro do Comitê Assessor de Ciência da Computação do CNPq e Bolsista de Produtividade em Pesquisa nível 1C do CNPq.

Patricia Tedesco é Professora Adjunto 4 do Centro de Informática da UFPE. Obteve o título de Ph.D. pela University of Leeds em 2001. Realizou o doutorado na área de Trabalho Colaborativo Suportado por Computador, com publicações nas duas maiores conferências da área (ITS e AIED). Recebeu prêmio de melhor artigo do SBIE'2001. Foi membro do Comitê de Programa de várias edições do SBIE, WIE e WEI, bem como revisora de conferências importantes da área de Inteligência Artificial, como o AAMAS. Tem uma publicação no *International Journal of Artificial Intelligence in Education*, bem como publicações no JBCS, RBIE. Possui publicações em fóruns nacionais e internacionais na área de Inteligência Artificial na Educação, e já orientou vários alunos de doutorado e de mestrado nesta área.

Thiago Henrique Silva é pesquisador no Departamento de Ciência da Computação da UFMG. Em 2011, esteve como pesquisador visitante na Telecom Italia, Itália, onde trabalhou na área de ambientes inteligentes. Durante o seu doutoramento, fez estágios na University of Birmingham, Reino Unido, e no INRIA Paris, França. Ao longo de sua carreira acadêmica foi agraciado com os prêmios de melhor trabalho/menção honrosa nos seguintes eventos: *IEEE International Conference on Cyber, Physical and Social Computing* (2012), no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (2009, 2010 e 2013), no Simpósio Brasileiro de Computação Ubíqua e Pervasiva (2012) e no concurso de aplicativos em dados abertos promovido pelo Ministério da Justiça (2013).

Capítulo

1

Cidades Inteligentes: Conceitos, plataformas e desafios

Fabio Kon, Eduardo Felipe Zambom Santana

Abstract

With the growth of the urban population, the infrastructural problems and limited resources of thousands of cities around the world affect negatively the lives of billions of people. Making cities smarter can help improving city services and increasing the quality of life of their citizens. Information and communication technologies (ICT) are a fundamental means to move towards smarter city environments. Using a software platform on top of which Smart City applications can be deployed facilitates the development and integration of such applications. However, there are, currently, significant technological and scientific challenges that must be faced by the ICT community before these platforms can be widely used. This chapter presents the state-of-the-art and the state-of-the-practice in Smart Cities environments. We analyze eleven smart city platforms and eleven smart city initiatives with respect to the most used enabling technologies as well as functional and non-functional requirements. Finally, we enumerate open research challenges and comment on our vision for the area in the future.

Resumo

Com o crescimento da população urbana, problemas de infraestrutura e de acesso limitado a recursos em diversas cidades ao redor mundo afetam negativamente a vida de bilhões de pessoas. Tornar as cidades mais inteligentes pode ajudar a melhorar os serviços urbanos aumentando a qualidade de vida de seus cidadãos. A Tecnologia da Informação e Comunicação (TIC) são meios fundamentais para esse objetivo. Uma plataforma de software pode ser usada para facilitar enormemente a criação e integração de aplicações robustas para cidades inteligentes. Entretanto, ainda existem desafios técnicos e científicos significativos que necessitam ser enfrentados antes que essas plataformas possam ser amplamente utilizadas. Este capítulo apresenta o estado da arte e o estado da prática em iniciativas e ambientes de cidades inteligentes. Para isso, analisamos onze

iniciativas internacionais de cidades inteligentes e onze projetos científicos para a criação de plataformas de software para cidades inteligentes. Descrevemos as tecnologias mais utilizadas por esses projetos e seus requisitos funcionais e não-funcionais. Finalmente, apresentamos os desafios de pesquisa ainda em aberto e discutimos a nossa visão para a área no futuro.

1.1. Introdução

Desde 2009, a maior parte da população mundial vive em cidades (United Nations 2009) e a infraestrutura e os recursos existentes nessas cidades muitas vezes não são suficientes para comportar o crescimento e a concentração da população. Uma forma de enfrentar esse problema é tornando as cidades mais inteligentes, otimizando o uso dos seus recursos e infraestrutura de uma forma sustentável e melhorando a qualidade de vida de sua população. Para atingir esse objetivo, tecnologias de informação e comunicação (TIC) podem ser empregadas para coletar e analisar uma grande quantidade de informações geradas por diversas fontes de dados da cidade como, por exemplo, redes de sensores, sistemas de trânsito e dispositivos dos cidadãos. Esses dados podem ser utilizados de formas inovadoras e criativas para a criação de aplicações integradas que melhorem os serviços da cidade e o uso de seus recursos. Entretanto, usar todos os dados de uma forma efetiva e eficiente é um desafio bastante complexo.

Neste capítulo serão mostradas aplicações e serviços de Cidades Inteligentes desenvolvidas em uma grande variedade de cenários, como por exemplo na melhoria e monitoramento do trânsito (Djahel et al. 2014; Barba et al. 2012), no monitoramento das condições da cidade (Vakali et al. 2014), para o controle de multidões (Franke et al. 2015), para o monitoramento do sistema de coleta de lixo (Perera et al. 2014), em sistemas de saúde (Hussain et al. 2015), de segurança pública (Galache et al. 2014) e o gerenciamento de recursos como água (Pérez-González and Díaz-Díaz 2015) e energia elétrica (Yamamoto et al. 2014).

Um problema na maioria das aplicações de cidades inteligentes, é que normalmente os sistemas são direcionados a um problema específico e são desenvolvidos sempre desde o início com pouco reuso de software e sem a comunicação entre esses sistemas. Essa abordagem leva a um maior trabalho, ao uso não otimizado dos recursos e impede a criação de aplicações que necessitem de dados e serviços de diversos domínios, o que é uma das principais características de cidades inteligentes.

Para resolver os problemas de integração entre as aplicações, uma das abordagens mais utilizadas tanto em pesquisas acadêmicas como em experimentos já realizados em algumas cidades é a utilização de uma plataforma de software que oferece diversos mecanismos e características não-funcionais para a utilização dos dados e serviços da cidade de uma forma integrada e com o objetivo de facilitar a implementação de aplicações de Cidades Inteligentes para cidadãos e administradores da cidade.

No entanto, muitos desafios técnicos e de pesquisa ainda precisam ser resolvidos antes que ambientes de Cidades Inteligentes eficazes e robustos sejam completamente desenvolvidos. Alguns dos maiores desafios são: permitir a interoperabilidade entre os diversos componentes da cidade, garantir a privacidade e a segurança dos cidadãos e sistemas da cidade, gerenciar o armazenamento e o processamento de grandes quantidades

de dados, oferecer a escalabilidade necessária para o aumento da população da cidade e lidar com a heterogeneidade de dispositivos como sensores e *smartphones*.

Este capítulo tem o objetivo de apresentar o conceito de Cidades Inteligentes, apontando as definições mais aceitas na literatura e discutindo as tecnologias necessárias para o desenvolvimento de cidades inteligentes. Depois serão descritas iniciativas realizadas em algumas cidades ao redor do mundo e projetos de pesquisa para a implementação de plataformas de software para Cidades Inteligentes incluindo os desafios técnicos e de pesquisa que ainda precisam ser resolvidos.

O restante deste capítulo está organizado da seguinte maneira, a Seção 1.2 discute as mais citadas definições de cidades inteligentes. A Seção 1.3 apresenta as principais tecnologias utilizadas para a implantação de cidades inteligentes. A Seção 1.4 apresenta diversas iniciativas de cidades inteligentes ao redor do mundo. A Seção 1.5 descreve plataformas de cidades inteligentes desenvolvidas em projetos de pesquisa e comerciais. A Seção 1.6 apresenta os requisitos identificados a partir das plataformas e iniciativas analisadas. A Seção 1.7 apresenta uma arquitetura de referência derivada a partir dos requisitos. A Seção 1.8 lista os principais desafios técnicos e de pesquisa para a criação de cidades inteligentes. A Seção 1.9 discute as implicações de Cidades Inteligentes para os diferentes papéis da cidade. Finalmente, a Seção 1.10 aponta as conclusões deste capítulo.

1.2. Definições de Cidades Inteligentes

Nesta seção serão apresentadas e discutidas diversas definições de Cidades Inteligentes encontradas na literatura. Essas definições consideram desde as mudanças sociais esperadas com Cidades Inteligentes como o empoderamento e a melhora na qualidade de vida da população até o uso de TICs para a melhora na infraestrutura e nos serviços da cidade e na otimização do uso dos recursos da cidade.

A Tabela 1.1 apresenta diversas definições de cidades inteligentes. A maioria dessas definições citam explicitamente que o objetivo de uma cidade inteligente é a melhoria da qualidade de vida do cidadão. Algumas definições (Giffinger et al. 2007; Guan 2012) não estabelecem por qual meio isso deve ser alcançado, enquanto outras definem que isso será alcançado através da construção de uma infraestrutura tecnológica para melhorar os serviços da cidade (Caragliu et al. 2011; Dameri 2013; Harrison et al. 2010).

A maioria das definições citam a necessidade do uso de tecnologia da informação para otimizar o uso da infraestrutura da cidade, o gerenciamento dos recursos e os serviços da cidade (Harrison et al. 2010; Washburn et al. 2009). Algumas dessas definições ainda adicionam a necessidade do desenvolvimento sustentável da cidade, com a melhoria no uso de recursos como água e energia elétrica (Caragliu et al. 2011; Dameri 2013).

Um aspecto relevante é a necessidade de uma cidade inteligente facilitar também o crescimento econômico da cidade (Dameri 2013) possibilitando a inclusão e participação de toda a população na sociedade. Duas definições (Dameri 2013; Giffinger et al. 2007) citam a participação da sociedade na decisão dos governos através de governos participativos. Outras questões importantes levantadas pelas definições, é o monitoramento da infraestrutura da cidade, como ruas, pontes, linhas de trem (Hall et al. 2000), o monitoramento do uso de recursos como água e energia elétrica (Hall et al. 2000) e a integração

Tabela 1.1. Definições de Cidades Inteligentes

Definição	Autor
“A Smart City is a city well performing built on the ‘smart’ combination of endowments and activities of self-decisive, independent and aware citizens”	(Giffinger et al. 2007)
“A city to be smart when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance”	(Caragliu et al. 2011)
“A smart city is a well-defined geographical area, in which high technologies such as ICT, logistic, energy production, and so on, cooperate to create benefits for citizens in terms of well-being, inclusion and participation, environmental quality, intelligent development; it is governed by a well-defined pool of subjects, able to state the rules and policy for the city government and development”	(Dameri 2013)
“A city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, rails, subways, airports, seaports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens”	(Hall et al. 2000)
“A city connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city”	(Harrison et al. 2010)
“A smart city, according to ICLEI, is a city that is prepared to provide conditions for a healthy and happy community under the challenging conditions that global, environmental, economic and social trends may bring.”	(Guan 2012)
“The use of Smart Computing technologies to make the critical infrastructure components and services of city which include city administration, education, healthcare, public safety, real estate, transportation, and utilities more intelligent, interconnected, and efficient”	(Washburn et al. 2009)

entre todos os serviços da cidade (Harrison et al. 2010; Washburn et al. 2009).

Além das definições apresentadas anteriormente, Giffinger et al. (Giffinger et al. 2007) descreve seis dimensões para verificar o quão inteligente é uma cidade, que são: *Smart Economy*, *Smart People*, *Smart Governance*, *Smart Mobility*, *Smart Environment* e *Smart Living*, os quais traduzimos livremente como economia, população, governança, mobilidade, meio-ambiente e vida inteligentes. Muitos autores aceitam essa classificação (Hernández-Muñoz et al. 2011; Papa et al. 2013) e há ainda um *benchmark* desenvolvido para classificar as cidades mais inteligentes da Europa usando essas dimensões ¹. Essas dimensões são definidas da seguinte forma:

- **Economia Inteligente** mede o quão bem preparada economicamente uma cidade está, utilizando parâmetros como qualidade das empresas instaladas e o seu ambiente para empreendedorismo. Algumas ações desenvolvidas relacionadas a esta dimensão são incentivos a empresas para o desenvolvimento de soluções tecnológicas para a cidade e a melhoria do ambiente de negócios com legislação adequada à inovação e infraestrutura para negócios.
- **População Inteligente** mede o desenvolvimento da população da cidade usando parâmetros como educação, emprego e renda. Algumas ações relacionadas a esta dimensão são projetos para inclusão digital dos cidadãos e programas de educação científica e tecnológica.
- **Governança Inteligente** mede o qualidade e transparência dos órgãos públicos municipais com parâmetros como facilidade no uso dos serviços públicos, investimentos em tecnologia e transparência nos dados e no uso de recursos da cidade. Algumas ações relacionadas a esta dimensão são a criação de governos participativos e a divulgação de informações sobre a cidade em portais de transparência e de dados abertos.
- **Mobilidade Inteligente** mede a facilidade da mobilidade na cidade nos diversos modais de transporte como ônibus, metrô, carro e bicicleta. Usa parâmetros como quilômetros de congestionamento, tamanho da malha metroviária e quantidade de pessoas que usam transporte público ou não-poluente. Algumas ações relacionadas a esta dimensão são o monitoramento em tempo real do fluxo nas vias da cidade, o uso de sensores para indicar vagas de estacionamento livres e aplicações para facilitar e incentivar o uso de transporte público e sustentável, tais como bicicletas.
- **Meio-Ambiente Inteligente** mede a sustentabilidade na cidade usando parâmetros como poluição ambiental, eficiência no uso de recursos como água e energia elétrica e a quantidade de lixo reciclado. Algumas ações relacionadas a esta dimensão são a medição da qualidade do ar e água da cidade, o uso de fontes renováveis de energia e a medição em tempo real dos recursos utilizados em residências.
- **Vida Inteligente** mede a qualidade de vida da população usando parâmetros como entretenimento, segurança e cultura como quantidade de áreas verdes, número de

¹Smarts Cities in Europe - <http://www.smart-cities.eu>

bibliotecas e taxa de homicídios da cidade. Algumas ações relacionadas a esta dimensão são o uso de aplicações para o acompanhamento da saúde de idosos, o processamento automático de imagens de câmeras de segurança e aplicativos que mostram os eventos culturais programados na cidade.

Atualmente a expressão Cidades Inteligentes (*Smart Cities*) está bem estabelecido, porém existem algumas outras expressões que também indicam características similares à ideia de Cidades Inteligentes. Algumas dessas expressões são: Cidades Digitais (*Digital City*), Cidades do Conhecimento (*Knowledge City*) e Cidades Conectadas (*Wired City*).

Neste capítulo apenas a expressão Cidades Inteligentes (*Smart Cities*) foi considerado, isso porque, atualmente ele é o mais utilizado, como mostra a Figura 1.1 gerada pela ferramenta Google Trends², a qual mostra a quantidade de buscas feitas por cada expressão. Além disso, essa expressão é a única que claramente define que a cidade deve disponibilizar serviços integrados aumentando a inteligência da cidade para melhorar a qualidade de vida do cidadão.

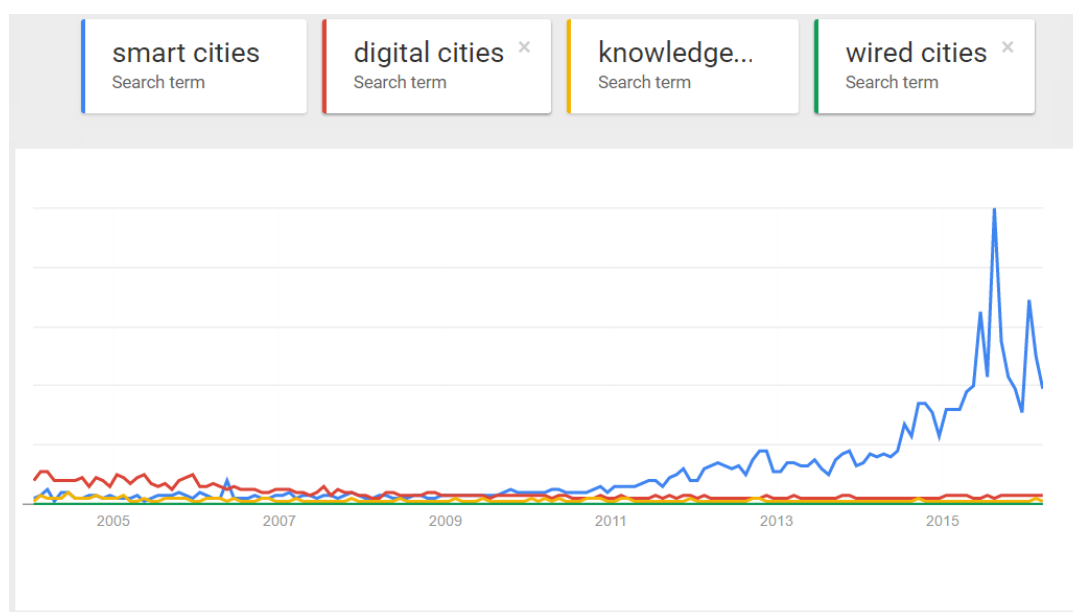


Figura 1.1. Pesquisas relacionados a expressões relacionadas a Cidades Inteligentes

As expressões Cidades Digitais e Cidades Conectadas estão relacionadas ao fornecimento de serviços digitais aos cidadãos utilizando uma infraestrutura de TI, mas sem necessariamente a integração entre os diversos tipos de aplicações e domínios que podem existir em uma cidade. A expressão Cidade do Conhecimento, além de ser pouco usado, está relacionado mais ao domínio da educação, tendo o objetivo de empoderar o cidadão através da educação. Dois trabalhos apresentam uma discussão sobre essas diferentes expressões (Cocchia 2014; Yin et al. 2015).

Existem iniciativas em diversas cidades ao redor do mundo, a maioria na Europa

²Google Trends - <https://www.google.com/trends/>

(Caragliu et al. 2011; Manville et al. 2014), diversas nos Estados Unidos ³, Japão e China (Liu and Peng 2013) e alguns projetos em outras partes do mundo como Brasil (Fortes et al. 2014), Emirados Árabes (Janajreh et al. 2013) e Coreia do Sul (Kshetri et al. 2014). Esses dados mostram que a grande maioria dos projetos estão concentrados em países desenvolvidos, existem alguns poucos projetos em países em desenvolvimento. Nenhum projeto foi encontrado nos países mais pobres do globo. A Figura 1.2 mostra um mapa com as iniciativas encontradas na literatura ou páginas dos projetos.



Figura 1.2. Iniciativas de Cidades Inteligentes ao redor do mundo.

Na Seção 1.4 deste capítulo serão apresentadas algumas dessas iniciativas de Cidades Inteligentes selecionadas pela qualidade e pela quantidade de informações encontradas sobre o projeto.

1.3. Tecnologias e Conceitos

Apresentamos agora as principais tecnologias usadas na criação da infraestrutura de uma cidade inteligente, i.e., (1) Internet das Coisas, para possibilitar a conexão de diversos dispositivos na rede da cidade como sensores, sinais de trânsito e dispositivos de usuários; (2) Big Data, viabilizando o armazenamento e o processamento de grandes quantidades de dados coletados na cidade e (3) Computação em Nuvem, fornecendo um ambiente escalável e elástico que suporte a grande demanda de recursos computacionais necessários em uma cidade inteligente.

1.3.1. Internet das Coisas

A Internet das Coisas (*Internet of Things* ou IoT) é a conexão de objetos do cotidiano, tais como lâmpadas para iluminação pública, semáforos de trânsito, sensores de qualidade de água e ar, câmeras de vídeo, etc. à rede Internet. Os objetos devem ser identificados com um nome único, sua posição e estado conhecidos, e devem ser acessíveis por meio de uma rede interoperável (Coetzee and Eksteen 2011).

³10 Smartest Cities in USA - <http://www.fastcoexist.com/3021592/the-10-smartest-cities-in-north-america>

Podem haver uma grande variedade de “Coisas” conectadas em um sistema de IoT, desde celulares, relógios e computadores até veículos e geladeiras. O *Cluster of European Research Projects on IoT* (Sundmaeker et al. 2010) define “Coisas” como participantes ativos da rede que são capazes de interagir e comunicar-se com outros elementos da rede e com o ambiente. Essa comunicação ocorre para a troca de dados e informações sobre o ambiente. A Internet das Coisas conecta o mundo digital e físico adicionando serviços e inteligência para a internet sem a intervenção direta de seres humanos. Podemos destacar três componentes principais em um sistema de IoT: (1) o hardware, como sensores, atuadores e aparelhos de comunicação; (2) o middleware para o processamento e armazenamento dos dados capturados pelo hardware e (3) uma camada de apresentação na qual usuários ou administradores do sistema podem acessar, manipular e analisar os dados (Gubbi et al. 2013).

A Internet das Coisas é bastante adequada para o gerenciamento dos milhares de dispositivos que estarão conectados em uma cidade inteligente. Assim, os dados coletados na cidade são enviados para as plataformas de software ou para as aplicações para que sejam armazenados e processados possibilitando a criação de serviços inovadores para a cidade.

Diversas iniciativas de Cidades Inteligentes usam IoT para a manutenção e gerenciamento dos dispositivos da cidade, como por exemplo o SmartSantander (Sanchez et al. 2014) que já possui mais de 20 mil sensores instalados na cidade de Santander, o Padova Smart City (Zanella et al. 2014), que possui mais de 300 sensores instalados e o *Array of Things*, que está instalando uma grande rede de sensores na cidade de Chicago.

A Internet das Coisas possui uma enorme quantidade de aplicações potenciais em Cidades Inteligentes. Alguns exemplos são: monitoramento da estrutura de prédios históricos, detecção se latas de lixo estão cheias, monitoramento de barulho perto de áreas críticas como escolas e hospitais, monitoramento das condições de semáforos e lâmpadas de iluminação pública e o monitoramento do uso de energia elétrica e água em Casas Inteligentes (Zanella et al. 2014).

1.3.2. Big Data

A expressão *Big Data* se refere a um conjunto de técnicas e ferramentas para o armazenamento e manipulação de conjuntos de dados muito grandes, onde tecnologias tradicionais, como bancos de dados relacionais e ferramentas de processamento sequencial, não suportam o vasto volume de dados (Chen et al. 2014; Demchenko et al. 2014). Big Data possui quatro características marcantes ilustradas na Figura 1.3:

- **Volume:** a quantidade de dados gerados e coletados em diversos tipos de aplicações está aumentando exponencialmente e as ferramentas de Big Data devem ser capazes de lidar apropriadamente com esse desafio.
- **Variedade:** os dados podem ser coletados de diferentes fontes e com diferentes formatos e estruturas; como dados estruturados como os dados dos cidadãos, dados semi-estruturados como os dados de sensores e os dados não-estruturados como câmaras de vídeo de segurança e de trânsito.

- **Velocidade:** o processamento de dados deve ser rápido e, em muitos casos, em tempo real, ou esses dados podem se tornar inúteis como dados coletados de sensores de veículos, a análise de redes sociais e informações sobre o trânsito da cidade.
- **Veracidade:** como os dados serão coletados de múltiplas fontes de dados, é importante garantir a qualidade desses dados, utilizando fontes confiáveis e consistentes. Isso é importante para evitar erros comprometendo a análise dos dados.

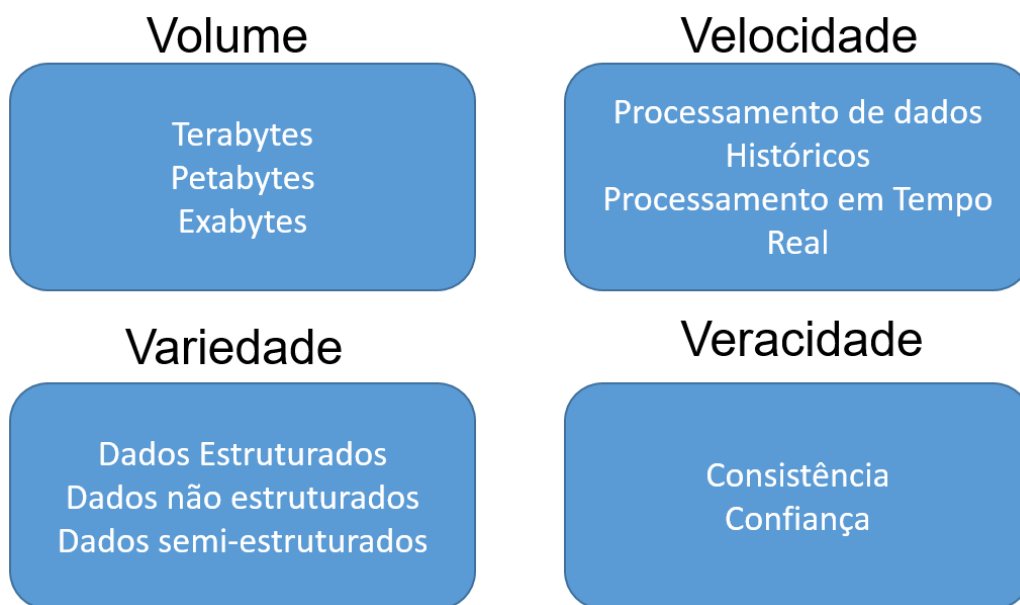


Figura 1.3. 4 Vs de Big Data

Big Data não é apenas uma estrutura de armazenamento moderna e escalável como bancos de dados NoSQL, ou ferramentas de processamento paralelo poderosas como o Hadoop (Polato et al. 2014; Goldman et al. 2012), mas sim a transformação de todo o ciclo de vida dos dados dentro de um aplicação, para suportar a coleta, armazenamento, processamento, análise e visualização de grandes conjuntos de dados.

No contexto de cidades inteligentes, ferramentas de Big Data estão sendo empregadas para permitir o gerenciamento da grande quantidade de dados gerados nas cidades. Por exemplo, dados que são gerados por sensores periodicamente sobre as condições da cidade como temperatura, qualidade do ar e pluviometria, dados gerados por cidadãos através de telefones celulares e redes sociais e veículos como ônibus que podem enviar periodicamente sua posição e velocidade para aplicações.

Muitas ferramentas de Big Data já estão sendo usadas por iniciativas e plataformas de software para cidades inteligentes. Algumas dessas ferramentas são bancos de dados NoSQL (Khan et al. 2013; Bain 2014) como o MongoDB e o HBase, ferramentas de processamento paralelo (Parkavi and Vetrivelan 2013; Takahashi et al. 2012) como o Apache Hadoop e o Apache Spark, processadores de fluxos de dados em tempo real (Girtelschmid et al. 2013) como o Apache Storm e ferramentas de visualização de dados (Khan et al. 2013) como o RapidMiner.

Os bancos de dados NoSQL são utilizados principalmente para armazenar dados não estruturados da cidade, como por exemplo leituras de sensores e posições de veículos. Ferramentas de processamento paralelo são utilizadas para o processamento de grandes conjuntos de dados, e são utilizados tanto para processamento de dados históricos com o Hadoop ou Spark quanto para processamento em tempo-real de grandes fluxos de dados também com o Spark e o Storm.

Há inúmeras possibilidades de aplicação de tecnologias e ferramentas de Big Data em cidades inteligentes. Podemos citar como exemplos o reconhecimento de padrões em trânsito usando dados históricos para descobrir as causas e evitar congestionamentos, facilitar as decisões de administradores da cidade usando análises sobre grandes conjuntos de dados, prever a quantidade de energia elétrica utilizada em diferentes dias e horários utilizando dados históricos e fluxos de dados em tempo real, prever a demanda do uso de transporte público utilizando dados históricos sobre a venda de passagens e detectar automaticamente problemas de segurança pública utilizando fluxos de dados de sensores e redes sociais (Al Nuaimi et al. 2015).

1.3.3. Computação em Nuvem

A Computação em Nuvem oferece uma infraestrutura elástica, robusta e altamente disponível para o armazenamento e processamento de dados, o que é essencial para aplicações de Cidades Inteligentes. Adicionalmente, uma cidade inteligente pode ser altamente dinâmica, requerendo reconfigurações automáticas de sua infraestrutura, o que é também facilitado pela computação em nuvem.

Alguns autores (Distefano et al. 2012; Aazam et al. 2014) descrevem um novo paradigma a partir da combinação da Computação em Nuvem e da Internet das Coisas, chamando-o de “Cloud of Things”. A ideia é armazenar e processar todos os dados coletados de uma rede IoT em um ambiente de Computação em Nuvem, o que já é realizado em diversas iniciativas de cidades inteligentes (Mitton et al. 2012; Tei and Gurgun 2014).

Outro conceito relacionado ao uso da Computação em Nuvem em Cidades Inteligente é o Software como Serviço (Software as a Service - SaaS) (Fox et al. 2013). Perera et al. estendem esse conceito, usando a expressão “Sensing as a Service” (Perera et al. 2014). O objetivo é fornecer a aplicações e serviços os dados de sensores em uma infraestrutura de Computação em Nuvem. A plataforma ClouT também usa esse conceito e define as expressões *City Application Software as a Service* (CSaaS) e *City Platform as a Service* (CPaaS) disponibilizando todos as funcionalidades da plataforma como serviços em um ambiente de Computação em Nuvem (Tei and Gurgun 2014).

Resumindo, Computação em Nuvem é ideal para fornecer a infraestrutura para armazenar e executar os serviços de uma cidade. Os dados podem ser colhidos por uma rede implantada com as ideias de Internet das Coisas e enviadas para a infraestrutura de Computação em Nuvem, onde os dados podem ser processados utilizando ferramentas de Big Data. Essa combinação ajuda a oferecer diversos requisitos não-funcionais importantes como escalabilidade, elasticidade e segurança (Chen et al. 2014; Aazam et al. 2014).

1.4. Iniciativas de Cidades Inteligentes

Nesta seção, apresentamos iniciativas e experimentos de cidades inteligentes realizados em cidades em diversos países ao redor do mundo. Alguns exemplos dessas cidades são: (1) Santander, cidade Espanhola que, utilizando a plataforma SmartSantander, realizou diversos experimentos de aplicações e serviços para o cidadão, (2) Barcelona, também na Espanha, que utiliza serviços inteligentes para a coleta de lixo e a utilização de carros elétricos, (3) Amsterdã na Holanda, onde foram desenvolvidos diversos projetos na área de trânsito, de monitoramento da cidade e apoio a realização de grandes eventos e (4) Chicago, nos Estados Unidos, que através da plataforma WindyGrid coleta diversos dados gerados na cidade, como ligações de emergência e dados de trânsito, para o monitoramento da cidade.

1.4.1. Santander

Santander é uma cidade no norte da Espanha, capital da Cantábria e tem uma população de aproximadamente 180 mil habitantes. Nela foi implantada um projeto experimental, para o desenvolvimento de uma plataforma de cidade inteligente, chamado SmartSantander (Sanchez et al. 2014) financiado pela *European Comission*.

Para esse projeto, foi implantada uma rede de mais de 20 mil sensores e atuadores na cidade que coletam uma grande quantidade de dados em diversas regiões da cidade, como temperatura, espaços livres de estacionamento, identificadores de pontos de interesse e luminosidade. Além dos sensores, a plataforma também coleta dados de ônibus, caminhões de lixo e táxis utilizando dispositivos móveis instalados nos veículos. A Figura 1.4 mostra um mapa no qual cada ponto é uma elemento da cidade que envia dados para a plataforma.

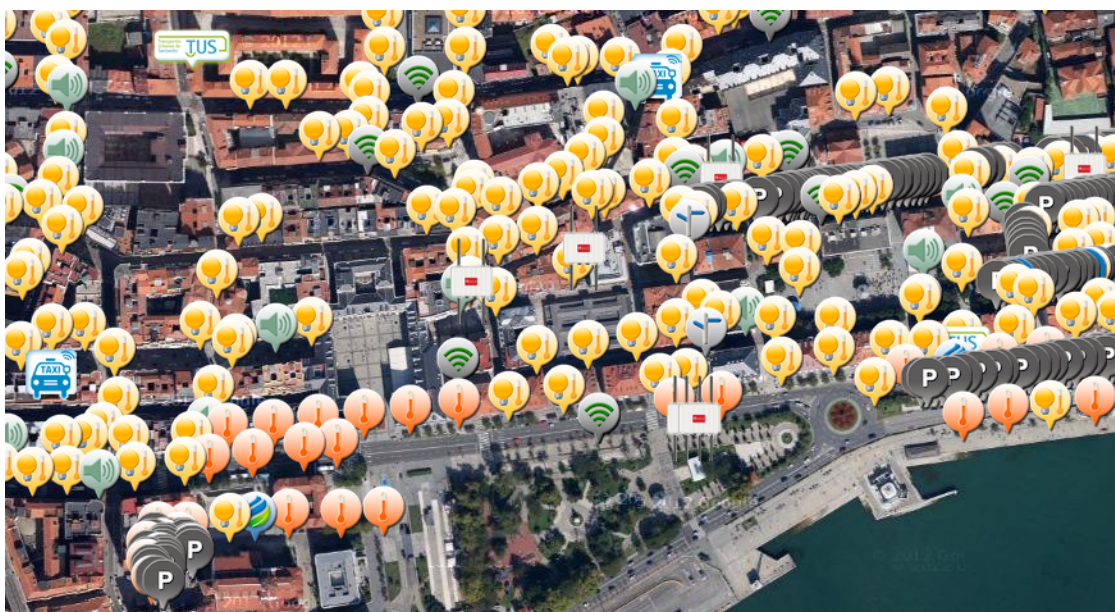


Figura 1.4. Mapa dos elementos da plataforma Smart Santander

A plataforma SmartSantander foi utilizada para o desenvolvimento de vários projetos como, por exemplo, para mostrar para os usuários os lugares livres para estaciona-

mento na cidade e também para prever a utilização desses lugares em eventos na cidade (Vlahogianni et al. 2014).

Foi desenvolvida também uma aplicação de realidade aumentada⁴ para dispositivos móveis que contém informação de mais de 2700 pontos de interesse da cidade como museus, livrarias, pontos de ônibus, oficinas de turismo e estações de aluguel de bicicletas, além de mostrar em tempo real a posição de ônibus e taxis.

1.4.2. Barcelona

Barcelona é uma cidade Espanhola, capital da região da Catalunha, e tem uma população de aproximadamente 4,7 milhões de habitantes. O projeto BCN Smart City⁵ desenvolve diversas iniciativas científicas e comerciais para tornar a cidade mais inteligente. Alguns exemplos de projetos são a implantação de uma rede de sensores para notificar quando as lixeiras da cidade estão cheias, o desenvolvimento de *dashboards* para o monitoramento das condições da cidade e um portal de dados abertos da cidade que podem ser utilizados para a implementação de aplicações e serviços para os cidadãos.

A cidade de Barcelona também está desenvolvendo projetos para incentivar o uso de formas sustentáveis de transporte, como um projeto para estimular o uso de carros elétricos, no qual mais de 300 pontos de recarga de carros foram instalados na cidade e o projeto para o uso de bicicletas compartilhadas que conta com mais de 420 estações para o empréstimo de bicicletas.

Outro aspecto interessante, é a liberação de diversos dados para que a população possa acompanhar e fiscalizar o poder público. O portal de dados abertos da cidade⁶ disponibiliza uma grande quantidade de dados da administração pública como orçamento e despesas, dos serviços oferecidos pela cidade e dados sobre a população da cidade.

Além das aplicações apresentadas, a prefeitura da cidade de Barcelona patrocinou o desenvolvimento da plataforma Sentilo (Bain 2014), que é responsável pela administração, monitoramento e coleta de dados de diversos sensores espalhados pela cidade. Alguns exemplos de sensores utilizados nessa plataforma são: sensores que indicam a utilização de lixeiras na cidade, mapeamento dos pontos de acesso a internet, sensores de luminosidade e de temperatura. Depois de concluída a implementação da plataforma, o código fonte foi liberado sob a licença LGPL3⁷. A Figura 1.5 mostra uma tela da plataforma com alguns dos sensores implantados na cidade.

1.4.3. Amsterdã

Amsterdã é a capital da Holanda e tem uma população de aproximadamente 2,5 milhões de habitantes. Nessa cidade, estão sendo realizados diversos projetos e experimentos para tornar a cidade mais inteligente e aumentar a qualidade de vida da população. Essas iniciativas contam com a colaboração do governo, universidades, empresas e da população da cidade. Existem projetos em diversos domínios de aplicações como *Smart Grids*, trânsito, redução da emissão de poluentes e no sensoriamento da cidade.

⁴SmartSantanderRA - <https://play.google.com/store/apps/details?id=es.unican.tlmat.smartsantanderra>

⁵BCN Smart City - <http://smartcity.bcn.cat/en>

⁶OpenData BCN - <http://opendata.bcn.cat/opendata/ca>

⁷Sentilo - <https://github.com/sentilo/sentilo>

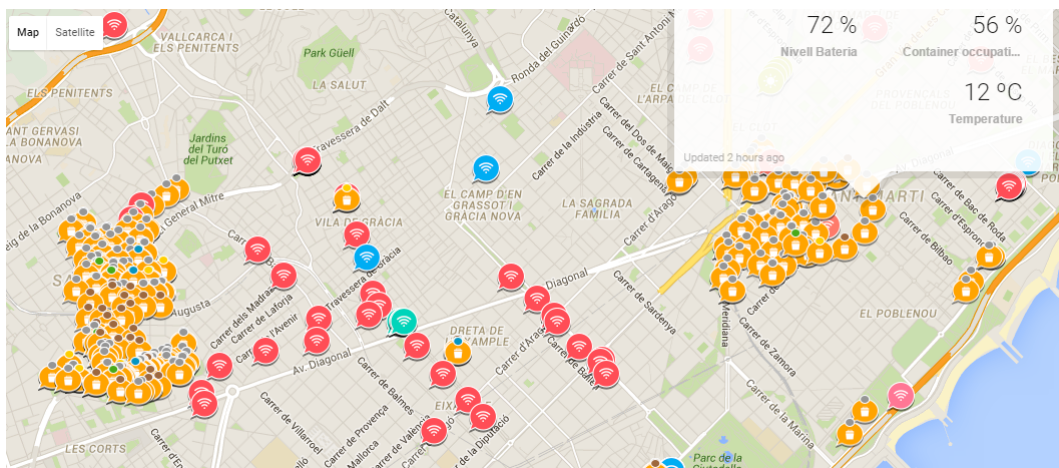


Figura 1.5. Sensores implantados com a plataforma Sentilo

O primeiro *Smart Electricity Grid* da Holanda está sendo implantado em uma região da cidade com aproximadamente 10 mil habitações. Nessa rede é possível que os usuários consumam e produzam energia e que possam acompanhar em tempo real o uso de energia em suas casas. Além disso, esse projeto também facilita o monitoramento e manutenção da rede pelas autoridades da cidade.

Na área de controle e monitoramento de trânsito, alguns projetos interessantes que estão sendo realizados na cidade são: o incentivo ao uso de carros elétricos, disponibilizando estações de recarga de bateria em várias partes da cidade, o monitoramento das principais vias da cidade para o rápido atendimento de problemas no trânsito, a reserva de vagas de estacionamento na cidade, evitando a busca por uma vaga, diminuindo a emissão de CO_2 e o incentivo ao uso de bicicletas.

Existem ainda diversos projetos para aumentar a transparência dos gastos e ações dos administradores da cidade, entre eles o *Budget Monitoring* que permite que cidadãos e entidades acessem e façam sugestões para o orçamento da cidade, o *Smart City SDK*, que permitem desenvolvedores de aplicações utilizem os dados coletados na cidade em tempo real como dados de trânsito, chegadas e partidas de aviões e o clima e o *AmsterdamOpent* que é uma plataforma para que cidadãos façam sugestões para os governantes da cidade.

Outro projeto interessante, é o *CitySDK Tourism API* (Pereira et al. 2015), uma ferramenta que permite o desenvolvimento de aplicações para ajudar os turistas que visitam a cidade. Essa ferramenta coleta os dados do portal de dados abertos da cidade, que estão em arquivos difíceis de serem processados por computadores como CSV, XLS e arquivos texto e os disponibiliza em uma API de fácil acesso e processamento para as aplicações. Alguns dos dados compartilhados são os pontos de interesse da cidade como museus, parques e construções históricas, eventos que estão acontecendo na cidade e itinerários turísticos.

1.4.4. Chicago

O *WindyGrid* (Thornton 2013) é uma plataforma que tem o objetivo de coletar, armazenar e processar os dados da cidade. Com isso, é possível visualizar as operações da cidade

de forma unificada utilizando dados em tempo real e dados históricos. Essa ferramenta é uma iniciativa da administração municipal de Chicago nos Estados Unidos.

Alguns dos dados coletados da cidade são estáticas e eventos sobre o trânsito da cidade, ligações de telefones de emergência (911), dados sobre edifícios públicos e publicações sobre a cidade em redes sociais como o Twitter. Na implementação da plataforma foram utilizadas ferramentas de Big Data como o banco de dados NoSQL MongoDB e ferramentas de processamento paralelo.

A plataforma disponibiliza três funções principais para os administradores da cidade: monitoramento de incidentes utilizando dados das ligações de emergência e de redes sociais; visualização de dados históricos, no qual o usuário pode ver todos os dados relacionados a um mesmo evento; e a análise de dados avançados em tempo real, na qual são mostrados, em um mapa da cidade, eventos que estão ocorrendo na cidade utilizando diversas fontes de dados.

Além da plataforma WindyGrid, Chicago possui um completo portal de dados abertos⁸ com milhares de fontes de dados sobre a cidade como por exemplo, o nome, salário e cargo de todos os funcionários da administração municipal, registro de ocorrências policiais, mapa veículos abandonados e dados censitários da população da cidade.

1.4.5. Outras Iniciativas de Cidades Inteligentes

Búzios, no estado do Rio de Janeiro, é uma das primeiras cidades do Brasil a (Fortes et al. 2014) iniciar um projeto para a implantação de uma infraestrutura de Cidade Inteligente. O projeto tem três objetivos principais, tornar a cidade mais sustentável, com uso mais racional dos recursos e com mais eficiência. Entre as principais ações realizadas na cidade estão a implantação de uma rede de energia elétrica inteligente, a criação de prédios inteligentes, onde seja possível monitorar o uso de recursos de casas e edifícios públicos e comerciais e a melhoria dos sistemas de comunicação da cidade utilizando tecnologias de comunicação como Wi-Fi, redes Mesh e pelas linhas de energia (Power Lines Communication - PLC).

Em Dublin, na Irlanda, existem dois projetos principais para tornar a cidade mais inteligente. O primeiro é a plataforma de dados abertos Dublinked (Stephenson et al. 2012) que possibilita a cidadãos, empresas e pesquisadores o acesso a mais de 200 conjuntos de dados, entre eles dados em tempo real das posições dos ônibus, monitoramento da cidade e de estações de aluguel de bicicleta. O segundo, é um conjunto de dashboards⁹ que disponibiliza diversas informações como temperatura, qualidade do ar níveis de ruído e nível dos rios em diversas partes da cidade.

Manchester, na Inglaterra, possui diversos projetos para tornar a cidade mais inteligente com três objetivos principais: engajar a população na tomada de decisões da cidade, aumentar a quantidade de empregos através da educação e criar serviços digitais inovadores para a população através de uma moderna infraestrutura tecnológica (Manville et al. 2014). O principal projeto sendo desenvolvido na cidade é o de construção de casas inteligente, na qual os moradores podem verificar em tempo real o uso de recursos

⁸Chicago Data Portal - <https://data.cityofchicago.org>

⁹Dublin DashBoards - <http://www.dublindashboard.ie/>

utilizados como energia elétrica e água. O objetivo desse projeto é diminuir a quantidade das emissões de carbono na cidade e na economia de recursos naturais.

O projeto *Intelligent Thessaloniki* (Komninos and Tsarchopoulos 2013) desenvolve diversos projetos na cidade de Thessalonica, Grécia, com o objetivo de implementar redes sem fim, espaços urbanos inteligentes, aplicações na Internet e serviços digitais para facilitar a vida dos cidadãos. Nesse projeto a cidade foi dividida em 5 distritos, e em cada um deles são testados diferentes projetos ligados a comunidade local, como por exemplo, na região portuária foram implantados projetos para melhoria do trânsito, na região central projetos para melhoria dos negócios e para o treinamento dos cidadãos nos serviços digitais e na região da universidade, projetos para melhoria de educação e para a criação de *startups*.

Seattle é considerada por alguns rankings a cidade mais inteligente dos Estados Unidos ¹⁰. Nessa cidade foi realizada uma pesquisa (AlAwadhi and Scholl 2013) com cidadãos e agentes públicos questionando quais os principais serviços, aplicações e iniciativas que estão sendo desenvolvidas para a melhoria da qualidade de vida da população da cidade e quais os seus maiores benefícios. Entre os projetos citados estão o portal de dados abertos da cidade ¹¹, a infraestrutura para apoiar o uso de carros elétricos e a instalação de um CRM (*Customer Relationship Management*) para controlar a comunicação de cidadãos com a prefeitura. A maioria dos benefícios apontados na adoção desses projetos são a melhoria dos serviços da cidade, a diminuição de custos, o aumento na eficiência e a economia de energia elétrica.

Masdar é um bairro na cidade de Abu Dhabi, nos Emirados Árabes Unidos que está sendo construído com o objetivo de testar diversas iniciativas de Cidades Inteligentes, principalmente no uso de fontes energéticas renováveis, no uso consciente de água e na redução da quantidade de lixo gerado. Além disso, a cidade foi planejada com uma rede de transporte inteligente para reduzir a necessidade do uso de veículos individuais, diminuindo a emissão de poluentes. Nesse bairro, todas as construções são projetadas de forma que economizem os recursos e produzam a própria energia com o uso de painéis solares.

Em São Paulo, diversas iniciativas estão sendo tomadas para a melhoria da qualidade de vida dos cidadãos, como por exemplo, a criação de faixas exclusivas de ônibus e de ciclovias, o incentivo ao uso e compartilhamento de bicicletas e a criação de laboratórios de inovação para buscar maneiras criativas de utilizar a tecnologia da informação para resolver problemas da cidade. Entre os projetos desenvolvidos recentemente na cidade, encontram-se: o portal de dados abertos da cidade ¹², o GeoSampa ¹³, que disponibiliza diversos dados cartográficos da cidade como a localização dos equipamentos públicos, pontos de ônibus, árvores, feiras livres, pontos de alagamento, etc. e a API Olho Vivo ¹⁴, que permite a descoberta em tempo real da posição de todos os ônibus da cidade, informação esta que permitiu o desenvolvimento de vários aplicativos móveis oferecendo

¹⁰<http://www.fastcoexist.com/3021592/the-10-smartest-cities-in-north-america>

¹¹data.seattle.gov - <https://data.seattle.gov/>

¹²Dados Abertos São Paulo - <http://saopauloaberta.prefeitura.sp.gov.br/>

¹³GeoSampa - <http://geosampa.prefeitura.sp.gov.br/>

¹⁴API Olho Vivo - <http://www.sptrans.com.br/desenvolvedores/APIOlhoVivo.aspx>

informações sobre o transporte público na cidade.

1.5. Plataformas de Cidades Inteligentes

Nesta seção serão apresentadas diversos projetos de pesquisa para o desenvolvimento de plataformas de software para cidades inteligentes que utilizam as tecnologias descritas na seção 1.4. O objetivo da maioria dessas plataformas é oferecer de forma integrada diversos requisitos funcionais e não-funcionais para facilitar a implementação de aplicações de cidades inteligentes.

1.5.1. OpenIoT

O OpenIoT é uma plataforma para suportar a criação de aplicações baseadas na Internet das Coisas, essa plataforma é utilizada no projeto Vital (Petrolo et al. 2014) para a implantação de um ambiente de Cidades Inteligentes em diversas cidades da Europa como Londres, Turim e Madrid. A Figura 1.6 apresenta uma visão geral da arquitetura dessa plataforma, a qual possui três planos: o **Plano Físico**, o *Plano Virtualizado* e o *Plano de Utilidades e Aplicações*.

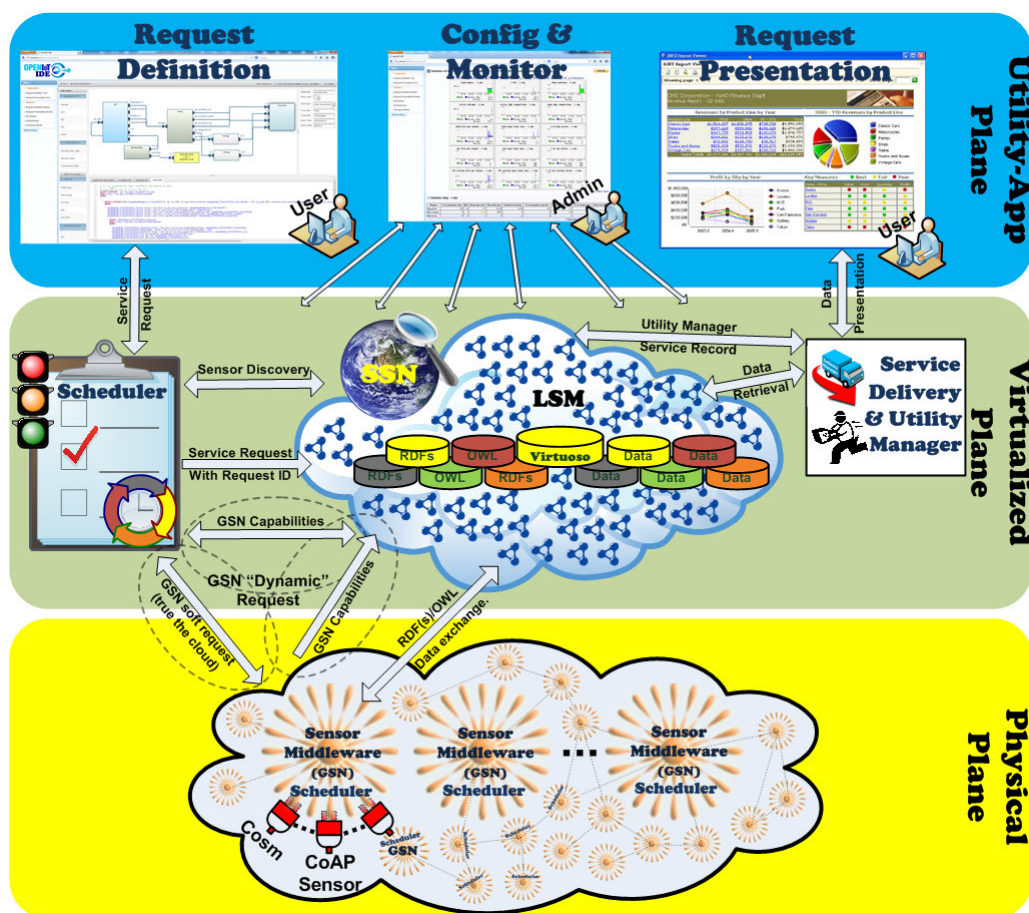


Figura 1.6. Plataforma OpenIoT (Petrolo et al. 2014)

O Plano Físico é um middleware responsável por coletar, filtrar, agregar e limpar os dados de sensores, atuadores e diversos outros tipos de dispositivos. Ele age como

uma interface entre o mundo físico e a plataforma OpenIoT. A versão atual do OpenIoT usa o X-GSN (Calbimonte et al. 2014), um middleware de código aberto para gerenciar, controlar e monitorar dispositivos IoT.

O Plano Virtualizado tem o objetivo de armazenar os dados, executar serviços e agendar a execução desses serviços. Os principais componentes dessa camada são os seguintes:

- O **Agendador (Scheduler)** que recebe requisições por serviços e garante o acesso aos recursos que esses serviços necessitam como fluxos de dados e tempo de processamento. É também responsável por identificar os sensores necessários para a execução dos serviços.
- O **Armazenamento de Dados na Nuvem (Cloud Data Storage)** armazena todos os dados da plataforma, i.e., os dados coletados da rede de sensores da cidade, de configuração da plataforma e das aplicações que são executadas na plataforma. Para o armazenamento e processamento dos dados capturados na cidade é utilizado o *Linked Sensor Middleware*, que possibilita o armazenamento e processamento de dados utilizando ontologias com o formato padrão RDF no banco de dados Virtuoso¹⁵ (Le-Phuoc et al. 2012).
- O **Gerenciador de Serviços e Utilidades (Service Delivery and Utility Manager)** tem quatro funções principais: possibilitar a definição de serviços sobre a plataforma, executar os serviços requisitados por usuários e aplicações, permitir a definição de parâmetros de configuração da plataforma e fazer o monitoramento de toda a infraestrutura que é executada na plataforma. Adicionalmente, esse componente mantém um histórico de todos os serviços utilizados na plataforma para permitir a cobrança pelo seu uso.

A Camada de Utilidades e Aplicações é a interface da plataforma com o usuário e possui os três componentes seguintes:

- A **Definição de Requisições (Request Definition)** permite que usuários definam novas aplicações usando os serviços e os dados que estão disponíveis na plataforma, permitindo inclusive a composição de serviços.
- A **Apresentação de Requisições (Request Presentation)** executa uma aplicação criada no componente Definição de Requisições. Quando uma aplicação é executada, ela se comunica com os componentes Gerenciador de Serviços e Utilidades para recuperar os resultados da execução dos serviços na plataforma.
- A **Configuração e Monitoramento (Configuration and Monitoring)** permite a configuração dos parâmetros da plataforma como, por exemplo, intervalos para a leitura de sensores, prioridade de serviços e aplicações e permissões de usuários e o monitoramento da execução de todos os componentes e dispositivos da plataforma.

¹⁵Virtuoso - <https://github.com/openlink/virtuoso-opensource>

Utilizando a plataforma OpenIoT, pesquisadores desenvolveram um sistema para o gerenciamento de lixo de uma cidade (Anagnostopoulos et al. 2015). Nessa aplicação, existem sensores dentro das cestas de lixo indicando se ela está cheia ou vazia. As cestas podem ser priorizadas para o recolhimento do lixo dependendo de sua localização, por exemplo, cestas próximas a escolas ou hospitais. Além disso, o sistema calcula a quantidade de lixo coletado e a quantidade de quilômetros rodados pelos caminhões de lixo para possibilitar a análise dos dados.

O OpenIoT é uma plataforma bastante completa atendendo à maioria dos requisitos necessários para a criação de uma cidade inteligente. Os pontos fortes dessa plataforma são o middleware para o armazenamento dos dados coletados na cidade, suas ferramentas para a definição dos serviços e o fato da plataforma ser de software livre. Entretanto, a plataforma não oferece coleta de dados de outras fontes importantes como redes sociais e não oferece suporte para o pré-processamento dos dados, o que é bastante relevante quando a quantidade de dados é muito grande.

1.5.2. SmartSantander

SmartSantander é uma plataforma experimental para o desenvolvimento de aplicações e serviços para Cidades Inteligentes. Ela é composta por um grande número de dispositivos IoT implantados em diversos cenários urbanos que coletam diferentes tipos de dados, uma rede de computadores, chamados *Gateways* que gerenciam e monitoram esses dispositivos e Servidores, que armazenam e processam os dados coletados na cidade. A arquitetura da plataforma SmartSantander é formada pelas três camadas seguintes:

- **IoT Nodes** são os nós físicos implantados na cidade. A maioria desses dispositivos são de baixa capacidade de processamento e na maioria dos casos apenas coletam algum dado da cidade. Por estarem implantados no ambiente urbano, esses dispositivos estão sujeitos a falhas e vandalismo, por isso é necessário a monitoração e manutenção dos dispositivos.
- **IoT Gateways** responsáveis por conectar os dispositivos espalhados pela cidade aos servidores nos quais os dados serão armazenados e processados. Essa camada também é responsável pela monitoração dos dispositivos que estão conectados a cada *gateway*. Eles são responsáveis pela gestão desses dispositivos e podem reconfigurar os dispositivos automaticamente e em tempo de execução.
- **Servers** nos quais os dados serão armazenados e processados. Essa camada é composta por hardware de alto poder computacional para garantir a escalabilidade e elasticidade da plataforma. Esses servidores podem servir como repositórios de dados, servidores de aplicações e serviços para a mineração e processamento dos dados.

A Figura 1.7 mostra um exemplo da arquitetura do SmartSantander sendo utilizada por uma aplicação para o sensoriamento participativo na cidade. Na camada do servidor existem componentes para fazer o diretório dos dispositivos que existem na cidade, um subsistema de apoio a aplicações que é responsável pelo armazenamento e acesso aos dados e o componente específico da aplicação.

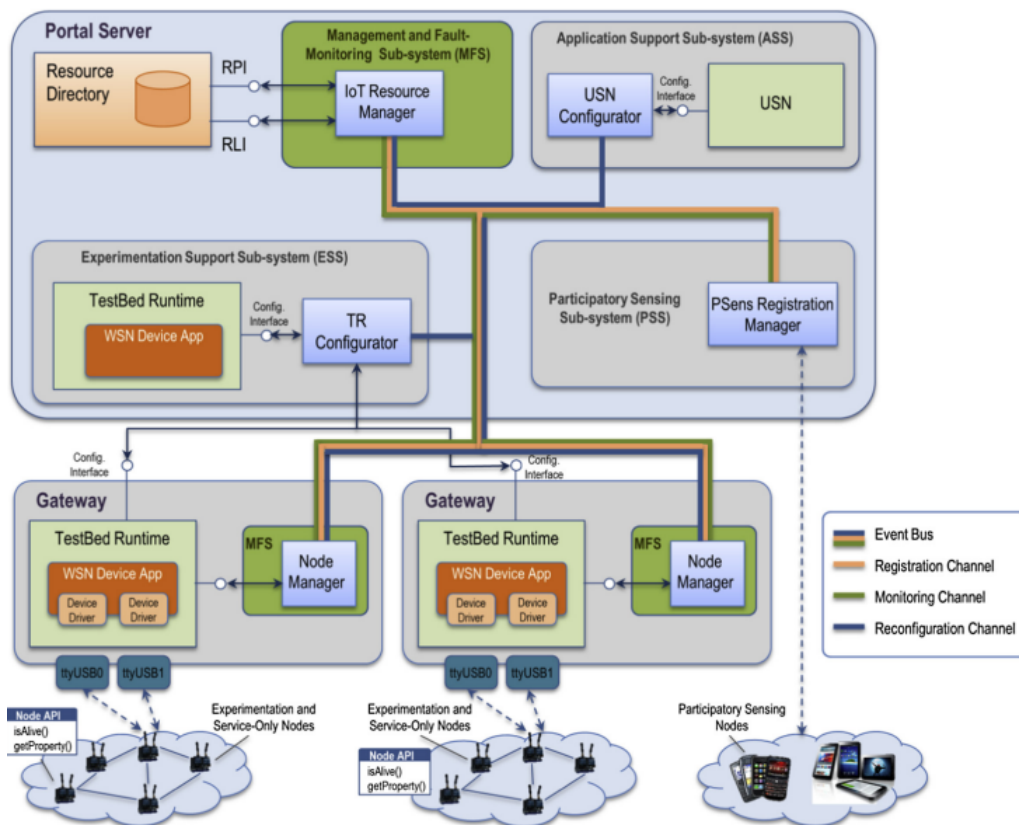


Figura 1.7. Arquitetura da Plataforma SmartSantander (Sanchez et al. 2014)

A Figura 1.7 mostra também que podem existir diversos *gateways* que se conectam a um conjunto de nós espalhados pela cidade. Os *gateways* recebem os dados dos dispositivos e os enviam para o servidor e também monitoram os dispositivos.

A plataforma SmartSantander tem como pontos fortes (1) a coleta de dados de uma grande rede de sensores da cidade de Santander, que mostra que é possível suportar o grande fluxo de dados em uma plataforma de Cidade Inteligentes, (2) a monitoração dos dispositivos espalhados pela cidade, que possibilita a fácil solução de problemas como dispositivos que perdem comunicação com a plataforma ou quebrados e (3) a possibilidade do uso da plataforma para o desenvolvimento de várias aplicações para a cidade.

Usando a infraestrutura do SmartSantander foram desenvolvidos diversos outros projetos, como por exemplo o SEN2SOC (Vakali et al. 2014) que captura fluxos de dados dos sensores da cidade e de redes sociais dos cidadãos para criar aplicações. Dois exemplos de aplicações são a reação dos cidadãos a algum evento na cidade e a construção de mapas de calor com os dados sobre a poluição do ar na cidade. Outro projeto é o CiDAP (Cheng et al. 2015) que será apresentado a seguir.

1.5.3. CiDAP

A plataforma CiDAP (*City Data and Analytics Platform*) utiliza ferramentas de Big Data com o objetivo de processar o grande volume de dados coletados da cidade para adicionar

inteligência e contexto nas aplicações e serviços desenvolvidos para a cidade. Os dados processados pela plataforma são coletados por um Middleware IoT independente. A plataforma foi testada utilizando os dados do SmartSantander (Cheng et al. 2015). A Figura 1.8 apresenta os cinco principais componentes da sua arquitetura:

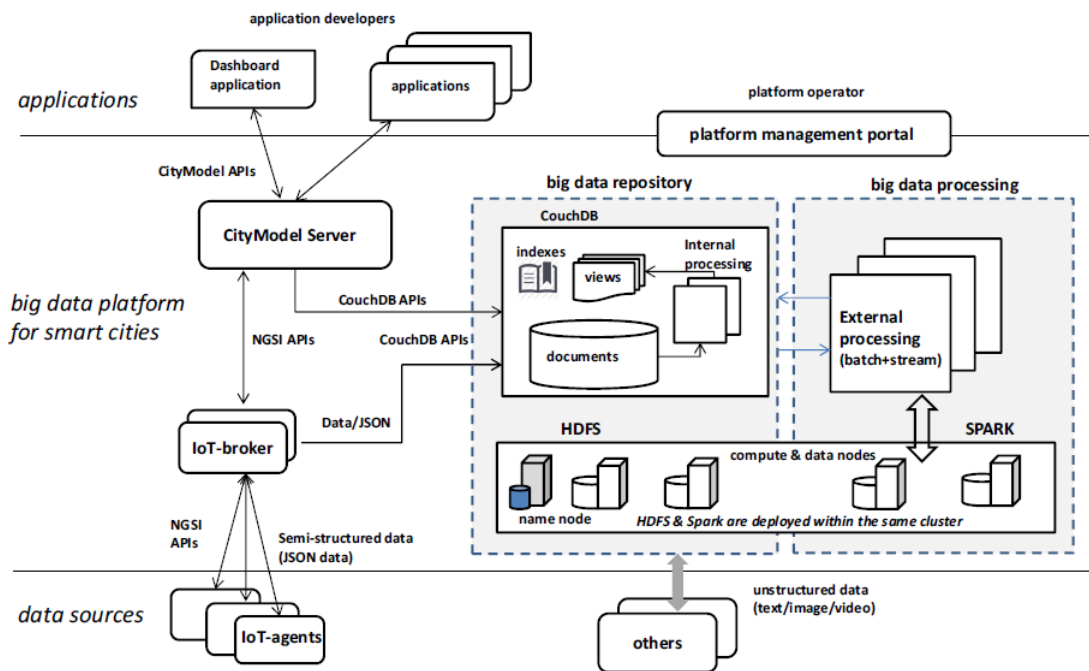


Figura 1.8. Plataforma CiDAP (Cheng et al. 2015)

- Os **IoT-Agents** se conectam com o middleware IoT como um *gateway* para coletar os dados dos dispositivos e armazenar na plataforma. Cada fonte de dados disponível no middleware IoT é mapeado em um IoT Agent.
- Os **IoT-Brokers** agem como uma interface unificada para os *IoT Agents*, facilitando o acesso aos dados coletados pelo middleware. Esse componente se comunica com o Repositório de Big Data para enviar os dados que serão armazenados ou com o CityModel para serem utilizados diretamente nas aplicações.
- O **Big Data Repository** armazena os dados coletados da cidade e também os dados processados utilizando o componente de processamento de Big Data. A plataforma utiliza o banco de dados NoSQL CouchDB ¹⁶, o qual armazena os dados em documentos no formato JSON. Esse componente possui também uma ferramenta de processamento interno para fazer processamentos simples e rápidos nos dados como a transformação dos dados em novos formatos ou a criação de novas tabelas e visões.
- O **Big Data Processing** é responsável por processamentos complexos ou demorados como a agregação dos dados ou algoritmos de aprendizado de máquina usando

¹⁶<http://couchdb.apache.org>

os dados armazenados no repositório de Big Data. Além disso, esse componente também processa dados históricos utilizando processamento em lote ou processa os dados em tempo real utilizando fluxos de dados. Esse componente é implementado utilizando a ferramenta Apache Spark (Zaharia et al. 2010).

- O **City Model Server** é a interface da plataforma para as aplicações externas. Ele é implementado como uma API com uma interface para os principais dados da cidade, como dados de sensores e resultados de análise dos dados. Esse componente possibilita dois tipos de consultas aos dados da plataforma, uma utilizando uma API REST na qual é possível efetuar consultas aos últimos valores coletados por um sensor ou outro dispositivo ou a resultados de algum processamento. A outra forma de consulta é a assinatura a uma fonte de dados, assim, a aplicação recebe periodicamente dados dessa fonte através de um mecanismo de publicação/assinatura.

A plataforma CiDAP tem como objetivo armazenar e processar um grande volume de dados urbanos. Isso é importante porque a quantidade de dados coletados em uma cidade inteligente será muito grande. Os pontos fortes dessa arquitetura são a preocupação com a escalabilidade e elasticidade das estruturas de armazenamento e processamento. Além de disponibilizar ferramentas para o processamento de dados históricos e em tempo real, outro fator interessante é que a plataforma já foi efetivamente testada utilizando os dados reais do SmartSantander.

1.5.4. Arquitetura Baseada em Computação em Nuvem e Big Data para Cidades Inteligentes

Pesquisadores de três universidades inglesas propuseram uma arquitetura baseada em Computação em Nuvem e Big Data para a construção de uma plataforma de cidades inteligentes com o objetivo de analisar todos os dados coletados de uma cidade (Khan et al. 2015). Além da arquitetura, no projeto também foi feita uma implementação experimental da plataforma utilizando apenas ferramentas de código aberto. A Figura 1.9 apresenta as camadas e componentes dessa arquitetura.

A arquitetura é composta por três camadas, a camada inferior consiste em repositórios distribuídos e heterogêneos e diversos sensores que estão conectados à plataforma. O objetivo dessa camada é coletar, limpar e classificar os dados utilizando padrões abertos para a representação de dados como XML e JSON e algoritmos de aprendizado de máquina. Na implementação experimental foi utilizado o banco de dados Cassandra¹⁷ para a implementação dessa camada.

A camada intermediária apoia o desenvolvimento de *workflows* para o desenvolvimento de serviços utilizando os repositórios de dados. Entretanto, os dados coletados podem estar em diferentes formatos e sem nenhuma relação entre eles, por isso, existe um componente para o entendimento e a ligação dos dados utilizando ferramentas de Web Semântica. Para a implementação experimental foi utilizado o banco de dados Virtuoso, que possibilita a criação de banco de dados semânticos com RDF (*Resource Description Framework*).

¹⁷Apache Cassandra - <http://cassandra.apache.org>

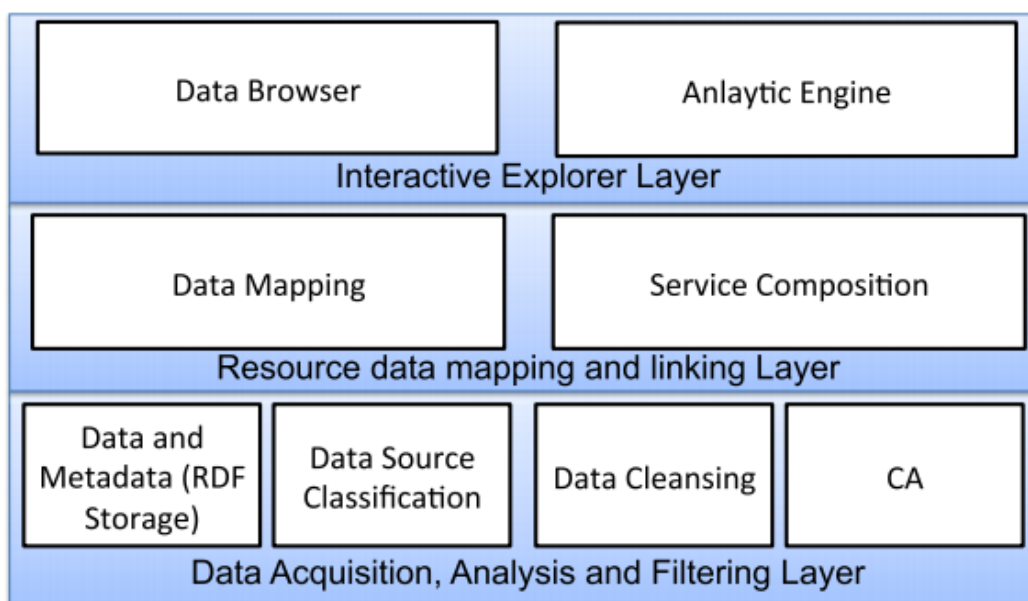


Figura 1.9. Arquitetura Baseada em Computação em Nuvem e Big Data para Cidades Inteligentes

Na camada superior, existe um motor de análises que utiliza os serviços desenvolvidos com os *workflows* e os dados ligados para fazer processamentos que sejam necessários para as aplicações desenvolvidas sobre as plataformas. Nessa camada, podem ser utilizadas ferramentas que processam uma grande quantidade de dados com algoritmos de aprendizado de máquina para entender o comportamento da cidade. Na implementação experimental da plataforma, foi utilizado o Apache Spark (Zaharia et al. 2010).

Para testar a implementação experimental da plataforma, foram utilizados os dados da cidade de Bristol, na Inglaterra, que possui um portal de dados abertos¹⁸. Para esse teste, foram coletados diversos conjuntos de dados da cidade e, utilizando a plataforma, esses dados foram relacionados e algumas aplicações foram desenvolvidas.

A plataforma tem como pontos fortes o suporte a vários requisitos não-funcionais de cidades inteligentes como escalabilidade e elasticidade, a possibilidade da ligação entre os dados utilizando ferramentas de Web Semântica e o uso de diversas ferramentas prontas e de código livre em sua implementação como o Apache Spark e o Apache Cassandra.

1.5.5. Concinnity: Uma plataforma genérica para aplicações de Big Data

A plataforma Concinnity (Wu et al. 2014) tem o objetivo de facilitar a construção de aplicações utilizando dados de sensores, contribuição coletiva e compartilhamento de dados e serviços. Para isso, a plataforma disponibiliza os dados coletados de sensores para aplicações através de uma interface de acesso aos dados da plataforma disponibilizados em um ambiente de computação em nuvem e em uma ferramenta para o desenvolvimento de *workFlows*. A Figura 1.10 apresenta as camadas e componentes da arquitetura da plataforma Concinnity.

¹⁸Open Data Bristol - <https://opendata.bristol.gov.uk>

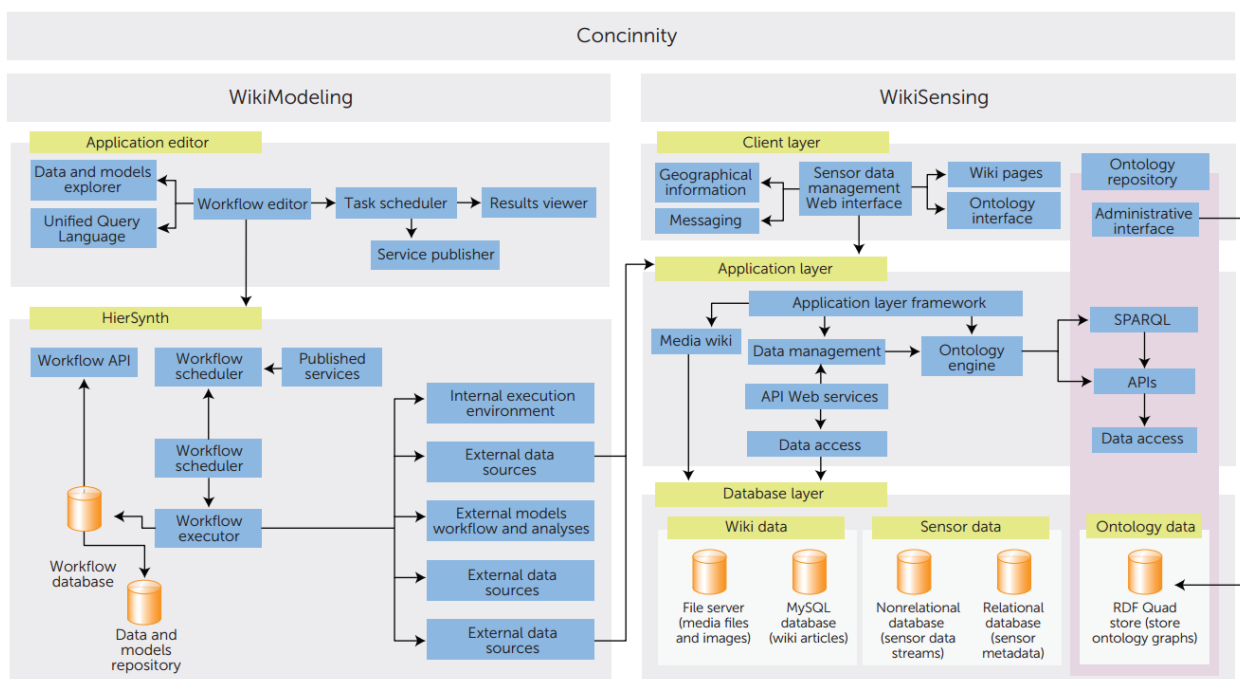


Figura 1.10. Plataforma Concinnity

A plataforma é formada por cinco camadas. *Application Editor* disponibiliza um editor de aplicações que facilita o uso dos recursos da plataforma para a construção de novas aplicações. *HierSynth* é um servidor para a execução de *workflows*. *Data Base Layer* é responsável por disponibilizar um ambiente elástico para o armazenamento de uma grande quantidade de dados. *Application Layer* é responsável pela execução das aplicações da plataforma. Finalmente, *Client Layer* é a interface da plataforma com os usuários finais.

A **Application Editor** é uma ferramenta colaborativa para o desenvolvimento rápido de aplicações que utilizam os dados disponíveis na plataforma. Essa ferramenta possui uma linguagem de consulta que possibilita a recuperação, filtragem e composição dos dados, um editor de *workflows* para a construção de fluxos de processamento dos dados, um agendador de tarefas para programar a execução das aplicações, um visualizador dos resultados para o teste das aplicações e, finalmente, um publicador de serviços, que disponibiliza as aplicações desenvolvidas para todos os usuários da plataforma.

O **HierSynth** é um servidor disponível em um ambiente de Computação em Nuvem para a execução dos serviços necessários para a execução das aplicações que foram desenvolvidos no editor de aplicações.

A **Data Base Layer** é o repositório de dados da plataforma, no qual são armazenados os dados dos sensores, aplicações e da plataforma em si. Na sua implementação, foram utilizados o banco de dados MongoDB¹⁹, que é um banco de dados NoSQL sem esquema, o que permite a flexibilização do modelo de dados da plataforma facilitando a

¹⁹MongoDB - www.mongodb.org

agregação dos dados, e o banco de dados relacional MySQL²⁰ para o armazenamento dos dados das aplicações e de configuração da plataforma. A plataforma usa ainda ontologias para consultas mais sofisticadas utilizando a linguagem SPARQL.

A **Application Layer** contém a lógica de negócios para a execução dos serviços, para o controle de acesso da plataforma, para a gestão dos dados e para permitir a colaboração dentro da plataforma. Essa camada também faz toda a comunicação com os serviços de execução de *workflows* e para acesso aos dados, mantendo a modularidade da plataforma.

A **Client Layer** disponibiliza uma interface Web para a visualização dos dados dos sensores, um catálogo das aplicações e serviços disponíveis na plataforma e uma aplicação baseada em uma Wiki para a contribuição e compartilhamento de conhecimento entre os usuários da plataforma.

Um estudo de caso utilizando a plataforma em cidades inteligente foi realizado utilizando dados de 140 sensores de qualidade do ar da cidade de Londres. A partir dos dados dos sensores e utilizando informações sobre o trânsito da cidade foi feita uma análise sobre o impacto das emissões de poluentes de veículos sobre as áreas mais poluídas da cidade. Também foram utilizados algoritmos de aprendizado de máquina para verificar a confiabilidade dos dados.

A plataforma tem como pontos fortes o suporte para a implementação de aplicações dentro da plataforma, contribuindo para a colaboração entre seus usuários, a utilização de tecnologias como bancos de dados NoSQL e Computação e Nuvem para permitir a escalabilidade e elasticidade da plataforma e o uso de várias representações dos dados para facilitar seu uso.

1.5.6. ClouT - Cloud of Things for empowering the citizen in smart cities

A projeto nipo-europeu ClouT (Tei and Gurgen 2014) tem o objetivo de integrar a Computação em Nuvem e a Internet das Coisas para possibilitar, através de serviços digitais, a criação de cidades inteligentes explorando múltiplas fontes de dados. A plataforma disponibiliza diversos requisitos funcionais para facilitar o desenvolvimento de aplicações como a coleta, integração, armazenamento e processamento dos dados da cidade. A Figura 1.11 apresenta a arquitetura geral da plataforma.

A plataforma é composta por duas camadas, a ClaaS (*City Infrastructure as a Service*) e a CPaaS (*City Platform as a Service*). Essas camadas disponibilizam seus recursos através de serviços em um ambiente de Computação em Nuvem.

A primeira camada, ClaaS, gerencia um conjunto de recursos físicos garantindo a interoperabilidade entre eles utilizando tecnologias de virtualização. Todos os recursos dessa camada são acessados através de APIs, permitindo o acesso a qualquer recurso físico da cidade inteligente. Essa camada possui os cinco componentes principais seguintes:

- O componente **Sensorization and Actuatorization** é responsável pela comunicação da plataforma com diversas fontes de dados da cidade e dos usuários para a co-

²⁰MySQL - <https://www.mysql.com>

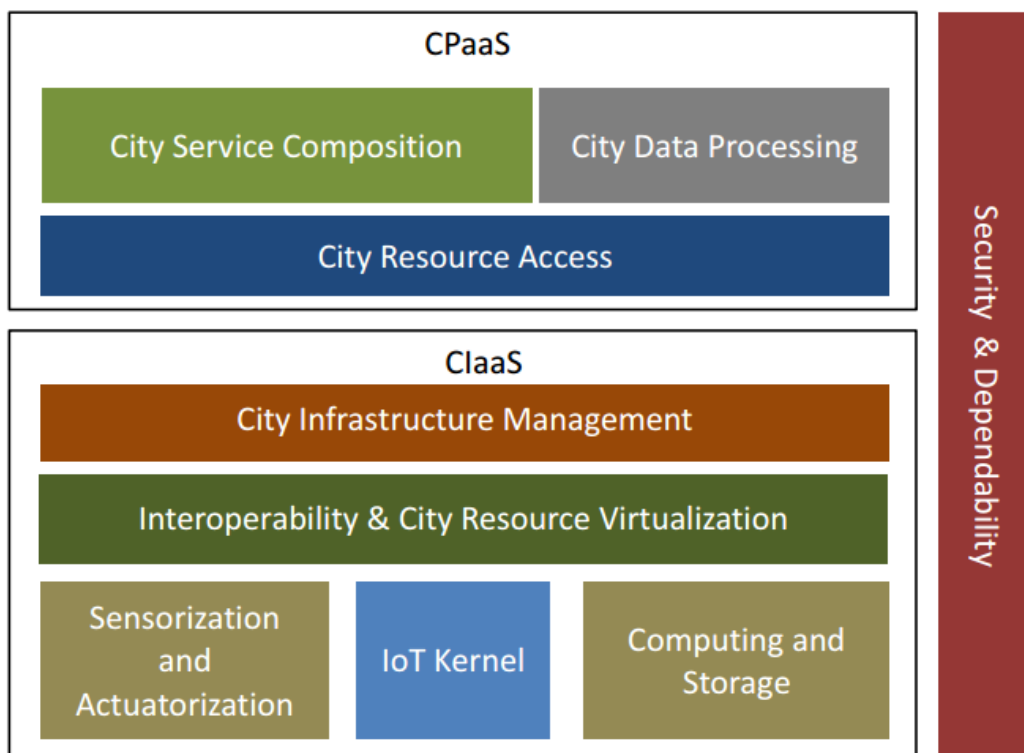


Figura 1.11. Plataforma ClouT

leta dos dados, como aplicações legadas, *smartphones* e redes sociais, transformando-os em sensores e atuadores integrados à plataforma.

- O **IoT Kernel** gerencia os dispositivos IoT da cidade, permitindo a exclusão, adição, edição e monitoramento dos dispositivos.
- O **Computing and Storage** inclui todo o software e hardware necessários para oferecer um infraestrutura escalável e confiável para armazenamento de dados e serviços e a hospedagem de aplicações.
- O **Interoperability and City Resource Virtualization** é responsável por validar, converter e relacionar os dados coletados na plataforma. Essa camada torna os serviços da plataforma transparente, unificando todo o acesso aos dados e serviços a partir de um modelo único da cidade.
- O **City Infrastructure Management** oferece funcionalidades centralizadas para a busca e descoberta de recursos na cidade bem como a gestão de eventos por eles gerados.

A segunda camada, a CPaaS, possui um conjunto de serviços que possibilitam o desenvolvimento e a composição de aplicações para as cidades. Os componentes dessa camada são:

- **City Resource Access** é o middleware que possibilita armazenamento e recuperação de dados e meta-dados sobre os recursos da cidade.

- **City Data Processing** possibilita a análise e extração de informações dos dados que estão armazenados na plataforma oferecendo um motor de inferências e um componente para a filtragem de dados inconsistentes.
- O **City Service Composition** oferece um conjunto de ferramentas para facilitar a integração e composição de dados e serviços desenvolvidos para as aplicações da plataforma. Essa ferramenta oferece uma interface gráfica para possibilitar a composição de serviços.

Acima dessas duas camadas estão as aplicações que podem ser desenvolvidas a partir do uso dos serviços oferecidos pela camada CPaaS. Vários exemplos de aplicações estão sendo desenvolvidas com a plataforma ClouT (Galache et al. 2014). Em Fujiwasa, no Japão, pesquisadores estão desenvolvendo um sistema para alertas de tsunamis analisando dados ambientais capturados por sensores e postados por cidadãos em redes sociais. Em Gênova, na Itália, já existe uma aplicação que gerencia e coleta dados de uma rede de sensores implantada na cidade, também com o objetivo de alertar a população sobre riscos ambientais ²¹.

Essa plataforma tem como ponto forte a disponibilização de todos os seus serviços em um ambiente de Computação em Nuvem, permitindo inclusive que outras plataformas ou aplicações acessem apenas componentes específicos da plataforma. Outra característica importante é a camada que unifica a visão sobre a infraestrutura da cidade utilizando um modelo único, o que facilita o entendimento da cidade e o desenvolvimento e integração das aplicações e serviços.

1.5.7. Arquitetura para Coleta de Fluxos de Dados em Tempo Real

O projeto austríaco *Analysis on semantically structured data from smart buildings and smart grids* propõe uma plataforma com o objetivo de processar fluxos de dados em tempo real em um ambiente de Computação em Nuvem para Cidades Inteligentes (Girtelschmid et al. 2013). Para facilitar a integração entre diferentes aplicações e dispositivos da cidade, a plataforma utiliza um modelo semântico baseado em ontologias. Como o processamento desses modelos tem um desempenho ruim, a arquitetura propõe o uso de ferramentas de Big Data para o processamento paralelo e distribuído de grandes fluxos de dados. Em uma implementação experimental foi utilizada a ferramenta de Big Data Apache Storm²² para fazer esse processamento. A Figura 1.12 apresenta a arquitetura dessa plataforma.

Os dados coletados de uma rede de sensores implantados na cidade são capturados em tempo real pela plataforma, que pode estar implantada em um *cluster* de vários computadores. A primeira camada da plataforma, formada pelo processador de fluxos de dados, é responsável pelo pré-processamento dos dados, como detectar mudanças bruscas nas leituras de sensores, detectar falhas ou *outliers* e também por pequenos processamentos como acumular ou calcular a média de leituras de sensores. Essa camada pode ainda acessar diretamente o repositório de ontologias e fazer mudanças nos modelos.

O repositório de ontologias armazena a leitura dos dados de sensores mais recentes e valores acumulados dessas leituras, como a média de temperatura e a quantidade de

²¹Io Non Rischio Gênova - <http://www.iononrischioclout.comune.genova.it>

²²Apache Storm - <http://storm.apache.org/>

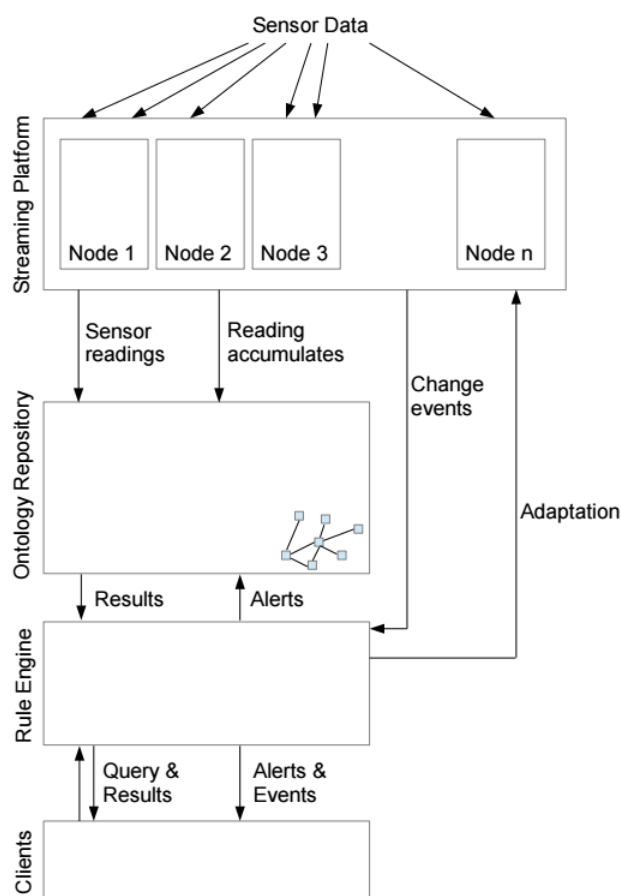


Figura 1.12. Arquitetura para Coleta de Fluxos de Dados em Tempo Real

chuvas acumulada em um determinado período. A ontologia contém ainda o modelo dos dados da cidade e facilita a execução das regras que processam os dados da plataforma.

O componente **Rule Engine** executa uma série de regras para o processamento dos dados armazenados nas ontologias sempre que uma requisição de uma aplicação cliente é feita para a plataforma. Os clientes podem acessar a plataforma de duas maneiras: consultas simples aos dados da cidade, como a última leitura de um sensor e a assinatura (*subscription*) a uma regra que sempre que for processada envia os dados a todos os clientes que realizaram a assinatura. Esse componente pode também adaptar o funcionamento do *cluster* que coleta os dados dos sensores como interromper a leitura de um sensor que aparenta estar com defeito e mudar a frequência da leitura de um sensor.

Por fim, as aplicações podem se registrar ou consultar os dados armazenados na plataforma. As aplicações podem ainda adicionar novas regras a plataforma ou adaptá-las para atender requisitos específicos da aplicação. A plataforma foi testada utilizando dados artificiais gerados por um simulador de Cidades Inteligentes que gerou dados sobre o consumo de energia elétrica em Casas Inteligentes.

Essa plataforma tem como ponto forte a captura e processamento dos fluxos de dados da cidade em tempo real utilizando uma ferramenta de grande poder computacional. Além disso, a plataforma permite a integração dos dados utilizando um modelo de

ontologias, o que facilita o desenvolvimento de aplicações que utilizam diversas fontes de dados.

1.5.8. Outras Plataformas

O projeto Padova Smart City (Zanella et al. 2014) utiliza uma plataforma baseada na Internet das Coisas para monitorar e coletar dados de uma rede de sensores com mais de 300 dispositivos instalados na cidade. Essa rede tem duas funções principais, coletar dados ambientais como emissões de CO_2 e temperatura do ar e monitorar a infraestrutura da cidade como postes de luz. Um requisito importante para a implementação de cidades inteligentes destacados por esse trabalho é o uso de protocolos e formatos de dados comuns para facilitar a interoperabilidade entre os diferentes sistemas da cidade.

Foram desenvolvidas duas aplicações utilizando a plataforma Padova Smart City. A primeira (Bui and Zorzi 2011) é um sistema de saúde, que tem como principal função monitorar a condição de pacientes e enviar seus dados diretamente para médicos, e chamar automaticamente serviços de saúde em caso de emergência. A segunda (Bressan et al. 2010) apresenta uma aplicação para monitorar todos os postes de luz da cidade, indicando onde é necessário fazer a manutenção e a troca de lâmpadas.

O Gambas é um middleware para o desenvolvimento de aplicações para Cidades Inteligentes (Apolinarski et al. 2014) que suporta a aquisição, distribuição e integração dos dados da cidade. A plataforma disponibiliza ainda um conjunto de ferramentas e um ambiente de execução de aplicações para facilitar o desenvolvimento de aplicações para a cidade. Outro requisito importante tratado pelo middleware é a sensibilidade ao contexto para possibilitar que as aplicações se adaptem a situação, comportamento e intenção do usuário. Toda a comunicação entre as aplicações e a plataforma são criptografadas para garantir a segurança e privacidade dos cidadãos.

O middleware Gambas foi usado para o desenvolvimento de duas aplicações utilizadas no transporte público de Madrid. A primeira é uma ferramenta que utiliza informações contextuais do usuário (localização e horário de um compromisso) para encontrar as melhores opções para sua viagem no transporte público da cidade (Foell et al. 2014). A segunda estima o número de usuários que estão dentro de um ônibus através de seus smartphones e os informa para a plataforma, possibilitando a análise dos horários que cada linha de ônibus é mais utilizada na cidade (Handte et al. 2014).

Scallop4SC (*SCALable LOGging Platform for Smart City*) (Takahashi et al. 2012; Yamamoto et al. 2014) é uma plataforma que emprega ferramentas de Big Data para processar um grande volume de dados coletados de Prédios Inteligentes. A plataforma utiliza diversos dados sobre esses prédios como consumo de água e energia elétrica, temperatura, umidade do ar e a quantidade de lixo gerado. Periodicamente, os prédios enviam os dados para a plataforma, para que eles possam ser processados e análises e serviços possam ser criados ou feitos sobre esses dados. Para essas análises é utilizado o algoritmo MapReduce do Apache Hadoop.

Duas aplicações foram desenvolvidas utilizando a plataforma Scallop4SC no domínio de gerência de energia elétrica. A primeira é uma ferramenta para a visualização do consumo de energia dos prédios conectados à plataforma, possibilitando análises em

diferentes níveis como ruas, bairros, região da cidade e a cidade inteira. O segundo é um serviço de alerta para detectar desperdício de energia, disponível para Casas Inteligentes (Yamamoto et al. 2014).

A Plataforma Europeia para Cidades Inteligentes (*European Platform for Intelligent Cities - EPIC*) (Ballon et al. 2011) implementa um middleware IoT completo para facilitar o uso e gerenciamento de uma rede de sensores em um Cidade Inteligente. Esse middleware tem o objetivo de lidar com diversos requisitos não-funcionais desse tipo de plataforma como interoperabilidade, heterogeneidade, escalabilidade, extensibilidade e configurabilidade.

Com essa plataforma foram desenvolvidas diversas aplicações, como uma para a chegada de novos moradores na cidade de Bruxelas, na Bélgica, que procura moradias que satisfaçam alguns requisitos das pessoas como preço e proximidade ao transporte público. Essa aplicação integra dados públicos sobre pontos de interesse e dados privados obtidos de aplicações para a venda e aluguel de imóveis na cidade. Outra aplicação é uma para o monitoramento do consumo de energia elétrica para edifícios públicos, testada em Manchester, na Inglaterra, essa aplicação testa a coleta de dados na plataforma e tem o objetivo de melhorar o consumo de energia elétrica, diminuindo o desperdício.

1.6. Requisitos de uma Plataforma de Cidades Inteligentes

A partir da análise das iniciativas introduzidas na Seção 1.4 e das plataformas descritas na Seção 1.5, apresentamos agora os principais requisitos que devem ser considerados na implementação de uma plataforma de Cidades Inteligentes.

1.6.1. Requisitos Funcionais

O principal objetivo de uma plataforma para cidades inteligente é facilitar o desenvolvimento de aplicativos para a cidade. Assim, a maioria das plataformas apresentadas oferecem funcionalidades para a coleta, armazenamento e compartilhamento dos dados urbanos e para o desenvolvimento e execução de serviços e aplicações para a cidade. Os principais requisitos funcionais para plataformas de software para cidades inteligentes são os seguintes.

- **Gerenciamento de Dados:** Uma Cidade Inteligente manipula uma quantidade enorme de dados, por isso, é necessário que as plataformas implementem diversas atividades relacionadas ao ciclo de vida dos dados da cidade, como a **coleta**, o **armazenamento**, a **análise** e a **visualização** dos dados. Diversas técnicas e ferramentas podem ser usadas para suprir esse requisito, como por exemplo bancos de dados NoSQL para dados não-estruturados ou semi-estruturados, bancos de dados relacionais para dados estruturados, ferramentas de *Big Data* para a análise e processamento dos dados e geradores de relatórios e imagens para a visualização dos dados (Hernández-Muñoz et al. 2011; Cheng et al. 2015).
- **Ambiente para Execução de Aplicações:** Algumas plataformas oferecem suporte para a **execução** de aplicações da cidade facilitando a **implantação** e a **integração** entre essas aplicações. Algumas plataformas oferecem um ambiente para a implantação de serviços e aplicações (Apolinarski et al. 2014); outras oferecem um

serviço para a execução de aplicações desenvolvidas com ferramentas da própria plataforma (Petrolo et al. 2014; Wu et al. 2014).

- **Gerencia da Rede de Sensores:** Uma das principais características de Cidades Inteligentes é a necessidade de gerenciar uma grande rede de dispositivos instalados na cidades como sensores que coletam dados do ambiente, sensores que verificam o funcionamento do mobiliário urbano e sensores que monitoram o trânsito. Essa rede pode ser complexa e grande, como por exemplo a rede do projeto SmartSantander que conta com mais de 20 mil sensores em uma cidade pequena; numa cidade grande como São Paulo ou Rio de Janeiro, idealmente deveríamos ter de centenas de milhares a milhões de sensores. Algumas das atividades necessárias nesse requisito são a **adição, remoção, monitoramento e coleta de dados dos sensores**. Além de Santander, outras cidades que já possuem uma rede de sensores razoavelmente explorada são Barcelona, Padua, Chicago, Dublin e Amsterdã.
- **Processamento de Dados:** O processamento dos dados coletados na cidade é essencial para a criação de diversos serviços e aplicações para Cidades Inteligentes como para o entendimento de algum fenômeno que ocorre na cidade, para pesquisar a melhor opção de transporte para o usuário e para identificar áreas de risco. Diversas ferramentas são utilizadas para o processamento de dados em Cidades Inteligentes, como máquinas de inferência (normalmente associadas ao uso de ontologias), processadores de *workflows* (para executar aplicações nas quais os dados passam por diversos estágios de processamento, como no atendimento de saúde de um paciente, ou em uma solicitação à prefeitura) e ferramentas de Big Data para o processamento de grandes quantidades de dados. Esses componentes processam grandes conjuntos de dados com o propósito de **analisar, verificar, agregar e filtrar** os dados coletados da cidade (Girtelschmid et al. 2013; Cheng et al. 2015).
- **Acesso aos Dados:** Para possibilitar o desenvolvimento de aplicações para a cidade, é essencial que os dados coletados e processados possam ser acessados por aplicações e serviços externos à infraestrutura da cidade. Diversas iniciativas já disponibilizam seus dados através de **portais de dados abertos**, mas é fundamental também, que os dados nesses portais sejam disponibilizados em **formatos padronizados** e com **meta-dados descritivos** associados. Para isso, várias cidades utilizam **APIs bem definidas** que facilitam o desenvolvimento de aplicações automatizadas. Outra forma interessante de disponibilizar os dados, é através de **serviços de publicação/assinatura** (*publish/subscribe*), por meio dos quais clientes que manifestam interesse em tópicos específicos recebem dados sempre que uma fonte gerar novos dados de interesse para cada cliente.
- **Gerenciamento de Serviços:** A maioria das plataformas e iniciativas de Cidades Inteligentes adota a **arquitetura orientada a serviços** para oferecer as funcionalidades da plataforma em um ambiente de Computação em Nuvem. Alguns dos serviços oferecidos são: acesso aos dados coletados pelos sensores e dados processados, componentes para o processamento de dados como máquinas de inferência e algoritmos de aprendizado de máquina, componentes para a execução de *workflows* e serviços de gerencia de usuários da plataforma. Algumas plataformas possibilitam

ainda que desenvolvedores implantem serviços na plataforma e os disponibilizem para outras aplicações da cidade. É importante também possibilitar operações sobre serviços, como composição e **coreografias** (Issarny et al. 2011) para a criação de novos serviços e aplicações (Lee and Rho 2010; Piro et al. 2014) de forma **automatizada e escalável** (Leite et al. 2014).

- **Ferramentas para o Desenvolvimento de Software:** Como um dos principais objetivos de uma Cidade Inteligente é o fornecimento de aplicações e serviços que facilitem a vida do cidadão, é necessário que as plataformas facilitem o desenvolvimento desses sistemas. Para isso, devem ser fornecidas ferramentas que facilitem a utilização dos serviços fundamentais e componentes básicos da plataforma. Alguns exemplos de ferramentas utilizadas são: interfaces visuais para a descrição de aplicações utilizando as fontes de dados e serviços disponíveis na plataforma, ferramentas para a descrição de *workflows*, a utilização de ferramentas para a geração de relatórios e análise de dados e até o desenvolvimento de um kit para o desenvolvimento de aplicações (*Software Development Kit* ou SDK) com diversas ferramentas integradas (Elmangoush et al. 2013; Apolinarski et al. 2014).
- **Definição de um Modelo da Cidade:** Para melhorar o entendimento do funcionamento de diversos serviços da cidade e permitir a análise e o processamento automático desse funcionamento, é fundamental o desenvolvimento de modelos da cidade. Tais **modelos podem representar aspectos estáticos da cidade**, como o mapa da cidade com a localização das ruas e equipamentos públicos, ou **aspectos dinâmicos da cidade**, como o fluxo de veículos, as zonas de congestionamento em diferentes horas do dia e dias da semana ou a variação na utilização dos serviços de saúde da cidade ao longo do progresso de epidemias de diferentes doenças infecciosas. Além de melhorar o entendimento da cidade, os modelos também facilitam a análise automática dos dados através de algoritmos de aprendizado de máquina. Algumas plataformas utilizam esses modelos para permitir as consultas aos dados da cidade em uma linguagem de consulta própria e outras utilizam os modelos para facilitar a definição das aplicações e serviços da cidade, utilizando linguagens de descrição de processos como linguagens de definição de *workflows* e BPMN (*Business Process Modeling Notation*) (Cheng et al. 2015; Privat et al. 2014).

Baseado nos requisitos funcionais descritos acima, é possível observar que as principais atividades das plataformas são controlar o ciclo de vida dos dados da cidade:

1. coletar os dados com a rede de sensores e atuadores,
2. gerenciar os dados na plataforma,
3. processar os dados da cidade utilizando seu modelo de dados e
4. compartilhar os dados coletados e processados permitindo acesso externo a esses dados.

Essas atividades são bastante relacionadas com as tecnologias usadas para a implementação de cidades inteligentes como IoT para a implementação da rede de sensores, o

gerenciamento e processamento de dados com Big Data e o gerenciamento de serviços com Computação em Nuvem.

1.6.2. Requisitos não-funcionais

A maioria dos requisitos não-funcionais de uma cidade inteligente são relacionados ao fato de que estamos diante de enormes e heterogêneos sistemas distribuídos. Isso exige bons níveis de escalabilidade, adaptabilidade e interoperabilidade. Outros requisitos são relacionados a manipulação de dados críticos dos cidadãos e da cidade como privacidade e segurança. Os principais requisitos não-funcionais de plataformas de software para cidades inteligentes são os seguintes.

- **Interoperabilidade:** Diferentes dispositivos, sistemas, aplicações e plataformas compõem um ambiente de uma cidade inteligente e todos esses componentes devem operar de uma maneira integrada. Por exemplo, sensores de múltiplos fabricantes, sistemas implementados em diferentes linguagens de programação e em diferentes sistemas operacionais, plataformas que compartilham dados e usuários e sistemas legados que devem se comunicar com as novas aplicações da cidade, tudo isso deve trabalhar em conjunto de forma harmoniosa. Existem diversas técnicas adotadas para alcançar esses requisitos: o uso de interfaces genéricas e padrões da indústria, a aplicação de web semântica para integração dos componentes da plataforma e o uso de serviços de nomes e de descoberta de recursos baseados em ontologias para identificar diferentes dispositivos e fontes de dados (Villanueva et al. 2013; Gurgun et al. 2013).
- **Escalabilidade:** A quantidade de usuários, dados, aplicações e serviços em uma Cidade Inteligente será muito grande e tende a aumentar ao longo do tempo, com a integração de mais serviços e o aumento da população. Por exemplo, na cidade de Santander, uma cidade média, já existem mais de 20.000 sensores instalados coletando uma grande quantidade de dados da cidade; a plataforma CiDAP que coletou mais de 50 GBs de dados em três meses. Esse requisito não funcional é importante para o funcionamento de diversos requisitos funcionais como o gerenciamento da rede de sensores e atuadores, o gerenciamento de dados e o gerenciamento de serviços (Bain 2014; Takahashi et al. 2012). Basicamente, o que devemos buscar é que a qualidade do serviço oferecido pela plataforma não seja prejudicada à medida em que a escala do sistema aumenta e que os acordos de nível de serviço (SLA) sejam respeitados.
- **Elasticidade:** A quantidade de usuários que acessam os serviços de uma cidade inteligente pode variar bastante durante o dia ou, por exemplo, durante os horários de pico. Os serviços relacionados ao trânsito da cidade terão muito mais requisições na hora do *rush* do que de madrugada, ou durante um grande evento cultural, os serviços relacionados ao atendimento a turistas terão uma grande demanda. Por isso, é importante que a infraestrutura da cidade seja redimensionada dinamicamente de acordo com a necessidade para evitar a falta de recursos quando eles forem necessários e evitar o desperdício quando não mais forem utilizados. A maioria das iniciativas que disponibilizam seus serviços em um ambiente de Computação em

Nuvem (Wu et al. 2014; Khan et al. 2015; Petrolo et al. 2014) citam a necessidade desse requisito, uma vez que as tecnologias da Nuvem surgiram tendo elasticidade como um dos objetivos principais.

- **Segurança:** Usuários maliciosos podem fazer uso fraudulento dos serviços e dados disponibilizados pela plataforma. É fundamental que as plataformas implementem mecanismos específicos de segurança que ofereçam controle de acesso, criptografia, autenticação e a proteção dos dados da plataforma, da infraestrutura e das aplicações da cidade (Piro et al. 2014; Hernández-Muñoz et al. 2011; Petrolo et al. 2014).
- **Privacidade:** Um ambiente de cidade inteligente coleta e manipula diversos dados sensíveis a usuários, governos, empresas e ONGs da cidade como a localização de pessoas e suas ações, dados governamentais, registros médicos e dados sigilosos de negócios. É um desafio usar todos esses dados escondendo ou trocando informações para impossibilitar a identificação de quem está transmitindo a informação (Cardoso and Issarny 2007). Algumas das estratégias utilizadas para alcançar esse objetivo é o uso de criptografia, dispositivos para controlar o acesso aos dados da plataforma como certificados digitais e biometria bem como a anonimização dos dados (Apolinarski et al. 2014; Mylonas et al. 2015).
- **Sensibilidade ao Contexto:** Como a situação da cidade e dos cidadãos muda constantemente, muitas aplicações de cidades inteligentes podem apresentar melhores resultados usando informações contextuais. Alguns exemplos de informações sobre usuários que podem ser usadas são a localização, a atividade sendo realizada e a linguagem do dispositivo de acesso. Exemplos das informações sobre a cidade utilizadas são condições de tráfego, clima e qualidade do ar (Khan et al. 2013; Cheng et al. 2015). Alguns exemplos de aplicações que usam sensibilidade ao contexto são: mostrar a aplicação em uma língua diferente para turistas, mudar a rota de um motorista para evitar áreas congestionadas ou poluídas e definir a recomendação do uso de modo de transporte dependendo da previsão do tempo.
- **Adaptabilidade:** Ainda relacionado à sensibilidade ao contexto, muitas plataformas adaptam o seu comportamento de diferentes formas, em diferentes dimensões, baseado no contexto dos usuários ou da cidade. Alguns dos objetivos desse requisito são aumentar a tolerância a falhas, utilizar um servidor mais próximo a um usuário para atender sua requisição, decidir se um processamento será em tempo real ou em lote e adaptar dados de diferentes fontes para uma representação comum. Adaptação de dados, por exemplo, é bastante utilizada em plataformas que utilizam os conceitos de Internet das Coisas para adaptar o funcionamento das rede de sensores da cidade (Girtelschmid et al. 2013; Privat et al. 2014).
- **Extensibilidade:** A capacidade de adicionar serviços, componentes e aplicações à plataforma é importante para possibilitar a evolução da plataforma para atender a novos requisitos e funcionalidade que surgem ao longo do tempo. Para tanto é fundamental a utilização de boas práticas de Programação Orientada a Objetos e de Arquitetura de Software tais como os princípios SOLID e uma boa metodologia

ágil de desenvolvimento baseada fortemente em testes automatizados (Fox et al. 2015).

- **Configurabilidade:** Uma plataforma de cidade inteligente possui uma grande quantidade de configurações e parâmetros para adaptar o seu funcionamento a diferentes contextos em tempo de execução, por exemplo definindo limites de poluição e congestionamento e a prioridade de um serviço. Assim, é importante permitir a (re)configuração de diversas variáveis da plataforma. Pode-se utilizar um portal para centralizar as configurações da plataforma, porém, devido ao grande número de configurações necessárias, é desejável que a própria plataforma conheça o seu contexto de execução e consiga alterar dinamicamente suas configurações sem a necessidade da intervenção de um operador humano (Wan et al. 2012; Privat et al. 2014).

Observando os requisitos não-funcionais descritos nesta seção, é possível observar que eles são importantes para diversos requisitos funcionais. Por exemplo, Escalabilidade é importante para o gerenciamento da rede de sensores e dos dados da cidade Segurança e Privacidade são importantes para todos os requisitos funcionais relacionados ao gerenciamento de dados da plataforma e Configurabilidade é importante para todas as funcionalidades da plataforma. Boa parte desses requisitos funcionais são também os principais desafios técnicos e científicos para o desenvolvimento de sistemas para cidades inteligentes que serão descritos na próxima seção.

1.7. Uma Arquitetura de Referência para Plataformas de Cidades Inteligentes

A partir dos requisitos funcionais e não-funcionais levantados na seção anterior, apresentamos agora uma arquitetura de referência destacando os elementos mais importante para uma plataforma de software para cidades inteligentes e como eles se inter-relacionam. O principal objetivo desta arquitetura é facilitar a compreensão, implementação e a integração de serviços e aplicações para cidades inteligentes. A Figura 1.13 apresenta uma visão geral da arquitetura.

A camada **Nuvem e Infraestrutura de Rede** é responsável pela hospedagem e comunicação entre os dispositivos e os serviços de software implantados na cidade. O objetivo desse componente é possibilitar a integração física de todos os dispositivos que estão conectados à plataforma, incluindo servidores, sensores, atuadores e dispositivos de usuários. A Computação em Nuvem é usada como mecanismo de suporte a diversos requisitos não-funcionais essenciais na plataforma, tais como escalabilidade, elasticidade e extensibilidade.

Em um nível de abstração superior à camada de Nuvem e Rede, a arquitetura de referência prevê o **Middleware de IoT** e o **Middleware de Serviços**. O primeiro administra e faz a interface das “coisas” implantadas na cidade possibilitando uma efetiva comunicação desses dispositivos com a plataforma. O Middleware de Serviços gerencia os serviços que a plataforma irá disponibilizar para as aplicações que serão implementadas utilizando a plataforma, fornecendo funcionalidades como implantação, publicação, descoberta, monitoração e composição de serviços.

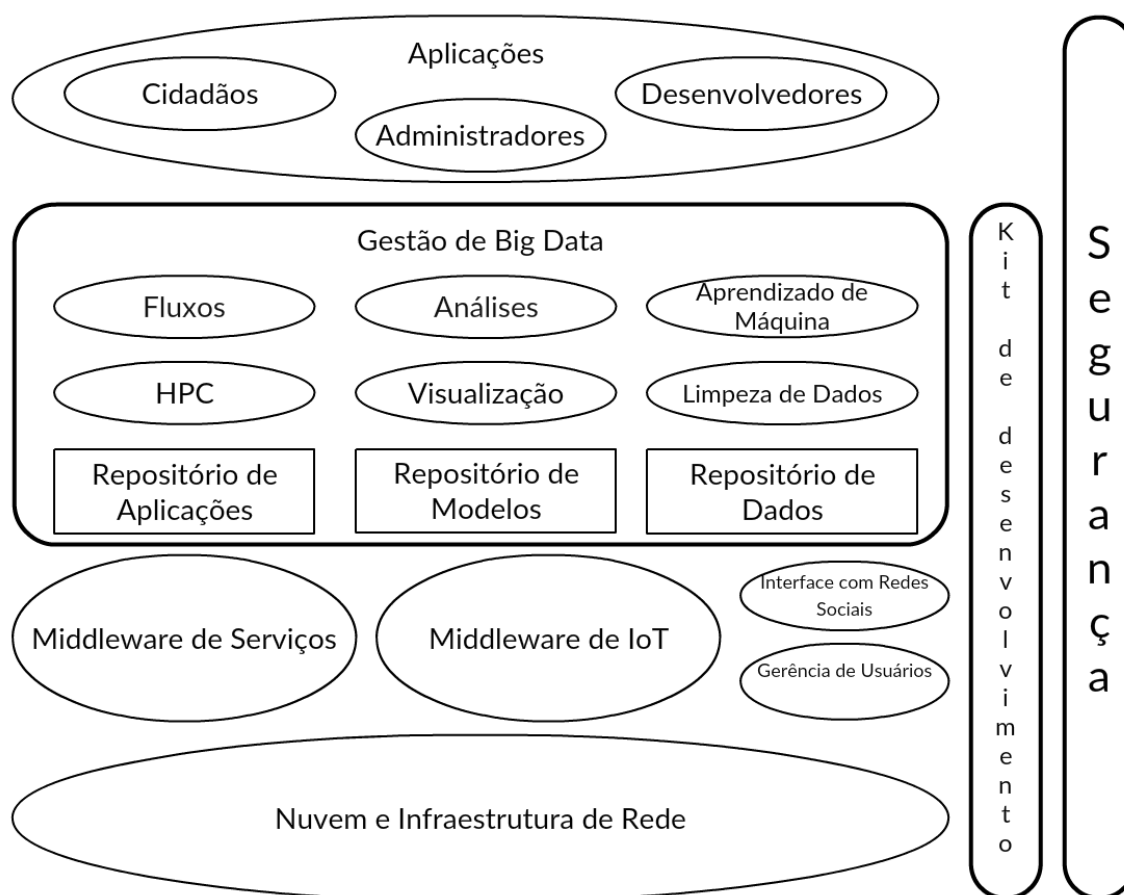


Figura 1.13. Arquitetura de Referência para Plataformas de Cidades Inteligentes

Para disponibilizar serviços e aplicações melhores para os cidadãos, é importante que a plataforma armazene alguns dados e preferências dos usuários, isso é papel do componente de **Gestão de Usuários**. Porém, para garantir a privacidade do usuário, esses dados devem ser devidamente protegidos e devem ser coletados apenas com a expressa autorização dos usuários. Além disso, como haverão diversas aplicações sendo executadas na plataforma, esses dados serão úteis para oferecer um mecanismo único de autenticação (*single sign-on*).

Redes Sociais serão bastante importantes para cidades inteligentes, podendo ser usadas para coletar dados sobre as condições da cidade a partir de atualizações de cidadãos e também como um canal eficiente de comunicação entre a plataforma, os administradores da cidade e a população. Portanto, é importante facilitar a integração da plataforma com as redes sociais existentes, o que é de responsabilidade do **Gateway de Rede Social**.

A **Gestão de Big Data** é uma camada reunindo vários módulos que cuidam da gestão dos dados na plataforma. Entre suas responsabilidades está o armazenamento dos dados da cidade inteligente e, para isso, são utilizados três repositórios: (1) um **Repositório de Aplicações** para armazenar as aplicações, incluindo seu código fonte, arquivos executáveis e documentação; (2) um **Repositório de Modelos** para armazenar os diversos modelos que descrevem a cidade tais como modelos de trânsito, da rede de sensores, da

rede de distribuição de energia e mapas e (3) um **Repositório de Dados** para armazenar dados coletados dos sensores, cidadãos e aplicações. Devido à diversidade de tipos e à quantidade de dados que serão coletados, existirão diversos repositórios de dados espalhados pela cidade que utilizarão tanto bancos de dados relacionais quanto NoSQL.

Além do armazenamento de dados, a camada de Gestão de Big Data também é responsável pelo processamento dos dados da cidade. Existem dois tipos de processamento que são mais adequados para diferentes situações: **Processamento de Fluxos de dados** para realizar o processamento de fluxos de dados contínuos e de processamento em tempo real e **Processamento em Lote**, para analisar grandes conjuntos de dados e realizar o processamento de dados históricos. Ainda, esse módulo deve ser capaz de realizar diversos tipos de pré-processamento como filtragem, normalização e transformações.

Existe ainda um componente incluindo algoritmos de aprendizado de máquina que tem como objetivo “descobrir” de forma automatizada o comportamento da cidade inferindo modelos que descrevem a dinâmica dos agentes presentes na cidade. Dessa forma, será possível o processamento dos dados históricos a fim de gerar modelos que serão utilizados para prever o comportamento futuro da cidade. Finalmente, como uma cidade inteligente vai produzir uma quantidade enorme de dados, um componente para fazer a **Limpeza de Dados** é necessário, excluindo dados que não são mais necessários e armazenando dados antigos em dispositivos mais lentos e com maior capacidade.

Inúmeras ferramentas de software livre de alta qualidade podem ser utilizadas para a implementação dessa camada. Para o processamento de fluxos de dados alguns exemplos são o Apache Storm e o Apache Spark. Para o processamento em lote de grandes conjuntos de dados o Apache Hadoop e o Apache Spark. Para os repositórios de dados, bancos de dados relacionais como o PostgreSQL e o MySQL para armazenar dados sobre a plataforma como usuários e configurações e bancos de dados NoSQL, tais como como o CouchDB, MongoDB ou Cassandra, para o armazenamento de dados de sensores e dispositivos.

Utilizando a plataforma e os componentes descritos acima, desenvolvedores de software poderão desenvolver mais facilmente aplicações para cidades inteligentes. Disponibilizando dados e serviços abertos da cidade, cidadãos comuns podem executar, ou até mesmo desenvolver novas aplicações sobre a infraestrutura da cidade inteligente utilizando os dados de sensores, dispositivos e de serviços da cidade. As aplicações, além de usar os dados da plataforma, também poderão gerar novos dados que serão salvos na plataforma e que poderão ser usados por outras aplicações.

A plataforma deve ainda disponibilizar ferramentas para facilitar o desenvolvimento das aplicações com um ambiente integrado de desenvolvimento (*Integrated Development Environment - IDE*), bibliotecas de componentes e *frameworks* e um simulador para permitir testes com diferentes cenários e situações.

Todos os componentes da plataforma devem levar em consideração vários dos requisitos não-funcionais apresentados na Seção 1.6.2 como escalabilidade, segurança, privacidade e interoperabilidade. Escalabilidade é fundamental por causa da quantidade de dados, dispositivos, usuários e serviços que uma plataforma de cidades inteligentes terá que gerenciar. Privacidade e Segurança são importantes porque uma plataforma de

idades inteligentes coleta, armazena e processa dados sensíveis dos cidadãos, empresas, ONGs e da administração da cidade. Interoperabilidade permitirá a operação integrada de diferentes tipos de serviços, dispositivos e aplicações.

1.8. Desafios Científicos e Tecnológicos

Nesta seção serão apresentados os principais desafios de pesquisa para a implantação de plataformas, aplicações e da infraestrutura de cidades inteligentes. Entre esses desafios estão a segurança e a confiabilidade das informações dos cidadãos, o custo e as dificuldades da criação e manutenção de infraestrutura, sistemas e plataformas e os desafios inerentes à implementação e implantação de grandes sistemas distribuídos como escalabilidade e heterogeneidade. Além disso, há ainda desafios sociais e culturais importante, como o incentivo à colaboração da população e o bom uso dos recursos públicos.

1.8.1. Privacidade

Diversos dados dos cidadãos serão manipulados pela infraestrutura, pelas aplicações e pelas plataformas de uma cidade inteligente. Esses dados devem ser protegidos para evitar que usuários mal-intencionados consigam acesso aos dados. Também é indispensável que a forma que os dados serão armazenados e usados sejam notificados aos usuários e que qualquer alteração seja avisada antecipadamente.

1.8.2. Segurança

Além de garantir a privacidade dos dados dos usuários, a infraestrutura da cidade também deve ser segura. Deve ser possível a prevenção e recuperação de ataques à segurança dos sistemas. Isso é importante, pois diversos serviços dependerão do correto funcionamento de todos os elementos da infraestrutura da cidade, como sensores, aplicações e plataformas. Além disso, com a evolução das cidades inteligentes, o cidadão deve ficar dependente desses serviços, assim como hoje a maioria da população de uma grande cidade é dependente do sistema de transporte público.

Alguns exemplos de ataques que uma cidade inteligente pode sofrer são: ataque à infraestrutura de hardware, alterando os valores de leitura de sensores, ataques de negação de serviço, que deixam serviços indisponíveis e vandalismo nos elementos físicos da cidade (Piro et al. 2014; Hernández-Muñoz et al. 2011; Petrolo et al. 2014).

Alguns autores discutem que a cidade deve assegurar que seus sistemas sejam a prova de ciberterrorismo e cibervandalismo destacando que uma cidade com uma rede de sensores e atuadores deve ser especialmente segura, pois um usuário malicioso pode controlar a infraestrutura da cidade causando sérios problemas como acidentes, erros na leituras de dados e ataques aos serviços públicos (Hancke et al. 2012; Gurgun et al. 2013).

1.8.3. Gestão dos Dados

Uma das principais características de cidades inteligentes é o grande volume de dados gerados. Esses dados podem ser estruturados, como as informações dos cidadãos, semi-estruturados como dados de leituras de sensores e não-estruturados como os fluxos de imagens de câmeras de tráfego e de segurança.

Alguns dos desafios de pesquisa nessa área são:

- **Armazenamento:** A quantidade de dados que deverão ser salvos em uma cidade inteligente é muito grande, por isso são necessárias ferramentas que sejam capazes lidar com esse volume. Também serão necessários mecanismos que facilitem o armazenamento e recuperação de dados de estruturados e não estruturados.
- **Processamento:** Assim como no item anterior, para comportar o grande volume de dados, serão necessárias ferramentas que consigam fazer o processamento em um tempo aceitável. Algumas ferramentas já estão começando a ser usadas para esse fim como o Spark, o Hadoop e o Storm, mas muita pesquisa ainda é necessária para torná-las eficazes para a enorme quantidade de possibilidades de uso que se abre com as aplicações em cidades inteligentes.
- **Modelos:** Além do desempenho, outro desafio é a dificuldade da construção de modelos de dados completos e eficientes para que seja possível o processamento dos dados, a implementação de aplicações e a utilização de algoritmos de agregação, clusterização e inferência. Pesquisas científicas e tecnológicas nas áreas de Inteligência Artificial, Mineração de Dados, Reconhecimento de Padrões e Aprendizado de Máquina serão necessárias para solucionar esses problemas.
- **Confiança:** Os diversos dispositivos que compõem uma cidade inteligente estão sujeitos a falhas. Por isso, é necessário o uso de ferramentas e algoritmos de validação dos dados. Por exemplo, em uma região da cidade em que o sinal do GPS não estiver bom, a leitura da posição de um ônibus ou de um sensor pode ser incorreta. Os desafios aqui são tanto diferenciar os dados bons dos dados ruins quanto desenvolver mecanismos para inferir os dados corretos a partir do conjunto total de dados disponíveis.

Em todos os desafios citados acima, é necessário garantir diversos requisitos não-funcionais como elasticidade (permitindo que a quantidade de recursos utilizados seja adequada à demanda), escalabilidade (para que o hardware e os serviços suportem picos de demanda) e privacidade (porque os dados analisados pertencerão aos cidadãos ou organizações da cidade).

1.8.4. Escalabilidade

Todos os componentes de uma cidade inteligente devem ser escaláveis para conseguir atender o crescimento da demanda por serviços e dados da cidade. Isso é necessário devido ao aumento populacional que a cidade pode ter e também para suportar eventos inesperados que podem fazer o acesso às aplicações e plataformas aumentarem ordens de magnitude em um pequeno intervalo de tempo, tais como grandes eventos, engarrafamentos e desastres naturais.

Além disso, é esperado que a quantidade de dados coletados aumente constantemente, devida à implantação de mais dispositivos, à criação de novos serviços e aplicações e ao aumento populacional. Por isso, um dos principais desafios na implantação da infraestrutura e na implementação de plataformas, serviços e aplicações é garantir que eles sejam escaláveis.

1.8.5. Heterogeneidade

A interoperabilidade entre a infraestrutura, aplicações e plataformas é um dos principais desafios para a construção de uma cidade inteligente. Para isso, é necessário lidar com a grande heterogeneidade dos componentes de hardware e software que compõem o ambiente de uma cidade inteligente.

Alguns exemplos de onde esse desafio ocorre são: na instalação de múltiplos sensores e atuadores de diferentes fabricantes que possuem diferentes protocolos, nos diferentes tipos de semáforos que já estão instalados nas cidades e utilizam diferentes protocolos de comunicação e nas aplicações legadas da cidade que foram implementadas com diferentes linguagens de programação e interfaces quase nunca compatíveis.

Alguns autores citam diferentes módulos de uma cidade inteligente que devem lidar com esse desafio. Naphade et al. (Naphade et al. 2011) discute o problema dos dados que são coletados de diversas fontes e que necessitam de um modelo comum para poderem ser agregados e processados. Outros autores (Wenge et al. 2014) defendem a definição de padrões em vista dos dispositivos, sistemas e domínios heterogêneos.

1.8.6. Manutenção e Implantação da Infraestrutura

A criação de um ambiente de cidade inteligente demandará diversos investimentos para a implantação da infraestrutura necessária, como a implantação de uma rede de sensores e atuadores, a melhoria das redes sem fio, a integração entre os diferentes sistemas da cidade e a coleta de dados da infraestrutura já existente como pontos de ônibus e semáforos.

Após a implantação, será necessário também fazer a manutenção de todos esses componentes, pois todos estão sujeitos a falhas parciais e a quebras por completo. Por exemplo, um sensor pode ser danificado por vandalismo ou por acidente, ou pode fornecer dados errados por uma falha natural do equipamento. Isso será especialmente desafiador devido à grande quantidade de dispositivos instalados.

Este é um desafio técnico importante pois, mesmo a manutenção da infraestrutura já existente como ruas, semáforos, pontos de ônibus, sinalização e praças, já não é a ideal em muitas partes do mundo.

A dificuldade da manutenção de todo o software e hardware necessário para uma cidade inteligente, principalmente por causa da quantidade de dispositivos espalhados pela cidade é um desafio enorme (Perera et al. 2014; Wenge et al. 2014; Hancke et al. 2012). Parece ser fundamental a adoção de estratégias baseadas em monitoramento automático acopladas à solução automatizada de problemas com a infraestrutura.

1.8.7. Custos

Um problema para a implantação de uma infraestrutura de cidades inteligentes são os custos para a construção de todos os componentes necessários. Esse custo inclui a aquisição e instalação de todos os dispositivos necessários na cidade como sensores, atuadores, servidores e equipamentos de comunicação, o desenvolvimento do software, a contratação de equipes para manutenção e gerenciamento e a adaptação às mudanças nos processos da cidade.

Al Nuaimi et al. (Al Nuaimi et al. 2015) discutem a possibilidade de um projeto de cidade inteligente não ser corretamente desenvolvido, acarretando em custos elevados que serão desperdiçados. Alguns exemplos citados são o uso de um novo sistema de semáforos, que se for mal implementado pode piorar o trânsito e até causar acidentes. Por isso, é citada a necessidade de monitoramento constante de todas as iniciativas de cidades inteligentes e o desenvolvimento de projetos pilotos em escala reduzida antes da implantação em toda a cidade.

Outro problema, também relacionado a custos, é que os benefícios da implantação de muitos dos serviços de cidades inteligentes apenas são percebidos pela população no longo prazo. Isso desincentiva os políticos, que muitas vezes estão interessados apenas na próxima eleição, a investirem em projetos desse tipo que possuem um custo elevado e que não necessariamente trazem benefícios a curto prazo.

1.8.8. Colaboração

Um desafio importante é incentivar a população a utilizar os serviços que serão disponibilizados na cidade e a compartilhar dados e informações sobre esse uso. Muitos dos serviços de cidades inteligentes dependem do engajamento da população. Wu et al. (Wu et al. 2014) cita a necessidade da criação de sistemas que incentivem a colaboração criando um ambiente onde os usuário de serviços e aplicações tenham uma relação de benefício mútuo quando eles contribuem, compartilham e usam dados da cidade.

Um projeto interessante para aumentar o engajamento da população em aplicações de Cidades Inteligentes é o sistema MITOS (*Multi-Input TranspOrt planning System*) (Diamantaki et al. 2013) que incorpora na plataforma SmartSantander elementos e mecanismos de jogos. O sistema permite a distribuição de premiações para usuários que realizam uma ou um conjunto de tarefas, como por exemplo o usuário que utilizou mais o transporte público em um mês, ou que pegou um determinado número de ônibus em um dia.

1.9. Implicações

Para melhorar a qualidade de vida das pessoas e otimizar o uso dos recursos da cidade, diversas iniciativas e intervenções serão necessárias na cidade para que elas fiquem mais “inteligentes”. Isso traz diversas implicações para diversos agentes envolvidos nos processos da cidade como cidadãos, prefeitos, vereadores, desenvolvedores, administradores de sistemas, empreendedores e cientistas.

O principal objetivo de cidades inteligentes é a melhora na qualidade de vida dos cidadãos, isso será alcançado através do desenvolvimento de serviços e aplicações inteligentes utilizando os dados coletados na cidade e atuando sobre o seu espaço otimizando o funcionamento dos equipamentos e da infraestrutura urbana. Isso alterará profundamente o dia-a-dia das pessoas, facilitando tomadas de decisões baseadas em informações mais confiáveis e economizando tempo com novos e melhores serviços. Além disso, para melhorar os resultados das novas aplicações e serviços, as pessoas devem participar ativamente desses sistemas, compartilhando seus dados com as aplicações e com outros cidadãos.

As tecnologias citadas neste capítulo indicam a infraestrutura necessária para a implantação de ambientes de Cidades Inteligentes. Isso pode ajudar administradores da cidade a tomar decisões quanto aos investimentos necessários como a compra, instalação e manutenção da rede de sensores, a quantidade de servidores e dispositivos de comunicação necessários e os custos para o desenvolvimento dos sistemas necessários para a cidade. Além disso, é possível analisar as iniciativas de cidades inteligentes existentes e definir quais terão maior impacto na cidade.

Para desenvolvedores de aplicações e *startups* que implementam aplicações relacionadas a cidades, essas iniciativas abrem uma grande oportunidade de negócios, principalmente com a utilização dos dados que as prefeituras disponibilizam em portais de dados abertos ou via APIs com dados em tempo real.

Finalmente, para pesquisadores da área de cidades inteligentes, é necessário entender os desafios técnicos e científicos ainda em aberto para a criação de uma cidade verdadeiramente inteligente. Também será necessário pesquisar quais os impactos (positivos e negativos) que essas tecnologias trarão para o cidadão, administradores, meio ambiente e os serviços da cidade.

1.10. Conclusões

Com o crescimento da população das grandes cidades ao redor mundo e o grande número de problemas que elas enfrentam, a necessidade de tornar as cidades mais inteligentes é cada vez maior. Os principais benefícios disso são a otimização da infraestrutura e serviços da cidade, o uso mais sustentável dos recursos e, conseqüentemente, a melhoria da qualidade de vida da população.

Este capítulo mostrou diversas iniciativas de cidades inteligentes ao redor do mundo como nas cidades de Santader, Amsterdã e Barcelona. Esses exemplos mostram que existem diversos projetos interessantes sendo desenvolvidos, mas que, em 2016, nenhuma cidade ainda possui uma infraestrutura completa de hardware e software para a coleta e análise dos dados urbanos e o desenvolvimento de aplicações e serviços para os cidadãos.

Além dessas iniciativas práticas, discutimos diversos projetos de pesquisa que estão desenvolvendo plataformas de software servindo de suporte para a implementação de aplicações e processamento dos dados coletados na cidade. Entre as funcionalidades oferecidas por essas plataformas estão o gerenciamento de dados e serviços e o fornecimento de ferramentas que facilitam o desenvolvimento de aplicações. Além disso, as plataformas buscam prover um conjunto essencial de requisitos não-funcionais como escalabilidade, adaptação e sensibilidade ao contexto.

Ainda existem uma grande quantidade de desafios técnicos e de pesquisa no desenvolvimento de cidades inteligentes que precisa ser melhor explorados. Entre eles destacamos a necessidade de garantir a privacidade dos dados dos usuários nos sistemas da cidade, a segurança para tornar a infraestrutura e os sistemas da cidade a prova de usuários mal-intencionados, a escalabilidade na comunicação, armazenamento e processamento de dados para que mais cidadãos possam usar os serviços oferecidos, os altos custos ainda proibitivos em muitos casos e a dificuldade de manutenção de toda a infraestrutura de hardware e software da cidade que podem tornar as iniciativas de Cidades Inteligentes

inviáveis.

Nas próximas duas décadas, presenciaremos o surgimento de centenas de cidades ao redor do mundo onde a Tecnologia da Informação e da Comunicação estará totalmente permeada no ambiente urbano ajudando a tornar as cidades mais sustentáveis, agradáveis e eficientes e menos estressantes e violentas.

Esperamos que a comunidade científica, trabalhando conjuntamente com nossos empreendedores, governantes e a população, possam desenvolver soluções criativas e eficazes para os desafios elencados acima de forma a atingir o objetivo essencial das cidades inteligentes: contribuir para a melhoria da qualidade de vida de todos os habitantes das cidades.

Referências

- Aazam, M., Khan, I., Alsaffar, A. A., and Huh, E.-N. (2014). Cloud of things: Integrating internet of things and cloud computing and the issues involved. In *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on*, pages 414–419. IEEE.
- Al Nuaimi, E., Al Neyadi, H., Mohamed, N., and Al-Jaroodi, J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications*, 6(1):25.
- AlAwadhi, S. and Scholl, H. J. (2013). Aspirations and realizations: The smart city of seattle. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 1695–1703. IEEE.
- Anagnostopoulos, T., Kolomvatsos, K., Anagnostopoulos, C., Zaslavsky, A., and Hadji-efthymiades, S. (2015). Assessing dynamic models for high priority waste collection in smart cities. *Journal of Systems and Software*, 110:178 – 192.
- Apolinarski, W., Iqbal, U., and Parreira, J. X. (2014). The gambas middleware and sdk for smart city applications. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 117–122.
- Bain, M. (2014). Sentilo - sensor and actuator platform for smart cities.
- Ballon, P., Glidden, J., Kranas, P., Menychtas, A., Ruston, S., and Van Der Graaf, S. (2011). Is there a need for a cloud platform for european smart cities? In *eChallenges e-2011 Conference Proceedings, IIMC International Information Management Corporation*.
- Barba, C. T., Mateos, M. A., Soto, P. R., Mezher, A. M., and Igartua, M. A. (2012). Smart city for vanets using warning messages, traffic statistics and intelligent traffic lights. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 902–907. IEEE.
- Bressan, N., Bazzaco, L., Bui, N., Casari, P., Vangelista, L., and Zorzi, M. (2010). The deployment of a smart monitoring system using wireless sensor and actuator networks. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 49–54.

- Bui, N. and Zorzi, M. (2011). Health care applications: a solution based on the internet of things. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, page 131. ACM.
- Calbimonte, J.-P., Sarni, S., Eberle, J., and Aberer, K. (2014). Xgsn: An open-source semantic sensing middleware for the web of things. In *7th International Workshop on Semantic Sensor Networks*, number EPFL-CONF-200926.
- Caragliu, A., Del Bo, C., and Nijkamp, P. (2011). Smart cities in europe. *Journal of urban technology*, 18(2):65–82.
- Cardoso, R. S. and Issarny, V. (2007). Architecting pervasive computing systems for privacy: A survey. In *The Working IEEE/IFIP Conference on Software Architecture, 2007. WICSA'07*. IEEE.
- Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209.
- Cheng, B., Longo, S., Cirillo, F., Bauer, M., and Kovacs, E. (2015). Building a big data platform for smart cities: Experience and lessons from santander. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 592–599.
- Cocchia, A. (2014). Smart and digital city: A systematic literature review. In Dameri, R. P. and Rosenthal-Sabroux, C., editors, *Smart City*, Progress in IS, pages 13–43. Springer International Publishing.
- Coetzee, L. and Eksteen, J. (2011). The internet of things – promise for the future? an introduction. In *IST-Africa Conference Proceedings, 2011*, pages 1–9.
- Dameri, R. P. (2013). Searching for smart city definition: a comprehensive proposal. *International Journal of Computers & Technology*, 11(5):2544–2551.
- Demchenko, Y., de Laat, C., and Membrey, P. (2014). Defining architecture components of the big data ecosystem. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 104–112. IEEE.
- Diamantaki, K., Rizopoulos, C., Tsetsos, V., Theona, I., Charitos, D., and Kaimakamis, N. (2013). Integrating game elements for increasing engagement and enhancing user experience in a smart city context. In *Intelligent Environments (Workshops)*, pages 160–171.
- Distefano, S., Merlino, G., and Puliafito, A. (2012). Enabling the cloud of things. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 858–863. IEEE.
- Djahel, S., Doolan, R., Muntean, G., and Murphy, J. (2014). A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *Communications Surveys Tutorials, IEEE*, PP(99):1–1.

- Elmangoush, A., Coskun, H., Wahle, S., and Magedanz, T. (2013). Design aspects for a reference m2m communication platform for smart cities. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*, pages 204–209.
- Foell, S., Kortuem, G., Rawassizadeh, R., Handte, M., Iqbal, U., and Marrón, P. (2014). Micro-navigation for urban bus passengers: Using the internet of things to improve the public transport experience. In *Proceedings of the First International Conference on IoT in Urban Space, URB-IOT '14*, pages 1–6, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- Fortes, M., Ferreira, V., Sotelo, G., Cabral, A., Correia, W., and Pacheco, O. (2014). Deployment of smart metering in the búzios city. In *Transmission & Distribution Conference and Exposition-Latin America (PES T&D-LA), 2014 IEEE PES*, pages 1–6. IEEE.
- Fox, A., Patterson, D. A., and Joseph, S. (2013). *Engineering software as a service: An Agile approach using Cloud Computing*. Strawberry Canyon LLC.
- Fox, A., Patterson, D. A., and Joseph, S. (2015). *Construindo Software como Serviço: Uma Abordagem Ágil Usando Computação em Nuvem*. Strawberry Canyon LLC.
- Franke, T., Lukowicz, P., and Blanke, U. (2015). Smart crowds in smart cities: real life, city scale deployments of a smartphone based participatory crowd management platform. *Journal of Internet Services and Applications*, 6(1):1–19.
- Galache, J. A., Yonezawa, T., Gurgen, L., Pavia, D., Grella, M., and Maeomichi, H. (2014). Clout: Leveraging cloud computing techniques for improving management of massive iot data. In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*, pages 324–327.
- Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovic, N., and Meijers, E. (2007). Smart cities-ranking of european medium-sized cities. Technical report, Vienna University of Technology.
- Girtelschmid, S., Steinbauer, M., Kumar, V., Fensel, A., and Kotsis, G. (2013). Big data in large scale intelligent smart city installations. In *Proceedings of International Conference on Information Integration and Web-based Applications and Services, IIWAS '13*, pages 428:428–428:432, New York, NY, USA. ACM.
- Goldman, A., Kon, F., Junior, F. P., Polato, I., and de Fátima Pereira, R. (2012). Apache hadoop: conceitos teóricos e práticos, evolução e novas possibilidades. *XXXI Jornadas de atualizações em informatica*.
- Guan, L. (2012). Smart steps to a battery city. *Government News*, 32(2):24–27.
- Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

- Gurgen, L., Gunalp, O., Benazzouz, Y., and Gallissot, M. (2013). Self-aware cyber-physical systems and applications in smart buildings and cities. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1149–1154.
- Hall, P. et al. (2000). Creative cities and economic development. *Urban studies*, 37(4):639–649.
- Hancke, G. P., Hancke Jr., G. P., et al. (2012). The role of advanced sensing in smart cities. *Sensors*, 13(1):393–425.
- Handte, M., Iqbal, M. U., Wagner, S., Apolinarski, W., Marrón, P. J., Navarro, E. M. M., Martinez, S., Barthelemy, S. I., and Fernández, M. G. (2014). Crowd density estimation for public transport vehicles. In *EDBT/ICDT Workshops*, pages 315–322.
- Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., and Williams, P. (2010). Foundations for smarter cities. *IBM Journal of Research and Development*, 54(4):1–16.
- Hernández-Muñoz, J. M., Vercher, J. B., Muñoz, L., Galache, J. A., Presser, M., Hernández Gómez, L. A., and Pettersson, J. (2011). Smart cities at the forefront of the future internet. volume 6656, pages 447–462.
- Hussain, A., Wenbi, R., da Silva, A. L., Nadher, M., and Mudhish, M. (2015). Health and emergency-care platform for the elderly and disabled people in the smart city. *Journal of Systems and Software*, 110:253 – 263.
- Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadist, P., Autili, M., Gerosa, M. A., and Hamida, A. (2011). Service-oriented middleware for the future internet: state of the art and research directions. *Journal of Internet Services and Applications*, 2(1):23–45.
- Janajreh, I., Su, L., and Alan, F. (2013). Wind energy assessment: Masdar city case study. *Renewable energy*, 52:8–15.
- Khan, Z., Anjum, A., and Kiani, S. L. (2013). Cloud based big data analytics for smart future cities. In *Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on*, pages 381–386.
- Khan, Z., Anjum, A., Soomro, K., and Tahir, M. A. (2015). Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing*, 4(1):1–11.
- Komninos, N. and Tsarchopoulos, P. (2013). Toward intelligent thessaloniki: From an agglomeration of apps to smart districts. *Journal of the Knowledge Economy*, 4(2):149–168.
- Kshetri, N., Alcantara, L. L., and Park, Y. (2014). Development of a smart city and its adoption and acceptance: the case of new songdo. *Communications & Strategies*, (96):113.

- Le-Phuoc, D., Nguyen-Mau, H. Q., Parreira, J. X., and Hauswirth, M. (2012). A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, 16:42–51.
- Lee, Y. W. and Rho, S. (2010). U-city portal for smart ubiquitous middleware. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 1, pages 609–613.
- Leite, L., Moreira, C. E., Cordeiro, D., Gerosa, M. A., and Kon, F. (2014). Deploying large-scale service compositions on the cloud with the choreos enactment engine. In *13th IEEE International Symposium on Network Computing and Applications (NCA)*, pages 121–128. IEEE.
- Liu, P. and Peng, Z. (2013). Smart cities in china. *IEEE Computer Society*, 16.
- Manville, C., Cochrane, G., Cave, J., Millard, J., Pederson, J. K., Thaarup, R. K., Liebe, A., Wissner, M., Massink, R., and Kotterink, B. (2014). Mapping smart cities in the eu.
- Mitton, N., Papavassiliou, S., Puliafito, A., and Trivedi, K. (2012). Combining cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*, 2012(1).
- Mylonas, G., Theodoridis, E., and Munoz, L. (2015). Integrating smartphones into the smartsantander infrastructure. *Internet Computing, IEEE*, 19(2):48–56.
- Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., and Morris, R. (2011). Smarter cities and their innovation challenges. *Computer*, 44(6):32–39.
- Papa, R., Gargiulo, C., and Galderisi, A. (2013). Towards an urban planners’ perspective on smart city. *TeMA Journal of Land Use, Mobility and Environment*, 6(01):5–17.
- Parkavi, A. and Vetrivelan, N. (2013). A smart citizen information system using hadoop: A case study. In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pages 1–3.
- Pereira, R. L., Sousa, P. C., Barata, R., Oliveira, A., and Monsieur, G. (2015). Citysdk tourism api-building value around open data. *Journal of Internet Services and Applications*, 6(1):1–13.
- Perera, C., Zaslavsky, A. B., Christen, P., and Georgakopoulos, D. (2014). Sensing as a service model for smart cities supported by internet of things. *Trans. Emerging Telecommunications Technologies*, 25(1):81–93.
- Pérez-González, D. and Díaz-Díaz, R. (2015). Public services provided with ict in the smart city environment: The case of spanish cities. *Journal of Universal Computer Science*, 21(2):248–267.
- Petrolo, R., Loscri, V., and Mitton, N. (2014). Towards a Cloud of Things Smart City. *IEEE COMSOC MMTTC E-Letter*, 9(5):44–48.

- Piro, G., Cianci, I., Grieco, L. A., Boggia, G., and Camarda, P. (2014). Information centric services in smart cities. *Journal of Systems and Software*, 88(0):169 – 188.
- Polato, I., Ré, R., Goldman, A., and Kon, F. (2014). A comprehensive view of hadoop research—a systematic literature review. *Journal of Network and Computer Applications*, 46:1–25.
- Privat, G., Zhao, M., and Lemke, L. (2014). Towards a shared software infrastructure for smart homes, smart buildings and smart cities. In *International Workshop on Emerging Trends in the Engineering of Cyber-Physical Systems, Berlin*.
- Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., et al. (2014). Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238.
- Stephenson, M., Di Lorenzo, G., and Aonghusa, P. M. (2012). Open innovation portal: A collaborative platform for open city data sharing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 522–524. IEEE.
- Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things.
- Takahashi, K., Yamamoto, S., Okushi, A., Matsumoto, S., and Nakamura, M. (2012). Design and implementation of service api for large-scale house log in smart city cloud. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 815–820.
- Tei, K. and Gurgun, L. (2014). Clout: Cloud of things for empowering the citizen clout in smart cities. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 369–370. IEEE.
- Thornton, S. (2013). Chicago’s windygrid: Taking situational awareness to a new level.
- United Nations (2009). Urban and rural areas 2009.
- Vakali, A., Anthopoulos, L., and Krco, S. (2014). Smart cities data streams integration: Experimenting with internet of things and social data flows. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), WIMS ’14*, pages 60:1–60:5, New York, NY, USA. ACM.
- Villanueva, F. J., Santofimia, M. J., Villa, D., Barba, J., and Lopez, J. C. (2013). Civitas: The smart city middleware, from sensors to big data. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pages 445–450.
- Vlahogianni, E. I., Kepaptsoglou, K., Tsetsos, V., and Karlaftis, M. G. (2014). Exploiting new sensor technologies for real-time parking prediction in urban areas. In *Transportation Research Board 93rd Annual Meeting Compendium of Papers*, pages 14–1673.

- Wan, J., Li, D., Zou, C., and Zhou, K. (2012). M2m communications for smart city: An event-based architecture. In *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on*, pages 895–900.
- Washburn, D., Sindhu, U., Balaouras, S., Dines, R., Hayes, N., and Nelson, L. (2009). Helping cios understand “smart city” initiatives. *Growth*, 17:2.
- Wenge, R., Zhang, X., Cooper, D., Chao, L., and Hao, S. (2014). Smart city architecture: A technology guide for implementation and design challenges. *Communications, China*, 11(3):56–69.
- Wu, C., Birch, D., Silva, D., Lee, C.-H., Tsinalis, O., and Guo, Y. (2014). Concinnity: A generic platform for big sensor data applications. *Cloud Computing, IEEE*, 1(2):42–50.
- Yamamoto, S., Matsumoto, S., Saiki, S., and Nakamura, M. (2014). Using materialized view as a service of scallop4sc for smart city application services. In *Soft Computing in Big Data Processing*, pages 51–60. Springer.
- Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., and David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, pages 1–18.
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, volume 10, page 10.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *Internet of Things Journal, IEEE*, 1(1):22–32.

Capítulo

2

Computação Urbana: Técnicas para o Estudo de Sociedades com Redes de Sensoriamento Participativo

Thiago H. Silva e Antonio A. F. Loureiro

Abstract

Urban computing is a recent research topic that aims to obtain and analyze urban data from various sources, such as traditional wireless sensor networks (WSNs) and emerging participatory sensor networks (PSNs) to understand and address issues that cities face. PSNs are particularly interesting because they rely on the participation of users in urban sensing, allowing the observation of large-scale actions of people in (almost) real time over long periods of time. PSN data increase our knowledge about different aspects of our lives in urban scenarios, which can be very useful in developing more sophisticated applications to various sectors, especially those related to the understanding of urban societies. The purpose of this short course is to discuss the concept of urban computing and urban sensing with participatory sensor networks. We aim to show the relevance of urban computing and motivate the construction of new applications that address issues related to the dynamics of cities and urban social behavior. In addition, this short course will discuss how to work with PSNs, by analyzing their properties and their usefulness in the development of new applications in urban computing.

Resumo

A computação urbana (urban computing) é um tema recente de pesquisa que visa obter e analisar dados urbanos de diversas fontes, como as tradicionais redes de sensores sem fio (RSSFs) e as emergentes redes de sensoriamento participativo (RSP), com o objetivo de entender e tratar questões enfrentadas pelas cidades. As RSPs são particularmente interessantes nesse caso, pois contam com a colaboração dos usuários no sensoriamento urbano, permitindo a observação das ações das pessoas em larga escala em tempo (quase) real durante longos períodos de tempo. Dados de RSPs aumentam o nosso conhecimento sobre diferentes aspectos de nossas vidas em cenários urbanos o que pode ser bastante útil no desenvolvimento de aplicações mais sofisticadas para diversos segmentos, principalmente os relacionados com o entendimento de sociedades urbanas. O objetivo deste minicurso é discutir o conceito de computação urbana e de sensoriamento urbano com redes de sensoriamento participativo. Visamos mostrar a relevância da computação urbana e motivar a

construção de novas aplicações que sirvam para tratar questões relacionadas com a dinâmica de cidades e do comportamento social urbano. Além disso, este minicurso discutirá como trabalhar com RSPs, ao analisar as suas propriedades e a sua utilidade no desenvolvimento de novas aplicações na área de computação urbana.

2.1. Introdução

A computação urbana (*urban computing*) [Kindberg et al. 2007, Kostakos and O’Neill 2008, Zheng et al. 2014a] é uma área interdisciplinar que diz respeito ao estudo e tratamento de questões enfrentadas pelas cidades utilizando tecnologia de computação. Por essa razão, a computação urbana conta com profissionais e aplicações em campos que incluem: antropologia, planejamento urbano, engenharia civil, ciência da computação, entre outros.

Como mais de 50% da população do mundo hoje vive em cidades [Martine et al. 2007], uma das consequências é uma enorme pressão sobre as suas infraestruturas, como transporte, habitação, água e energia, gerando difíceis desafios. Para entender e tratar essas e outras questões com o intuito de melhorar a qualidade de vida das pessoas que vivem em cidades, na computação urbana são usadas diversas fontes de dados sobre o ambiente urbano, alguns exemplos incluem: as tradicionais redes de sensores sem fio (RSSFs) [Loureiro et al. 2003]; e as emergentes redes de sensoriamento participativo (RSP) [Burke et al. 2006, Silva et al. 2014a].

As RSPs são particularmente interessantes nesse caso, pois contam com a colaboração dos usuários no sensoriamento urbano e permitem a observação em larga escala das ações das pessoas em tempo (quase) real durante longos períodos de tempo, possibilitando o entendimento da dinâmica da cidade e do comportamento social urbano. Com isso, as RSPs têm o potencial de se tornarem ferramentas fundamentais para a computação urbana. Dados de RSPs aumentam o nosso conhecimento sobre diferentes aspectos de nossas vidas em cenários urbanos o que pode ser bastante útil no desenvolvimento de aplicações mais sofisticadas em diversos segmentos, principalmente os relacionados com o entendimento de sociedades urbanas.

O objetivo deste minicurso é apresentar o conceito de computação urbana e de sensoriamento urbano com o auxílio de redes de sensoriamento participativo. Isso inclui uma visão geral de trabalhos específicos que ilustram as tendências de pesquisa e os principais desafios e oportunidades da área.

O restante do capítulo está organizado da seguinte forma. A Seção 2.2 apresenta o conceito de computação urbana, incluindo um arcabouço para o desenvolvimento de aplicações nessa área e algumas das principais fontes de dados. A Seção 2.3 discute em mais detalhes uma das fontes de dados urbanos: as redes de sensoriamento participativo. A Seção 2.4 discute o gerenciamento de dados urbanos, o que inclui a obtenção e tratamento desses dados. A Seção 2.5 analisa dados urbanos de RSPs, apresentando algumas de suas principais propriedades. A Seção 2.6 apresenta as abordagens e modelos utilizados em diversas aplicações e serviços relacionados ao estudo de sociedades urbanas utilizando dados de RSPs. A Seção 2.7 discute as principais técnicas utilizadas nos trabalhos mencionados nas seções anteriores, bem como algumas das tecnologias e ferramentas comumente utilizadas para a análise de dados. A Seção 2.8 apresenta alguns dos principais desafios relacionados

com a utilização de RSPs na computação urbana, já a Seção 2.9 apresenta várias oportunidades nessa mesma direção. Finalmente, a Seção 2.10 apresenta as nossas conclusões.

2.2. Computação Urbana

2.2.1. Definição

O termo “computação urbana” foi introduzido pela primeira vez por Eric Paulos na edição de 2004 da conferência UbiComp [Eric Paulos and Townsend 2004] e em seu artigo *The Familiar Stranger* [Paulos and Goodman 2004], publicado nesse mesmo ano.

Pode-se definir a computação urbana como um processo de aquisição, integração e análise de um grande volume de dados heterogêneos gerados por diversas fontes em espaços urbanos, tais como sensores, veículos e seres humanos, para ajudar na solução de diversos problemas que as cidades enfrentam tais como congestionamento de trânsito, poluição do ar, falta de água e aumento do consumo de energia. Assim um dos principais objetivos dessa área é ajudar a melhorar a qualidade de vida das pessoas que vivem em ambientes urbanos [Zheng et al. 2014a].

A computação urbana também nos auxilia a compreender a natureza dos fenômenos urbanos, bem como prever o futuro das cidades. Essa é uma área bastante interdisciplinar resultante da fusão da área de ciência da computação com áreas tradicionais, como transporte, economia e sociologia no contexto dos espaços urbanos. No domínio da ciência da computação, a computação urbana tem interseção com, por exemplo, sistemas distribuídos, interação humano-computador, redes de computadores, redes de sensores, sistemas cooperativos e inteligência artificial.

2.2.2. Arcabouço da Computação Urbana

Nesta seção apresentamos um arcabouço para a computação urbana. A Figura 2.1 mostra uma visão geral desse arcabouço, destacando os três componentes mais importantes: (i) gerenciamento dos dados urbanos; (ii) análise dos dados urbanos; e (iii) desenvolvimento de serviços e aplicações.

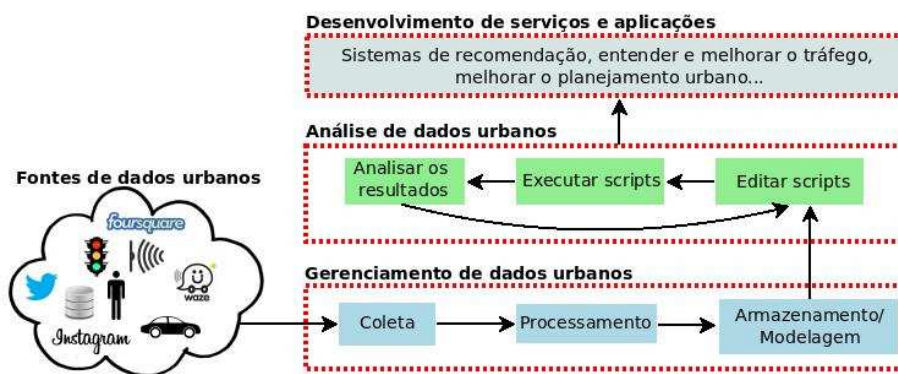


Figura 2.1. Visão geral do arcabouço da computação urbana.

Como ilustrado na figura, o componente gerenciamento de dados urbanos é composto de alguns passos importantes. O primeiro deles é o processo de coleta de dados urbanos, que podem ser obtidos de diversas fontes de dados, como discutido na próxima se-

ção. O segundo passo refere-se ao processamento desses dados. Após esse processamento podemos modelar os dados em diversos formatos, por exemplo, no formato de grafos, como discutido na Seção 2.4.

O componente análise dos dados urbanos é composto pela etapa de edição e execução de códigos, bem como a interpretação de resultados. Essa parte é fundamental, pois para utilizar dados urbanos é necessário conhecer suas propriedades. Mais detalhes sobre esse componente é descrito na Seção 2.5. Após a etapa de análise, o próximo passo é o desenvolvimento de serviços e aplicações com o conhecimento obtido. Essas aplicações podem ser de diversos tipos, como discutido na Seção 2.6.

2.2.3. Fontes de Dados Urbanos

Nesta seção apresentamos algumas das principais fontes de obtenção de dados urbanos. Esses dados oferecem suporte no desenvolvimento de novos serviços e aplicações na área de computação urbana.

- **Dados estatísticos oficiais:** fornecem dados referentes a um estudo estatístico sobre uma população, tais como dados demográficos, econômicos e sociais relativos a um momento determinado ou em certos períodos.

É possível encontrar diversas fontes de dados na Web disponibilizando dados dessa categoria para algumas localidades, como mostrado em [Barbosa et al. 2014]. No entanto, nem sempre esses dados estão disponíveis para a localidade que se deseja estudar. Outra dificuldade é a diversidade dos formatos nos quais os dados estão disponíveis, como em tabelas, mapas, gráficos, calendários, formulários, entre outros [Barbosa et al. 2014].

- **Redes de sensores tradicionais:** fornecem dados que são obtidos através da instalação de sensores específicos para algumas aplicações, por exemplo, sensores de presença em ruas e avenidas para detectar o volume de tráfego nesses locais, sensores para monitoramento da qualidade do ar em diversos pontos da cidade ou sensores para o monitoramento de níveis de ruídos.

Um problema com essa fonte de dados é a dificuldade de acesso aos dados. Além do custo de construção de uma rede de sensoriamento, geralmente, a implantação de sensores na cidade só é permitida pela prefeitura.

- **Infraestrutura das cidades:** fornecem dados que são capturados aproveitando as infraestruturas existentes da cidade, que são criadas para outros propósitos. Por exemplo, as redes de telefonia celular são construídas para comunicação móvel entre os indivíduos. No entanto, os sinais dos telefones celulares de um grande número de pessoas podem ser usados para tentar prever a mobilidade dos usuários e melhorar o planejamento urbano.

Outros tipos incluem a localização de veículos que possuem GPS. É cada vez mais comum ônibus, táxis, e veículos privados possuírem GPS embutidos. Esse tipo de dado contribui, por exemplo, para o entendimento do tráfego de uma cidade. Além disso, é possível obter dados de utilização do sistema de transporte público, já que é

bem comum esse tipo de sistema utilizar cartões RFID para registrar o uso de ônibus e metrô dos usuários.

A dificuldade de acesso a dados é também um problema dessa fonte, uma vez que somente a prefeitura ou empresas responsáveis, tipicamente, possuem acesso a esse tipo de dado.

- **Redes de sensoriamento participativos:** fornecem dados urbanos, que possuem uma escala bastante abrangente e podem ser mais fáceis de obter do que as outras fontes mencionadas, pois contam com a colaboração dos usuários na coleta de dados. Além disso, as RSPs podem possuir uma rede social online o que permite o estudo da estrutura social dos usuários, como relacionamentos e interações entre os usuários.

2.3. Sensoriamento Urbano com Redes de Sensoriamento Participativo

Como apresentamos na Seção 2.2, existem várias formas de obter dados urbanos, dentre elas podemos citar as emergentes redes de sensoriamento participativo (RSPs) [Silva et al. 2014a, Burke et al. 2006]. O tema será abordado da seguinte maneira: a Seção 2.3.1 apresenta a definição de uma RSP; a Seção 2.3.2 discute o funcionamento de uma RSP, enquanto a Seção 2.3.3 ilustra exemplos de RSPs.

2.3.1. O que é uma rede de sensoriamento participativo?

O sensoriamento participativo pode ser definido como um processo distribuído de coleta de dados pessoais e sobre diversos aspectos da cidade. Tal processo requer a participação ativa das pessoas para compartilhar voluntariamente informação contextual e/ou tornar seus dados sensorizados disponíveis [Burke et al. 2006], ou seja, o usuário determina manualmente como, quando, o quê e onde amostrar. Assim, através das RSPs é possível monitorar diversos aspectos das cidades, bem como o comportamento coletivo de pessoas conectadas à Internet em tempo (quase) real.

As RSPs têm se tornado populares graças ao aumento do uso de dispositivos portáteis, como *smartphones* e *tablets*, assim como a adoção mundial de sites de mídia social. Com isso, um elemento central de uma rede de sensoriamento participativo é a existência de um usuário capaz de realizar um sensoriamento, por exemplo, da cidade, com um dispositivo computacional portátil. Nesse cenário, as pessoas participam como sensores sociais, fornecendo dados voluntariamente sobre um determinado aspecto de um local que implicitamente capturam as suas experiências de vida diária. Esses dados podem ser obtidos com a ajuda de dispositivos de sensoriamento, como sensores incorporados a *smartphones* (e.g., GPS, acelerômetro, microfone, e outros), ou por meio de sensores humanos (e.g., visão). Nesse último caso, os dados são observações subjetivas produzidas pelos usuários [Silva et al. 2014a, Burke et al. 2006].

As RSPs oferecem oportunidades sem precedentes de acesso a dados de sensoriamento em escala planetária. Essa grande quantidade de dados facilita a obtenção de informações que não estão disponíveis prontamente com a mesma abrangência global, podendo ser usadas para melhorar os processos de tomada de decisão de diferentes entidades (e.g., pessoas, grupos, serviços, aplicações).

Vale ressaltar que vários termos definidos recentemente como, por exemplo, *Humans as Data Sources* e *Ubiquitous Crowdsourcing*, refletem basicamente a definição de redes de sensoriamento participativo considerada neste documento [Srivastava et al. 2012, Mashhadi and Capra 2011, Ganti et al. 2011]. É importante também mencionar que o termo sensoriamento oportunista [Lane et al. 2010], que denomina uma forma de sensoriamento que também utiliza dispositivos móveis dos usuários no processo de sensoriamento, pode gerar confusão com o termo sensoriamento participativo. O sensoriamento participativo difere de sensoriamento oportunista principalmente pela participação do usuário, onde, neste último tipo, a etapa de coleta de dados é automatizada, sem a participação do usuário [Lane et al. 2008, Lane et al. 2010].

O sensoriamento oportunista apoia o processo de sensoriamento de uma aplicação sem demandar esforços do usuário, determinando automaticamente quando os dispositivos podem ser usados para atender às demandas específicas das aplicações. Desta forma, os aplicativos podem aproveitar as capacidades de sensoriamento de todos os dispositivos dos usuários do sistema sem a necessidade de intervenção humana neste processo [Lane et al. 2008].

2.3.2. O funcionamento de uma RSP

De forma similar às tradicionais redes de sensores sem fio (RSSFs) [Loureiro et al. 2003], o dado sensoriado em uma RSP é enviado para o servidor, ou “nó sorvedouro”, onde os dados podem ser acessados (usando, por exemplo, APIs, como a API do Instagram¹). Mas, diferentemente das RSSFs, as RSPs têm as seguintes características: (a) nós sensores são entidades móveis autônomas, ou seja, uma pessoa com um dispositivo móvel; (b) o custo da rede é distribuído entre os nós sensores, proporcionando uma escalabilidade global; (c) o sensoriamento depende da vontade das pessoas participarem desse processo; e (d) nós sensores não possuem severas limitações de energia.

As RSPs têm o potencial para complementar as RSSFs em diversos aspectos. As tradicionais redes de sensores sem fio foram projetadas para sensoriar áreas de tamanho limitado, como florestas e vulcões. Em contrapartida, as RSPs podem alcançar áreas de tamanhos variados e de larga escala, como grandes metrópoles, países ou até mesmo todo o planeta [Silva et al. 2014a]. Além disso, uma RSSF está sujeita a falhas, uma vez que o seu funcionamento depende da correta coordenação das ações dos seus nós sensores que possuem severas restrições de energia, processamento e memória. Já as RSPs são formadas por entidades autônomas e independentes, os seres humanos, o que torna a tarefa de sensoriamento mais robusto a falhas individuais. Obviamente, RSPs trazem também vários novos desafios como, por exemplo, o seu sucesso está diretamente ligado à popularização dos *smartphones*, *tablets* e serviços que operam na Internet, principalmente as mídias sociais.

A Figura 2.2 ilustra uma RSP constituída de usuários com seus dispositivos móveis enviando dados sensorizados sobre suas localizações para sistemas na nuvem. A figura mostra as atividades de compartilhamento (representados por pontos na nuvem) de quatro usuários em três instantes diferentes no tempo, rotulados como “Tempo 1”, “Tempo 2” e “Tempo 3”. Note que um usuário não participa, necessariamente, no sistema em todos os instantes. Após um certo tempo, podemos analisar estes dados de diferentes maneiras. Por exemplo, a parte inferior mais à direita da figura mostra, por meio de

¹<http://instagram.com/developer>.

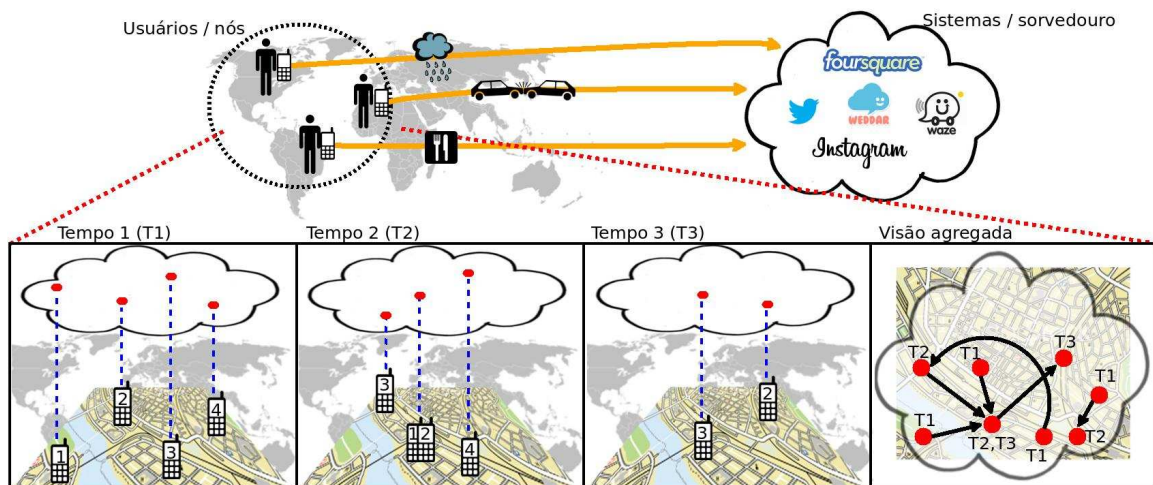


Figura 2.2. Ilustração de uma rede de sensoriamento participativo [Silva et al. 2014a].

uma visão agregada, um grafo dirigido em que os nós/vértices representam os locais onde os dados foram compartilhados e com arestas que conectam localidades que foram compartilhadas pelo mesmo usuário. Usando este grafo podemos extrair, por exemplo, padrões de mobilidade dos usuários que podem ser utilizados para efetuar um gerenciamento de carga de forma mais eficiente na infraestrutura urbana de redes sem fio. Na verdade, a descoberta de conhecimento em RSPs caminha junto com o uso da teoria de grafos/redes [Easley and Kleinberg 2010, Newman 2010, Newman 2003].

2.3.3. Exemplos de RSPs

As redes sociais baseadas em localização, que são um tipo especial de mídia social que combinam características de rede social *online*² e a possibilidade de compartilhar dados com informações espaço-temporais³, podem ser consideradas os exemplos mais populares de RSPs. É possível encontrar vários exemplos de tais sistemas em funcionamento, tais como: Waze, que serve para relatar condições de tráfego em tempo real; Foursquare, para compartilhar o local onde o usuário está visitando; e Instagram, para enviar imagens em tempo real para o sistema. Em particular, o Instagram pode ser visto como uma das mais populares RSPs atualmente, com 200 milhões de usuários [Instagram 2014]. Ao considerarmos essa rede, o dado sensoriado é uma foto de um lugar específico. Podemos extrair informação desse tipo de dado de diversas maneiras. Uma das possibilidades é visualizar em tempo real como está a situação de uma certa área da cidade. Outras possibilidades são discutidas na Seção 2.6.

Note que todos os sistemas descritos anteriormente são compostos de uma rede social *online*. No entanto, existem vários exemplos de RSPs que não contêm redes sociais *online*. Por exemplo, o Weddar⁴, para relatar condições meteorológicas, o NoiseTube⁵, para

²Plataforma virtual que constrói e reflete as relações sociais da vida real entre as pessoas.

³Tipo de dado que permite, por exemplo, a construção de serviços baseados em localização.

⁴<http://www.weddar.com>.

⁵<http://noisetube.net>.

o compartilhamento de nível de ruído em determinada região da cidade ou o Colab⁶, para o compartilhamento de problemas diversos das cidades.

Alguns outros tipos de mídia social, como o Twitter⁷, que permite aos seus usuários compartilhar atualizações pessoais em textos de até 140 caracteres, conhecidos como *tweets*, podem também ser exemplos de RSPs. Twitter é considerado um exemplo de RSP porque *tweets* podem, eventualmente, também permitir a monitorização de vários aspectos das cidades, bem como o comportamento coletivo das pessoas quase em tempo real. Por exemplo, as pessoas poderiam usar os seus dispositivos portáteis para compartilhar *tweets* com informações em tempo real sobre manifestações ou acidentes na cidade. Além desses exemplos, podemos também mencionar GarbageWatch [CENS/UCLA] para monitorar aspectos de lixo de uma cidade. Este exemplo é particularmente interessante porque ilustra que a utilização da web não é obrigatória em uma RSP. Dado de sensoriamento pode ser enviado para um aplicativo específico em execução na Internet, mas fora da Web.

2.4. Gerência de Dados Urbanos

2.4.1. Obtenção de Dados

Nesta seção apresentamos três das principais formas de obtenção de dados de RSPs. A primeira forma é através de APIs, como descrita na Seção 2.4.1.1. A segunda forma é utilizando um Web *crawler* (Seção 2.4.1.2). Por fim, apresentamos na Seção 2.4.1.3 a forma de obtenção de dados que utiliza aplicações.

2.4.1.1. Utilizando APIs

A Web está repleta de fontes de informação, o que representa uma grande oportunidade para pesquisadores de diversas áreas coletarem dados em larga escala e a partir deles extrair conhecimento [Benevenuto et al. 2011].

Algumas RSPs disponibilizam APIs que podem ser utilizadas para a extração de dados. Através desse processo, é possível obter dados de RSPs que podem ser utilizados em outras aplicações ou em análises específicas. Várias RSPs populares, como Foursquare, possuem APIs de acesso aos dados compartilhados pelos usuários. Entretanto, é comum existirem regras diferentes para a sua utilização.

Podemos citar duas formas de funcionamento de APIs: (1) baseadas em *streaming*; (2) baseadas em requisições. O método baseado em *streaming* permite coletar em tempo (quase) real os dados que são publicados em uma determinada RSP. A API de *streaming* do Twitter⁸, por exemplo, permite coletar em tempo (quase) real *tweets* públicos à medida que são publicados. Já o método baseado em requisições disponibilizam dados atendendo a uma solicitação específica, por exemplo, todos os seus últimos 10 *tweets*. Tanto métodos baseados em *streaming* quanto métodos baseados em requisições podem sofrer limitações na obtenção do volume de dados. Por exemplo, o Flickr permite 5000 requisições por hora em sua API, já a API de *streaming* do Twitter pode não fornecer todos os dados comparti-

⁶www.colab.re.

⁷<http://www.twitter.com>.

⁸<http://www.twitter.com>.

lhados⁹. Isso pode inviabilizar alguns tipos de análises que necessitam de um número maior de amostras no período de uma hora, por exemplo.

De fato, o uso de APIs é uma forma bastante popular para a obtenção de dados. Dados obtidos através de APIs como a do Twitter foram utilizados das mais variadas formas, desde medir a influência de usuários na rede [Cha et al. 2010], até a previsão de terremotos [Sakaki et al. 2010a].

Um exemplo de uso da API de *streaming* do Twitter, escrito na linguagem de programação Python e utilizando a biblioteca TwitterAPI¹⁰, é mostrado no algoritmo mostrado na Figura 2.3. Nesse algoritmo fazemos acesso aos *tweets* buscando pela palavra-chave "4sq". Como podemos ver, em poucas linhas de código é possível coletar facilmente dados do Twitter. A Figura 2.4 ilustra esse resultado com dois *tweets* de resposta: tweet1 e tweet2. Esses *tweets* foram retornados no padrão JSON.

```
#Biblioteca que facilita a interação com a API DO Twitter
from TwitterAPI import TwitterAPI

#Um registro no website da API fornece as credenciais indicadas aqui
twitter_api = TwitterAPI(consumer_key='xxxxxx', consumer_secret='xxxxxx',
» access_token_key='xxxxxx', access_token_secret='xxxxxx')

filters = {"track": ["4sq"]} #palavra que deseja buscar em tweets

stream = twitter_api.request('statuses/filter', filters)

for item in stream.get_iterator():
    print item #exibe todo conteúdo do tweet
```

Figura 2.3. Exemplo de obtenção de dados do Twitter.

Existem RSPs que possuem APIs, mas com acesso bastante restrito aos dados. Esse é o caso do Foursquare, pois poucos dados são possíveis de serem coletados sem a autorização do usuário. A maioria dos dados disponíveis através dessa API são referentes aos locais, como dicas, listas, localização e fotos.

Essas limitações estimulam a obtenção de dados de forma indireta ou alternativa. Por exemplo, em [Silva et al. 2014c] os autores obtiveram dados sobre os *check-ins* do Foursquare através de mensagens públicas compartilhadas no Twitter. Isso é possível, pois o Foursquare possibilita aos usuários anunciarem seus *check-ins* nesse sistema. Esse procedimento é mostrado na Figura 2.4. Essa figura ilustra um *tweet* proveniente do Foursquare que possui uma URL que representa uma página Web com mais informações sobre o *check-in* anunciado. No exemplo da figura, a página representa um *check-in* dado em um restaurante. Para obter mais dados sobre o *check-in* disponíveis nessa página é utilizada a técnica de coleta Web *crawler*, apresentada a seguir.

2.4.1.2. Utilizando um Web Crawler

Nem todas as fontes de dados disponíveis na Internet fornecem acesso direto a esses dados através de APIs. Por isso é necessário utilizar outras formas de obtenção de dados. Uma

⁹Ao solicitar dados os dados compartilhados no Twitter estima-se que será entregue 1%.

¹⁰<https://github.com/geduldig/TwitterAPI>.



Figura 2.4. Etapas de coleta de dados do Foursquare através de tweets.

dessas alternativas é a chamada *Web crawler*, que são programas que analisam páginas Web em busca de dados relevantes [Benevenuto et al. 2011]. Um *Web crawler* funciona como um robô que acessa páginas Web predeterminadas e recupera dados a partir dessas páginas.

A coleta através de *Web crawlers* depende da estrutura da fonte da qual desejamos obter dados, bem como da abordagem utilizada. A estrutura da fonte é onde os dados que queremos extrair estão disponibilizados; nas páginas Web, por exemplo, são *tags* HTML que apresentam os dados ao usuário. Com isso a construção de um *Web crawler* demanda tipicamente a mineração de texto para a extração dos dados necessários na página Web estudada. No entanto, outras formas não convencionais de extração de dados usando páginas Web são possíveis. Por exemplo, em [Tostes et al. 2014] os autores construíram um *Web crawler* para coletar informações de tráfego tirando fotos (*screenshots*) de mapas com essas informações, como as disponíveis no Bing Maps¹¹. Mais informações sobre esse procedimento são fornecidas em [Tostes et al. 2014].

```
import urllib
#url obtida através do tweet Saída do código
url = "http://4sq.com/1mhXj6u"
pagina = urllib.urlopen(url).read()
print pagina
```

```
{
  "venue": {
    "name": "Başkent Cafe",
    "stats": {
      "checkinsCount": 7565,
      "usersCount": 4097,
      "city": "Pendik",
      "crossStreet": "Ankara Caddesi",
      "lng": 29.23510950768732,
      "context": "Turkey",
      "state": "Istanbul",
      "neighborhood": "Pendik",
      "address": "Ankara Cad. Pendik",
      "cc": "TR",
      "id": "59acf230e4b9fbae226eac6",
      "canonical": "http://4sq.com/1mhXj6u",
      "canonical": "http://4sq.com/1mhXj6u",
      "pluralName": "Cafés",
      "name": "Café"
    }
  }
}
```

Figura 2.5. Exemplo de coleta de uma página de um *check-in* do Foursquare usando a biblioteca URLLIB da linguagem Python.

A Figura 2.5 ilustra um código de um simples *Web crawler* em Python. Esse *Web crawler* utiliza a biblioteca URLLIB para realizar a coleta de uma página referente a um *check-in* do Foursquare (a URL utilizada foi a encontrada no processo ilustrado na Figura 2.4). O resultado parcial da saída do código é também ilustrado nessa figura. Repare que foi assinalada uma informação relevante sobre o *check-in*: o tipo do local, informação que não é acessível através da API do Foursquare.

2.4.1.3. Utilizando Aplicações

Uma outra alternativa para a coleta de dados é a criação de aplicações em plataformas já existentes. Alguns sistemas populares, como Facebook, Instagram e Runkeeper, permitem a criação de aplicativos dentro de suas plataformas. Com isso, desenvolvedores podem oferecer serviços utilizando dados que são compartilhados nesses aplicativos.

¹¹<http://www.bing.com/maps>.

O Facebook, por exemplo, não permite a coleta de informações direta de seus usuários por APIs ou *Web crawlers*. No entanto, como permitem a criação de aplicações é possível obter dados compartilhados por seus usuários. Quando o usuário do Facebook instala um aplicativo e autoriza a leitura de seus dados, o desenvolvedor da aplicação pode ler e armazenar diversos dados, como os disponibilizados pelos usuários, por exemplo, o conteúdo compartilhado com seus amigos.

Em [Nazir et al. 2008] os autores utilizaram essa abordagem de coleta de dados. Eles criaram aplicações do Facebook especificamente para coletar dados que possibilitassem o estudo do comportamento das pessoas que fazem uso desse tipo de aplicação. Outro exemplo foi o aplicativo utilizado em [Youyou et al. 2015]. Os autores criaram uma aplicação no Facebook que captura os últimos *likes*¹² do usuário para traçar um perfil de personalidade.

É possível ainda a criação de aplicações que não dependem de plataformas de sistemas existentes. Esse foi o caso da RSP NoiseTube [Maisonneuve et al. 2009]. Os autores criaram uma aplicação que permite aos usuários reportarem níveis de ruído na cidade. Esses dados permitem identificar, por exemplo, quais áreas da cidade o nível de ruído está acima dos limites estipulados por lei. Outro exemplo é o Colab, citado anteriormente. Recentemente foi proposta uma plataforma chamada *ohmage*¹³ para facilitar a construção de aplicações que desejam utilizar dados de sensoriamento participativo.

Dessa forma, de posse de dados de RSPs, que podem ser obtidos por alguma dessas maneiras citadas, podemos extrair conhecimento de diversas formas, como é melhor discutido nas próximas seções

2.4.2. Reformatação e Limpeza dos Dados

Os dados brutos (sem tratamento) de RSPs podem não estar em um formato conveniente para executar uma análise particular. Dependendo do tipo do dado é possível encontrar erros semânticos, entradas ausentes ou formatação inconsistente. Nestes casos, eles precisam ser “limpos” antes da análise.

Programadores reformatam e limpam dados escrevendo *scripts* ou editando manualmente dados, por exemplo, numa planilha. Estas tarefas tendem a ser demoradas e tediosas, pois são tarefas inevitáveis que não produzem novos conhecimentos. No entanto, a tarefa de reformatação de dados e de limpeza pode proporcionar ideias sobre quais suposições são seguras de serem feitas sobre os dados, quais peculiaridades existentes no processo de coleta e quais os modelos e análises são apropriados para serem aplicados.

A integração de dados é um desafio relacionado nesta fase, mas é discutido na Seção 2.8. Muitas vezes, a programação envolve trabalhar os dados em diferentes ferramentas, convertendo de um formato de dados para outro, extraindo dados numéricos a partir de um texto, e administrando experimentos numéricos que envolvem um grande número de arquivos de dados e diretórios. Tais tarefas são muito mais rápidas para serem realizadas em uma linguagem como Python do que em Java ou C++, discutimos mais sobre isso na Seção 2.7.10.

¹²Um *like* é uma interação do usuário com o Facebook em que ele demonstra que gostou de um item compartilhado.

¹³<http://ohmage.org>.

2.4.3. Modelagem de Dados

Os dados gerados em espaços urbanos são geralmente associados com uma propriedade espacial ou espaço-temporais. Por exemplo, a localização de estabelecimentos são dados espaciais; dados meteorológicos e consumo de energia são dados temporais (também chamados de séries temporais, ou *stream*). Já os dados de RSPs possuem propriedades espaço-temporais simultaneamente.

Existem vários formatos de dados para modelar dados de RSPs, sendo bastante popular o uso de grafos [Zheng et al. 2014a]. Na Figura 2.2 ilustramos a criação de um grafo dirigido em que os nós representam os locais onde os dados foram compartilhados e com arestas que conectam localidades que foram compartilhadas pelo mesmo usuário. Usando este grafo podemos extrair diversas informações. De fato, a descoberta de conhecimento em dados de RSPs caminha junto com uma vasta gama de estudos que utilizam a teoria de grafos [Newman 2010, Newman 2003, Easley and Kleinberg 2010]. Como mostramos na Seção 2.6, técnicas bem conhecidas utilizadas para análise de grafos podem ser aplicadas diretamente para estudar grafos derivados de RSPs que refletem condições das cidades.

Alguns dos desafios sobre questões da dinâmica temporal relacionados com dados urbanos das RSPs são discutidos na Seção 2.8.

2.5. Análise de Dados Urbanos Provenientes de RSPs

Como os dados urbanos provenientes de RSPs podem ser muito complexos, um passo fundamental em qualquer investigação é caracterizar os dados coletados, a fim de entender suas limitações e utilidade. Com isso, nesta seção vamos estudar as propriedades de três RSPs para compartilhamento de localização (Foursquare, Gowalla e Brightkite¹⁴); uma RSP para compartilhamento de fotos (Instagram); bem como uma RSP para compartilhamento de alerta de trânsito (Waze).

2.5.1. Descrição dos Dados

Nesta subseção apresentaremos todos os *datasets* aqui considerados. Todos os dados foram coletados através do Twitter. Além de *tweets* de texto simples, os usuários também podem anunciar dados a partir de uma integração com outros serviços, como o Instagram, Foursquare e Waze. Neste caso, fotos do Instagram, *check-ins* do Foursquare ou alertas do Waze anunciadas no Twitter passam a ficar disponíveis publicamente, o que por padrão não acontece quando o dado é publicado unicamente nos sistemas analisados.

Alguns dos *datasets* que foram analisados: Foursquare1 (≈ 5 milhões de *check-ins* em abril de 2012 - 1 semana); Foursquare2 (≈ 12 milhões de *check-ins* entre fev2010-jan2011); Foursquare3 (≈ 4 milhões de *check-ins* em maio de 2013 - 2 semanas); Gowalla (≈ 6 milhões de *check-ins* entre fev2009-out2010); Brightkite (≈ 4 milhões *check-ins* entre abr2008-out2010); Instagram1 (≈ 2 milhões de fotos entre jun2012-jul2012); Instagram2 (≈ 2 milhões de fotos em maio 2013 - 2 semanas); Waze (+212 mil alertas entre dez2012-jun2013). Como podemos ver, os dados refletem diferentes períodos. Além disso, os *datasets* incluem uma quantia bastante significativa de dados: mais de 30 milhões de registros considerando todas as fontes.

¹⁴As RSPs para compartilhamento de localização Gowalla e Brightkite não estão mais em funcionamento.

Cada dado sensoriado (foto, *check-in* ou alerta) é composto de coordenadas GPS (latitude e longitude), do horário do compartilhamento do dado e do ID do usuário compartilhador. O *dataset* Foursquare1 possui informações extras sobre o tipo de local: categoria (por exemplo, comida) e um identificador do local. Mais informações sobre os *datasets* e como eles foram obtidos podem ser encontradas em [Cheng et al. 2011, Silva et al. 2012, Silva et al. 2013c, Silva et al. 2013d, Silva et al. 2013e].

2.5.2. Cobertura da Rede

Nesta seção, estudamos a cobertura das RSPs analisadas em diferentes granularidades espaciais, começando por todo o planeta, depois cidades e, por fim, áreas específicas de uma cidade. A primeira constatação ao analisar esses dados é que a cobertura é bastante abrangente e tem escala planetária [Cheng et al. 2011, Silva et al. 2013a, Silva et al. 2013e].

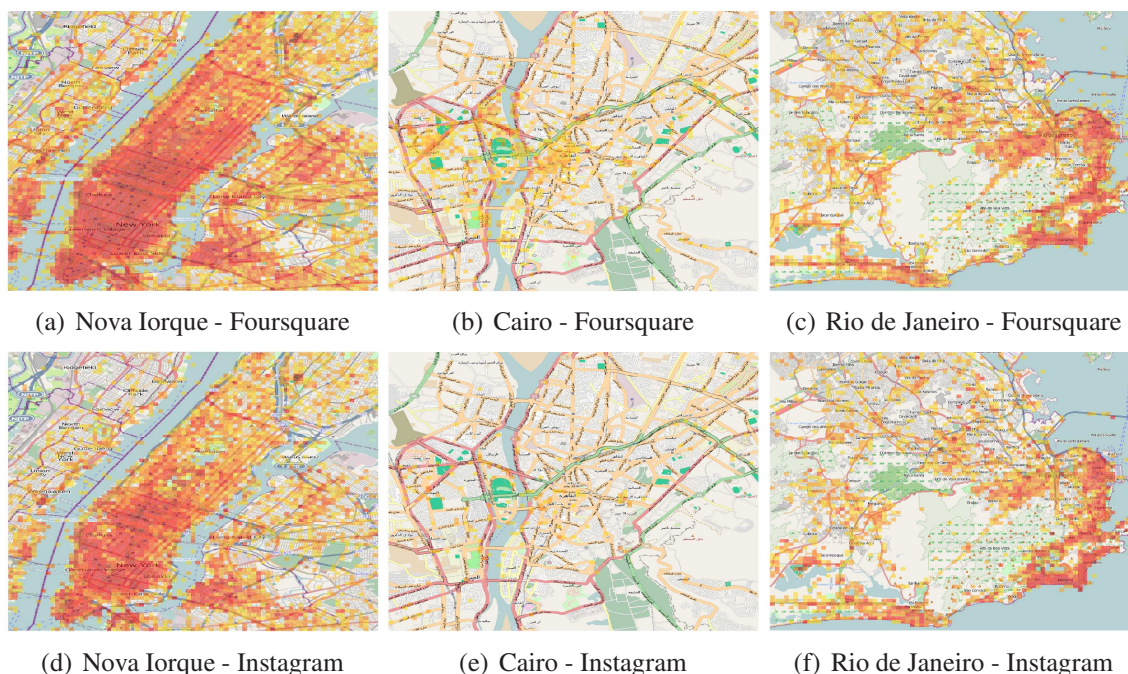


Figura 2.6. Cobertura espacial da RSP do Foursquare e Instagram em 3 cidades populosas ao redor do mundo [Silva et al. 2013a, Silva et al. 2013c].

Avaliamos agora a participação dos usuários em diversas cidades grandes, localizadas em regiões distintas, mostrando os resultados de algumas delas: Nova York, Rio de Janeiro e Cairo. A figura 2.6 mostra o mapa de calor da atividade de sensoriamento para cada uma dessas cidades. Mais uma vez, cores mais escuras representam um maior número de fotos em determinada área. Observamos uma alta cobertura para algumas cidades, como mostrado nas Figuras 2.6a e 2.6d (Nova York). No entanto, como podemos observar nas Figuras 2.6b e 2.6e, o sensoriamento no Cairo, que também possui um número elevado de habitantes, é significativamente mais baixo. Tamaña diferença na cobertura pode ser explicada por diversos fatores. Além dos aspectos econômicos, diferenças na cultura dos habitantes desta cidade, quando comparadas com as culturas presentes nas outras cidades estudadas, podem ter um impacto significativo na adoção e uso desses sistemas considerados [Barth 1969].

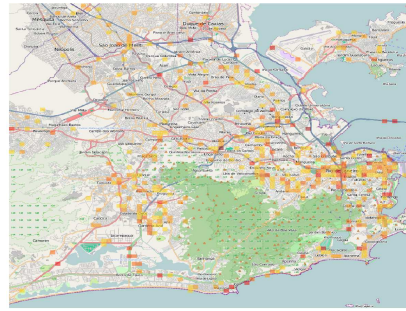


Figura 2.7. Cobertura espacial da RSP para compartilhamento de alerta de trânsito no Rio de Janeiro [Silva et al. 2013e].

Além disso, pode-se observar que a cobertura em algumas cidades, como no Rio de Janeiro (Figuras 2.6c e 2.6f), é bem mais heterogênea quando comparada com a cobertura de Nova York. Isto ocorre, provavelmente, por causa dos aspectos geográficos particulares dessas cidades, ou seja, grandes áreas verdes e grandes porções d'água. O Rio de Janeiro tem a maior floresta urbana do mundo, localizada no meio da cidade, além de muitas colinas de difícil acesso humano. Estes aspectos geográficos limitam a cobertura do sensoriamento. Além disso, os pontos de interesse público, tais como pontos turísticos e centros comerciais, são distribuídos de forma desigual pela cidade. Há grandes áreas residenciais com poucos pontos desse tipo, enquanto outras áreas têm grande concentração dos mesmos.

A cobertura espacial dos dados da RSP para alertas de trânsito não é tão abrangente, como das RSPs para compartilhamento de localização e de foto. Isso pode ser observado na Figura 2.7, que mostra o número de alertas em diferentes regiões do Rio de Janeiro por um mapa de calor. Um fator que pode ajudar a explicar isso é a população de usuários do *dataset* de alertas de trânsito, que é menor do que nos outros casos estudados. Outro fator é que os usuários podem ter menos oportunidades para compartilhar alertas de trânsito em comparação com oportunidades para compartilhar fotos ou *check-ins*.

Como a atividade de participação pode ser bastante heterogênea dentro de uma cidade, analisamos a cobertura de RSPs em áreas específicas de uma cidade. Para ter um ID de uma área específica da cidade para os *datasets* do Instagram e Waze, propomos dividir a área das cidades em espaços retangulares menores, como em uma grade¹⁵. Chamaremos cada área retangular de uma *área específica* dentro de uma cidade. Consideramos que uma área específica possui a seguinte delimitação: $1 \cdot 10^{-4}^\circ$ (latitude) \times $1 \cdot 10^{-4}^\circ$ (longitude). Isso representa uma área de aproximadamente 8×11 metros em Nova Iorque e 10×11 metros no Rio de Janeiro. Para outras cidades, as áreas também podem variar um pouco, mas não a ponto de afetar significativamente as análises realizadas.

A Figura 2.8 apresenta a função de distribuição acumulada complementar (*complementary cumulative distribution function* - CCDF) do número de dados compartilhados (*check-ins*, fotos ou alertas) por área específica de todas as localidades em nossos *datasets*. Primeiramente, observe que, em ambos os casos, uma lei de potência¹⁶ descreve bem

¹⁵Note que nas áreas selecionadas não são consideradas fronteiras.

¹⁶Matematicamente, uma quantidade x segue uma lei de potência se ela pode ser obtida de uma distribuição de probabilidade $p(x) \propto x^{-\alpha}$, onde α é um parâmetro constante conhecido como expoente ou parâmetro escalar, e é um valor tipicamente entre $2 < \alpha < 3$ [Clauset et al. 2009].

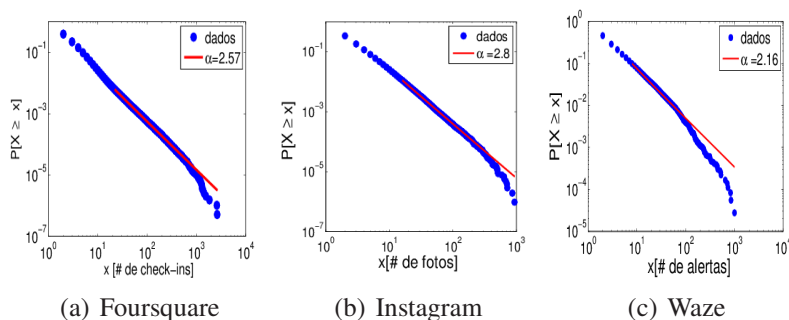


Figura 2.8. Distribuição do número de dados em áreas específicas (escala log-log) [Silva et al. 2013a, Silva et al. 2013a, Silva et al. 2013e].

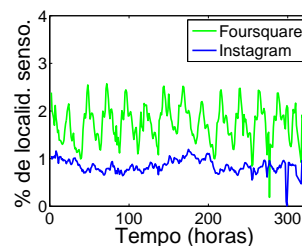


Figura 2.9. Porcentagem de áreas específicas sensoriadas ao longo do tempo [Silva et al. 2014a].

esta distribuição. Isso implica que, na maioria das áreas específicas, há poucos dados compartilhados, enquanto existem algumas poucas áreas com centenas de dados compartilhados. Estes resultados estão consistentes com os resultados apresentados em [Noulas et al. 2011a], trabalho que estudou a participação de usuários em sistemas de compartilhamento de localização. Nos sistemas analisados, é natural que algumas áreas possuam mais atividade que outras. Por exemplo, em áreas turísticas o número de fotos compartilhadas tende a ser maior do que em um supermercado, apesar de um supermercado ser geralmente um local bastante popular. Se uma determinada aplicação requer uma cobertura mais abrangente, é necessário incentivar os usuários a participarem em locais que eles usualmente não o fariam. Micro pagamentos ou sistemas de pontuação são exemplos de alternativas que poderiam funcionar nesse caso. Discutimos essas oportunidades na Seção 2.8.3.

Mostramos que uma RSP pode ter uma cobertura em escala planetária. No entanto, essa cobertura pode ser bastante desigual, em que grandes áreas ficam praticamente descobertas. Com isso em mente, a Figura 2.9 mostra a porcentagem de locais distintos onde os usuários compartilharam dados em um determinado intervalo de tempo no Instagram e Foursquare¹⁷, que possuem 598.397 e 725.419 locais, respectivamente. O percentual máximo de locais distintos compartilhados por hora é inferior a 3% para todos os sistemas. Isto indica que a cobertura instantânea destas RSPs é muito limitada quando consideramos todas as localidades que poderiam ser sensoriadas no planeta (considerando todas as localidades já sensoriadas pelo menos uma vez). Em outras palavras, a probabilidade de uma área específica aleatória ser sensoriada em um horário aleatório é bem baixa.

2.5.3. Rotinas e o Compartilhamento de Dados

Analisamos agora como a rotina dos humanos afeta o compartilhamento dos dados. A Figura 2.10 mostra o padrão semanal de compartilhamento de dados em todos os tipos de RSPs analisadas¹⁸. Como esperado, os dados compartilhados nas RSPs apresentam um padrão diurno, o que implica que durante a madrugada a atividade de sensoriamento é bastante baixa.

¹⁷Consideramos os *datasets* Instagram2 e Foursquare3, pois representam o mesmo intervalo de tempo.

¹⁸O horário do compartilhamento foi normalizado de acordo com o local onde o dado foi compartilhado, utilizando para isso a informação geográfica do local.

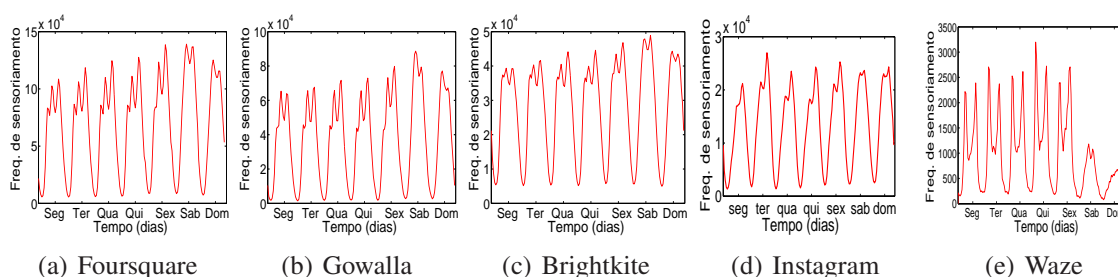


Figura 2.10. Padrão do compartilhamento de fotos durante os dias da semana [Silva et al. 2013a, Silva et al. 2013c, Silva et al. 2013e].

Considerando dias de semana, é possível observar um ligeiro aumento da atividade ao longo da semana, com poucas exceções quando há um pico de atividade. O trabalho [Cheng et al. 2011], que analisou sistemas para compartilhamento de localização, foi observado esse mesmo comportamento, sem nenhum dia como exceção.

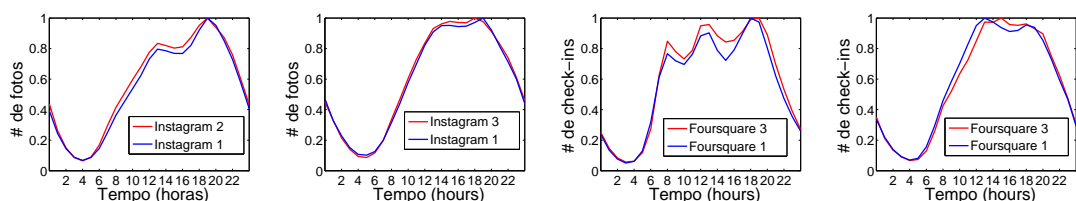
Podemos ainda observar que alguns picos de atividade variam ao longo do dia de acordo com o propósito da RSP. Como podemos ver na Figura 2.10, na RSP para compartilhamento de localizações (Figuras 2.10a–c) existem três picos evidentes por volta da hora do café da manhã, almoço e jantar. Isso também foi observado em [Cheng et al. 2011]. Já na RSP para compartilhamento de fotos (Figura 2.10d) existem apenas dois picos evidentes, que ocorrem por volta da hora do almoço e jantar. E no caso da RSP para compartilhamento de alertas de trânsito (Figura 2.10e) também existem dois picos evidentes, um por volta de 7:00 e 8:00 da manhã e outro por volta de 6:00 da tarde, coincidindo com horários típicos de maior intensidade no trânsito.

Analisando os diferentes padrões de comportamento para dias de semana e final de semana podemos observar que o padrão é significativamente diferente. Note que os picos observados nos dias de semana não são evidentes nos finais de semana. A falta de rotina bem definida nos fins de semana é uma das possíveis explicações para esse fato. Além disso, as diferenças entre dias de semana e final de semana possuem relação com o tipo de sistema analisado. Por exemplo, como nos fins de semana muitas pessoas não precisam dirigir, é natural esperar um volume menor de dados no Waze.

A Figura 2.11 mostra o padrão temporal de compartilhamento para o Instagram e o Foursquare considerando todos os *datasets*. Essa figura apresenta o número médio de dados compartilhados por hora durante, durante os dias de semana (de segunda a sexta-feira) e também durante o fim de semana (sábado e domingo). Surpreendentemente, vemos o mesmo padrão de compartilhamento para cada curva é muito semelhante, apesar do enorme intervalo entre as coletas (aproximadamente um ano). Isso acontece para os dias de semana e fins de semana, sugerindo que o comportamento do usuário em ambos os sistemas tende a se manter consistente ao longo do tempo. Esse é um resultado interessante e importante, pois mostra que podemos usar diferentes *datasets* para propósitos similares.

Mostramos agora como as rotinas impactam no comportamento de compartilhamento durante a semana. Para essa análise, consideramos os *datasets* do Instagram e Foursquare para Nova York, São Paulo e Tóquio. Os resultados são mostrados na Figura 2.12¹⁹.

¹⁹Cada curva é normalizada pelo número máximo de conteúdo compartilhado em uma região específica

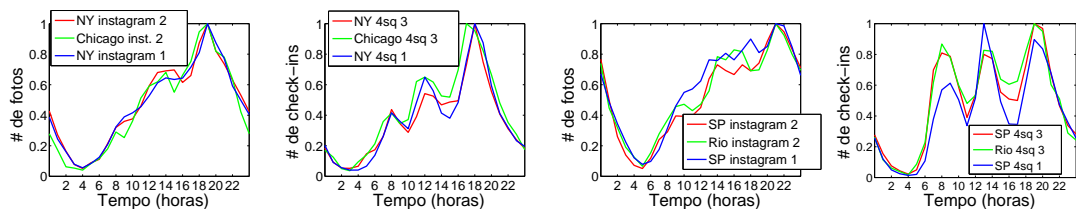


(a) Instagram – dia de semana (b) Instagram – fim de semana (c) Foursquare – dia de semana (d) Foursquare – fim de semana

Figura 2.11. Padrão de compartilhamento temporal no Instagram e Foursquare [Silva et al. 2013d].

Em todas as figuras nós exibimos dados dos *datasets* do mesmo período (Instagram2 e Foursquare3) para duas cidades do mesmo país, e dados de um *dataset* com período anterior (Instagram1 e Foursquare1) para uma dessas cidades, como uma referência de comparação.

Primeiramente, observe a distinção entre as curvas de cada cidade no mesmo sistema (por exemplo, Instagram, Figuras 2.12a, c, e) e também em diferentes sistemas (por exemplo, as Figuras 2.12a e 2.12b para Nova Iorque). Em seguida, observe que o padrão de compartilhamento para cada cidade no mesmo país é bastante semelhante, o que pode ser consequência dos padrões culturais dos habitantes desses países. Isso representa, de certa maneira, uma assinatura de aspectos culturais, o que ilustra, mais uma vez, o potencial desse tipo de dado para o estudo de dinâmica de cidades e do comportamento social urbano.



(a) Nova Iorque – Instagram (b) Nova Iorque – Foursquare (c) São Paulo – Instagram (d) São Paulo – Foursquare

Figura 2.12. Padrão de compartilhamento temporal do Instagram e Foursquare para Nova Iorque, São Paulo e Tóquio durante dias de semana [Silva et al. 2013d].

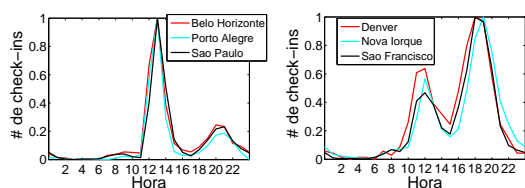
Podemos ainda analisar classes de locais específicos. As Figuras 2.13a e 2.13b²⁰ mostram o número de *check-ins* realizados em restaurantes ao longo das horas do dia, durante os dias de semana, em diferentes cidades do Brasil e dos Estados Unidos. Estes resultados capturaram diferenças importantes entre as culturas dos dois países: enquanto o jantar é a refeição principal para os americanos, o almoço desempenha um papel mais importante nos hábitos alimentares dos brasileiros.

2.5.4. Comportamento dos Nós

Nesta seção analisamos o desempenho dos nós da RSP (i.e., dos usuários) quanto ao compartilhamento de dados. A Figura 2.14 mostra a distribuição do número de dados (fotos e

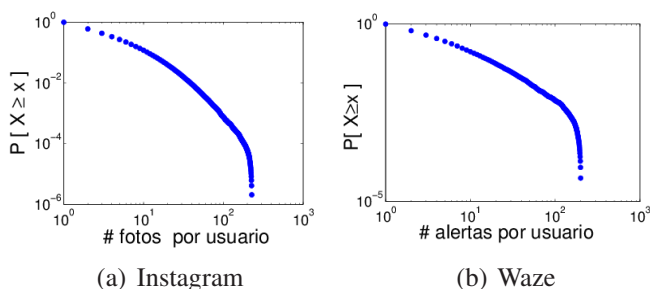
representando a cidade.

²⁰Os valores são normalizados pelo valor máximo encontrado em qualquer hora para a cidade específica.



(a) Cidades brasileiras (b) Cidades dos EUA

Figura 2.13. Número médio de *check-ins* em restaurantes durante dias de semana ao longo das horas do dia [Silva et al. 2014a].



(a) Instagram

(b) Waze

Figura 2.14. Distribuição do número de dados compartilhadas pelos usuários [Silva et al. 2013c, Silva et al. 2013e].

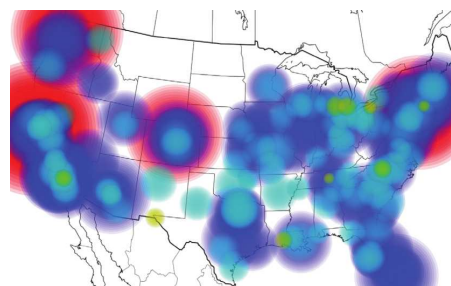


Figura 2.15. Resultado da métrica raio de giro [Cheng et al. 2011].

alertas) compartilhados por cada usuário da nossa base de dados. Como podemos observar, a distribuição possui cauda pesada, o que significa que a participação dos usuários pode ser muito desigual. Por exemplo, aproximadamente 40% dos usuários contribuíram com apenas uma foto no período considerado, enquanto somente 17% e 0,1% dos usuários contribuíram com mais que 10 e 100 fotos, respectivamente. É natural que essa variabilidade aconteça por diversos motivos. Por exemplo, alguns usuários podem dar mais importância para quesitos de privacidade do que outros. Uma cauda pesada também é observada na distribuição do número de *check-ins*, como foi mostrado em [Noulas et al. 2011a]. Cerca de 20% dos usuários realizaram apenas um *check-in*, 40 % acima de 10, ao passo que cerca de 10 % realizaram mais de 100 *check-ins*.

Além disso, em [Silva et al. 2013c, Silva et al. 2013e, Silva et al. 2013a] mostramos que há momentos em que muitos dados são compartilhados em intervalos de poucos minutos e momentos em que não há compartilhamento por horas. Isso pode indicar que a maioria do compartilhamento de dados acontece em intervalos específicos, provavelmente relacionados ao ciclo circadiano (ou rotina) das pessoas. Por exemplo, o compartilhamento de fotos em restaurantes tende a acontecer mais nos horários de almoço e jantar. Aplicações baseadas nesse tipo de sensoriamento devem considerar que a participação do usuário pode variar significativamente ao longo do tempo.

Observamos ainda que uma fatia significativa dos usuários realiza compartilhamento consecutivo de fotos num curto intervalo de tempo. Por exemplo, cerca de 20% de todo o compartilhamento de fotos observado acontece em até 10 minutos. Isso sugere que os usuários tendem a compartilhar mais de uma foto na mesma área. Em [Noulas et al. 2011a] os autores também observaram que uma parcela significativa dos *check-ins* no Foursquare

são realizados dentro de um curto intervalo de tempo. Por exemplo, mais do que 10% de *check-ins* ocorrem dentro de 10 minutos.

Em [Cheng et al. 2011] os autores analisaram *check-ins* compartilhados em vários serviços de compartilhamento de localização. Eles descobriram que os usuários possuem padrões simples e reproduzíveis, e também que o status social, além de fatores geográficos e econômicos, colaboram com a mobilidade.

Para fazer essa análise os autores usaram três propriedades estatísticas para estudar e modelar padrões de mobilidade humana: *deslocamento* (*displacement*); *raio de giro* (*radius of gyration*); e *probabilidade de retorno* (*returning probability*). Para ilustrar um de seus resultados, a Figura 2.15 mostra o raio médio de giro dos usuários em grandes cidades (com mais de 100.000 habitantes) nos EUA. As bolhas vermelhas²¹ são cidades com um raio de giro maior do que 500 milhas; as azuis são cidades com um raio maior do que 250 milhas; as de cor ciano possuem um raio maior do que 125 milhas; e as amarelas são o resto das grandes cidades analisadas. Usuários em cidades costeiras tendem a ter um raio maior de giro do que os usuários em cidades do interior, e as pessoas em estados centrais tendem a ter um alto raio de giro devido a viagens de longa distância para o litoral [Cheng et al. 2011].

Da mesma forma, em [Cho et al. 2011] os autores investigaram padrões de movimentos e como os laços sociais podem impactar nesses movimentos. Os autores observaram que viagens de curta distância são espacialmente e temporalmente periódicas e não são afetadas pela estrutura de rede social, enquanto as viagens de longa distância são mais influenciadas por laços da rede social.

2.5.5. Considerações Finais

Identificamos várias propriedades de RSPs em comum: (i) escala planetária; (ii) frequência altamente desigual de compartilhamento de dados, tanto espacialmente quanto temporalmente, o que é altamente correlacionado com a rotina típica das pessoas; (iii) a participação do usuário em relação ao número de dados compartilhados e onde esses dados são compartilhados pode variar significativamente; (iv) o padrão temporal de compartilhamento não varia consideravelmente ao longo do tempo para o mesmo tipo de sistema.

As propriedades identificadas revelam o potencial de RSPs para conduzir vários estudos sobre a dinâmica da cidade e do comportamento social urbano. Além disso, o entendimento do comportamento do usuário é o primeiro passo para modelá-lo. Com modelos que explicam o comportamento do usuário podemos fazer previsões de ações e desenvolver melhores sistemas para planejamento de capacidade de carga do sistema.

Na Seção 2.7.10 são discutidas algumas das principais tecnologias e ferramentas para a análise de dados de RSPs, que podem ser bastante úteis em futuras análises de outros dados.

2.6. Aplicações e Serviços Relacionados ao Estudo de Sociedades Urbanas

Nesta seção, discutiremos as abordagens e modelos utilizados em diversas aplicações e serviços relacionados ao estudo da dinâmica da cidade e do comportamento social urbano utilizando dados de RSPs. Para a construção de novos serviços e aplicações nessa área é de

²¹Consultar a versão digital disponível online em cores.

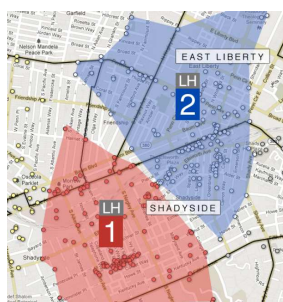


Figura 2.16. “Livehoods” encontrados em Nova Iorque [Cranshaw et al. 2012].

suma importância conhecer as propriedades dos dados da RSP em estudo.

Os estudos mostrados aqui foram agrupados em cinco classes: Funcionamento de Cidades (Seção 2.6.1); Mobilidade Urbana (Seção 2.6.2); Padrões Sociais, Econômicos e Culturais (Seção 2.6.3); Detecção de Eventos e Interesses (Seção 2.6.4); e Problemas das Cidades (Seção 2.6.5).

2.6.1. Funcionamento de Cidades

As informações obtidas a partir de RSPs têm o poder de mudar os nossos limites físicos percebidos, bem como ajudar a compreender melhor a dinâmica de cidades. Esta seção concentra na apresentação de estudos nessas direções.

Usando dados do Foursquare, em [Cranshaw et al. 2012] os autores propuseram um modelo para identificar regiões distintas de uma cidade que refletem padrões atuais de atividades coletivas, apresentando novos limites para os bairros. A ideia é expor a natureza dinâmica das áreas urbanas locais, considerando a proximidade espacial (derivado de coordenadas geográficas) e proximidade social (derivado da distribuição de check-ins) de locais.

Para isso, os autores utilizaram dados do Foursquare e desenvolveram um modelo que agrupa locais semelhantes considerando características sociais e espaciais. Cada *cluster* representa diferentes fronteiras geográficas dos bairros. O método de agrupamento utilizado é uma variação do agrupamento espectral proposto por [Ng et al. 2002].

A Figura 2.16 mostra dois *clusters* (ou “livehoods”, nome usado pelos autores), encontrados em Nova Iorque, representados pelos números 1 e 2. Nessa figura as linhas pretas indicam os limites oficiais da cidade. Veja que os limites dos *clusters* são bastante diferentes. Para tentar validar esses resultados os autores usaram resultados de entrevistas com moradores da cidade. De acordo com as respostas coletadas, esses e outros *clusters* eram esperados.

Em [Noulas et al. 2011b] os autores propuseram uma abordagem para classificar áreas e usuários de uma cidade usando categorias de locais do Foursquare. Isso poderia ser usado para identificar as comunidades de usuários que visitam categorias semelhantes de lugares, útil para sistemas de recomendação, ou na comparação de áreas urbanas dentro e entre as cidades. A abordagem utilizada é baseada em algoritmo de agrupamento espectral [Ng et al. 2002].

Mais especificamente, os autores consideram a atividade dos usuários do Foursquare

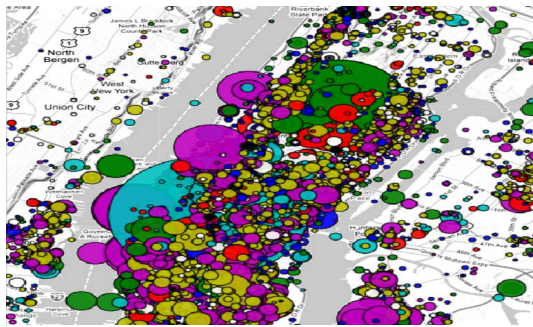


Figura 2.17. Atividade dos usuários do Foursquare para Nova Iorque. A&E (vermelho); Edu (preto); Outd (verde); NL (magenta); Shop (branco); and Trvl (ciano) [Noulas et al. 2011b].

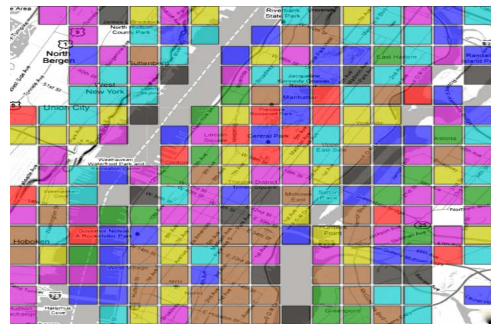


Figura 2.18. Visualização do agrupamento espectral. Cada cor simboliza um *cluster*. [Noulas et al. 2011b].

para Nova Iorque, como mostrado pela Figura 2.17. Nessa figura, um círculo representa um local e seu raio a popularidade em termos de número de check-ins. Cada cor corresponde a uma das oito categorias gerais introduzidas pelo Foursquare. Essa figura destaca a diversidade da atividade humana sobre a área considerada.

Em seguida, os autores dividiram uma cidade para ser analisada em áreas de tamanhos iguais, cada uma delas será um dado de entrada para o algoritmo de agrupamento. Para cada área é calculada a atividade realizada pelos usuários com base nas visitas em locais dessa região. Com isso, calcula-se a semelhança entre duas áreas utilizando a similaridade do cosseno (*cosine similarity*), entre as atividades representadas. Após esse processo, os autores realizam um agrupamento espectral. O resultado desse processo para a cidade de Nova Iorque é mostrado na Figura 2.18.

Em [Silva et al. 2014d] propusemos uma técnica chamada *City Image*, que fornece um resumo visual da dinâmica da cidade com base nos movimentos das pessoas. Esta técnica explora grafos de transição urbana para mapear os movimentos dos usuários entre locais da cidade. O grafo de transição urbana considerado é um grafo dirigido ponderado $G(V, E)$, em que um nó $v_i \in V$ é a categoria de um local específico (por exemplo, *food*) e uma aresta direcionada $(i, j) \in E$ marca uma transição entre duas categorias. Ou seja, uma aresta existe a partir do nó v_i para o nó v_j se pelo menos um usuário compartilhou um dado em um local categorizado por v_j logo após compartilhar um dado em um local categorizado por v_i . O peso $w(i, j)$ de uma aresta é o número total de transições que ocorreram a partir de v_i para v_j . Somente dados consecutivos compartilhados pelo mesmo usuário dentro de 24 horas, com início às 5:00, são considerados no cálculo de uma transição.

A *City Image* é uma técnica promissora que permite uma melhor compreensão da dinâmica de cidades, ajudando na visualização das rotinas comuns de seus cidadãos. Cada célula da *City Image* representa o quão favorável é uma transição de uma determinada categoria em um determinado lugar (eixo vertical) para outra categoria (eixo horizontal), valores que são calculados utilizando um modelo aleatório/nulo [Silva et al. 2014d]. As cores vermelhas representam rejeição, as cores azuis representam favorabilidade e a cor branca representa indiferença. Nós exemplificamos a técnica *City Image* para duas cidades²²: São Paulo

²²Utilizando dados do *dataset* Foursquare1.

(Figures 2.19a and 2.19b); e Kuwait (Figures 2.19c e 2.19d). Para ambos os casos, consideramos dias de semana durante o dia, que é o período típico de rotinas, e fim de semana durante a noite, que é um período representativo de atividades de lazer (fora da rotina).

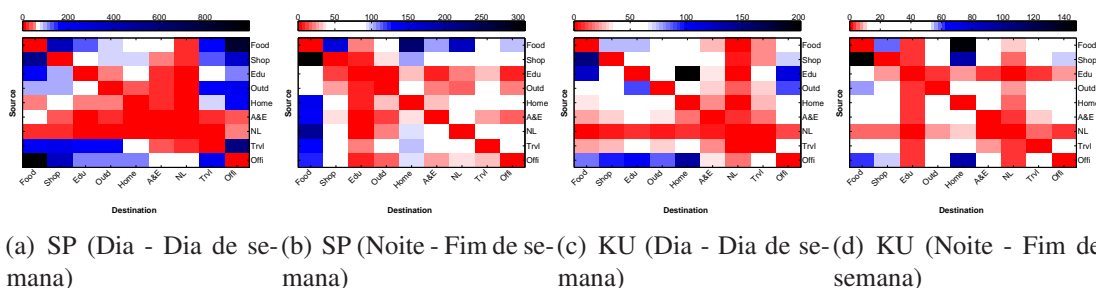


Figura 2.19. Imagens produzidas com a técnica *City Image* para São Paulo (SP) e Kuwait (KU) em diferentes períodos. Abreviaturas das categorias de locais (Nomes usados pelo Foursquare): *Arts & Entertainment (A&E)*; *College & Education (Edu)*; *Great Outdoors (Outd)*; *Nightlife Spot (NL)*; *Shop & Service (Shop)*; and *Travel Spot (Trvl)* [Silva et al. 2014a].

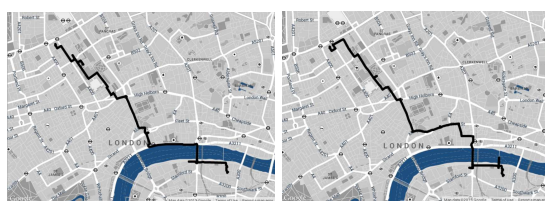
Primeiramente, observe que transições para *office* (locais de trabalho) são mais prováveis de acontecer nos dias de semana e durante o dia para ambas as cidades, como esperado. No entanto, note que as imagens da cidade de São Paulo e Kuwait também têm diferenças significativas que refletem diversidades culturais entre ambas as cidades. Note, por exemplo, que a imagem que representa transições nas noites do fim de semana (Figure 2.19d) mostra a falta de transições favoráveis para a categoria *nightlife* no Kuwait. Este não é o caso de São Paulo (Figure 2.19b), em que a transição *food* → *nightlife* é altamente favorável de acontecer. Isso sugere que em São Paulo as pessoas gostam de frequentar locais relacionados com comida (*food*) antes de ir para casas noturnas (*nightlife*). No Kuwait, em vez disso, as pessoas são provavelmente mais favoráveis a realizarem as transições *shop* → *food* e *food* → *home* nas noites do fim de semana.

Técnicas para facilitar a interpretação das rotinas de habitantes de uma cidade, tais como as mencionadas aqui, são ferramentas valiosas para ajudar os urbanistas a entender melhor a dinâmica de cidades e, conseqüentemente, tomar decisões mais eficazes em relação à problemas das cidades, por exemplo.

2.6.2. Mobilidade Urbana

Nesta seção apresentamos trabalhos que se concentram em estudar padrões de mobilidade urbana dos usuários com dados de RSPs. Esses dados incluem informações espaço-temporais, por exemplo, *check-ins* e fotos com coordenadas geográficas. O estudo da mobilidade é útil para muitas finalidades. Com dados de RSPs é possível entender, por exemplo, como os usuários alocam tempo para diferentes atividades, sendo, portanto, uma questão fundamental nas ciências sociais. Além disso, é possível projetar novas ferramentas para ajudar os engenheiros de tráfego a entender o fluxo de pessoas na cidade.

A modelagem dos padrões de mobilidade vem atraindo a atenção de pesquisadores em diferentes áreas, como física e computação [Brockmann et al. 2006, Zheng et al. 2009, Gonzalez et al. 2008]. É importante ressaltar que os dados derivados das RSPs são diferentes de dados provenientes de *traces* de GPS ou de dados tradicionais do uso do telefone



(a) Caminho mais curto (b) Caminho mais bonito

Figura 2.20. Mapas mostrando diferentes caminhos entre os mesmos locais [Quercia et al. 2014].

celular, como ligações telefônicas, e apresentam características especiais e variados contextos. Por exemplo, os *check-ins* em serviços de compartilhamento de localização ou fotos em um serviço de compartilhamento de fotos trazem informações extras sobre um lugar particular. Por exemplo, um *check-in* está associado com um tipo de local, e.g. bar, e uma foto pode trazer informações sobre a situação atual dentro deste local. Com isso, nosso foco aqui são estudos que analisam dados de RSPs.

Em [Quercia et al. 2014] os autores propuseram uma metodologia para recomendação de rotas que leva em consideração não somente o menor caminho, mas também características emocionais, por exemplo, beleza. Nem sempre o menor caminho é o que gostaríamos de percorrer. Um turista, por exemplo, pode optar por um caminho mais bonito, mesmo que a distância seja um pouco maior. A Figura 2.20 mostra dois caminhos entre os mesmos locais na cidade de Londres, em que um é o mais curto (Figura 2.20a) e o outro o mais bonito (Figura 2.20b).

Para quantificar o quão localidades urbanas são agradáveis, os autores usaram dados de uma plataforma de *crowd-sourcing* que mostra duas cenas de ruas em Londres (considerando centenas), e um usuário vota em qual acha mais bonita, tranquila e feliz. Em seguida, os autores traduzem os votos em medidas quantitativas de percepção de localização. Depois disso, os autores criam um grafo considerando essas localizações. Para isso, os autores dividiram a área da cidade em células de 200x200 metros. Cada célula é um nó no grafo e cada nó possui arestas com nós que representam células vizinhas. Esse grafo permite a descoberta de caminhos agradáveis.

Nguyen and Szymanski [Nguyen and Szymanski 2012] usaram dados do Gowalla para criar e validar modelos de mobilidade e relações humanas. Nesse trabalho, os autores propuseram um modelo de mobilidade baseado em amizade (FMM), que leva em conta os laços sociais, a fim de fornecer um modelo mais preciso da mobilidade humana. Com esse modelo, os autores foram capazes de estudar a frequência com que amigos viajam juntos. Ele pode melhorar a precisão de um número variado de aplicações, tais como engenharia de tráfego em redes de comunicação, sistemas de transporte e planejamento urbano.

O modelo de mobilidade proposto utiliza um modelo de Markov, onde os estados representam locais de *check-ins* e as ligações representam a probabilidade de ir de um lugar para outro. Por exemplo, a probabilidade de ir do trabalho para bar é definida como a razão entre o número de vezes que um determinado usuário executa um *check-in* em um bar logo após realizar um *check-in* no trabalho, e o número de vezes que o usuário realiza um *check-in* no trabalho.

Em [Zheng et al. 2012] os autores estudaram a mobilidade e padrões de viagem de turistas a partir de fotos compartilhadas no Flickr. A fim de extrair os padrões de viagem, os autores focaram as análises no movimento de turistas de acordo com as regiões atrativas e características topológicas de rotas de viagem feitas por diferentes turistas. Para isso, primeiro é construído um banco de dados de caminhos turísticos com base no conceito de entropia de mobilidade (considerando entropia de Shannon [Shannon 1948]), usada para discriminar o movimento turístico do não turístico.

Em seguida, os autores propõem um método para descobrir regiões atrativas em uma cidade, usando para isso o algoritmo de agrupamento DBSCAN [Ester et al. 1996]. Para estudar o movimento turístico, os autores consideraram um modelo de Markov criado a partir da sequência de visitas nas regiões atrativas. Com isso, os autores podem estimar as estatísticas de visitantes que viajam de uma região para outra. Para estudar as características topológicas de rotas de turismo, os autores realizam um agrupamento de rotas de viagem, aplicando uma versão modificada da maior subsequência comum (*longest common subsequence*), como uma métrica de similaridade para minimizar o ruído.

Esses esforços ilustram o crescente interesse e o potencial de utilização de dados compartilhados em RSPs para estudar padrões de mobilidade de humanos em larga escala.

2.6.3. Padrões Sociais, Econômicos e Culturais

Os dados de RSPs também podem ser usados para estudar aspectos sociais, econômicos e culturais dos habitantes de cidades. Por exemplo, pode-se argumentar que uma pequena quantidade de dados compartilhados em uma área da cidade pode indicar uma falta de acesso à tecnologia por parte da população local, pois o uso de serviços de compartilhamento de localização muitas vezes dependem de *smartphones* e planos de dados 3G ou 4G que, geralmente, são caros. Nessa direção, em [Silva et al. 2013a] nós mostramos que a análise de dados de RSPs permitem a visualização de fatos interessantes relacionados com questões socioeconômicas de uma cidade. Por exemplo, dados de uma RSP para compartilhamento de localização para a cidade do Rio de Janeiro são escassos em áreas pobres, incluindo as que são localizadas muito perto de áreas ricas. Essa informação pode ser útil para gerar melhores políticas públicas nessas áreas. Note que a mesma informação pode ser obtida utilizando métodos tradicionais, tais como questionários, mas esse processo é muito mais lento e caro.

Com o intuito de melhor entender padrões sociais a partir da análise de dados de RSPs, em [Quercia et al. 2012] os autores estudaram como comunidades virtuais, observadas nos sistemas analisados, se assemelham às comunidades da vida real. Os autores testaram se teorias sociológicas estabelecidas de redes sociais da vida real são válidas nessas comunidades virtuais. Eles descobriram, por exemplo, que os influentes (*social brokers*) no Twitter são líderes de opinião que se arriscam compartilhando *tweets* sobre diferentes temas. Eles também descobriram que a maioria dos usuários têm redes geograficamente locais, e que os influentes expressam não apenas emoções positivas, mas também negativas.

Para realizar este trabalho, os autores aplicaram métricas de rede que a literatura afirma que podem estar relacionadas com relações sociais, como a reciprocidade e restrição da rede [Quercia et al. 2012]. A reciprocidade r é a proporção de arestas em uma rede que são bidirecionais $L^{<->}$ em relação ao número total de arestas L : $r = \frac{L^{<->}}{L}$. Considerando

uma rede social focada em um vértice (“ego”) e vértices e arestas a quem o ego está diretamente conectado, valores baixos de reciprocidade poderiam indicar, por exemplo, uma rede social de uma celebridade. A restrição da rede mede as oportunidades de se tornar influente (*brokerage opportunities*). Um alto valor de restrição da rede significa menos oportunidades. Os autores usaram a formulação de Burt [Burt 1992] nesse caso específico. Mais detalhes sobre essas e outras métricas de rede que podem ser usadas para análises de relações sociais podem ser encontradas no livro: [Easley and Kleinberg 2010].

Numa direção similar, em [Joseph et al. 2012] os autores analisaram um conjunto de dados do Foursquare para identificar grupos de pessoas e os lugares que elas visitam. O modelo utilizado foi capaz de identificar grupos de pessoas que representam grupos espacialmente próximos e pessoas que parecem ter interesses semelhantes.

A abordagem utilizada se baseia na ideia de modelos probabilísticos para tópicos. Para isso, eles usaram o modelo de alocação latente de Dirichlet (LDA - *Latent Dirichlet Allocation*) [Blei et al. 2003], que é, geralmente, utilizado para estudar documentos. Na instanciação do modelo, cada *check-in* de um usuário é encarado como uma palavra de um “documento” que representa um usuário, isso de forma análoga a documentos de texto, onde um documento pode ter muitas palavras.

Além disso, ao estudar o comportamento social de áreas específicas, uma das primeiras perguntas que surgem é: o quão diferente uma cultura é de outra? Sabemos que os hábitos alimentares e de bebidas são capazes de descrever fortes diferenças culturais. Com base nisso, em [Silva et al. 2014c] propomos uma nova metodologia para a identificação de fronteiras culturais e semelhanças entre sociedades, considerando hábitos alimentares e de bebida. Para isso, foram usados *check-ins* do Foursquare para representar as preferências do usuário em relação ao que se come e bebe localmente, por exemplo, em uma determinada cidade.

Essa análise surpreendentemente diz muito sobre as diferenças e semelhanças entre as culturas. Para isso estudamos a correlação entre os *check-ins* dados em diferentes tipos de restaurantes para várias cidades ao redor do mundo. Observamos que as cidades de um mesmo país, onde os habitantes normalmente possuem cultura e hábitos alimentares semelhantes, têm as correlações mais fortes com relação às preferências de restaurante. Além de preferências para as categorias de alimentos, também podemos ver diferenças nos horários em que as pessoas vão a restaurantes e compartilham dados, como foi apresentado na Seção 2.5.3. Essas análises permitiram a proposição de uma metodologia para a identificação de culturas semelhantes, que pode ser aplicada em regiões de tamanhos variados, como países, cidades ou até mesmo bairros [Silva et al. 2014c]. Nessa metodologia é utilizado um algoritmo de agrupamento baseado em particionamento (*k – means* [Hartigan and Wong 1979]), bem como a técnica de análise de componentes principais [Jolliffe 2002]. Os resultados para países e cidades são ilustrados nas Figuras 2.21a e 2.21b, mostrando como culturas semelhantes são bem separadas. Nessas figuras foram usados os dois principais componentes apenas para mostrar os resultados, no entanto a obtenção do resultado considerou todos os componentes.

As diferenças culturais utilizando dados de RSPs também foram estudadas em [Hochman and Schwartz 2012], que investigaram as preferências de cores em fotos compartilhadas no Instagram. Os autores encontraram diferenças consideráveis entre imagens

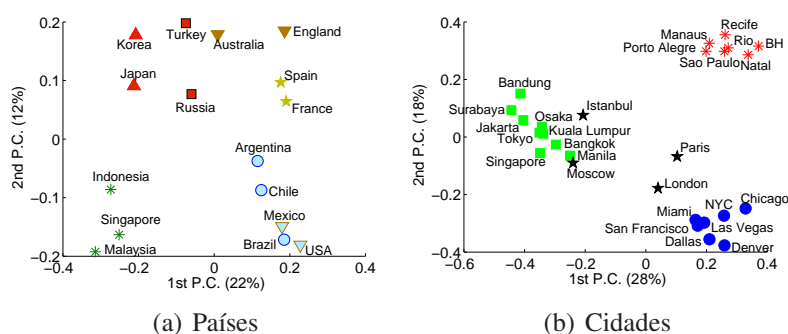


Figura 2.21. Grupos encontrados utilizando a metodologia de separação de culturas. Cada símbolo reflete um grupo [Silva et al. 2014c].



Figura 2.22. Localização das lojas analisadas [Karamshuk et al. 2013].

de países com culturas distintas. Na mesma direção, em [Poblete et al. 2011] os autores investigaram como o comportamento de divulgação de conteúdo no Twitter varia entre alguns países, bem como as possíveis explicações para essas diferenças. A investigação das distinções culturais entre diferentes cidades e países é valiosa em muitas áreas e pode auxiliar várias aplicações. Por exemplo, como cultura é um aspecto importante por razões econômicas, a identificação de semelhanças entre os lugares que estão geograficamente separados pode ser necessária para empresas que possuem negócios em um país e querem avaliar a compatibilidade de preferências entre diferentes mercados.

Relacionado com o aspecto econômico das cidades, em [Karamshuk et al. 2013] os autores estudaram o problema da alocação ótima de lojas de varejo na cidade. Eles usaram dados do Foursquare para compreender como a popularidade de três redes de lojas de varejo em Nova Iorque (veja a Figura 2.22) é definida em termos de número de *check-ins*.

Foram avaliadas um conjunto diversificado de características (*features*), modelando informações espaciais e semânticas sobre os locais e padrões de movimentos dos usuários na área ao redor do local analisado. Os autores observaram que a presença de locais que atraem muitos usuários naturalmente, como estação de trem ou aeroporto, bem como lojas de varejo do mesmo tipo das analisadas, que definem a concorrência comercial local de um área, são os indicadores mais fortes de popularidade.

2.6.4. Detecção de Eventos e Interesses

A identificação de eventos e pontos de interesse através de dados de RSPs é beneficiada pela natureza de tempo (quase) real das RSPs. Eventos podem ser naturais, tais como terremotos, ou não naturais, tais como a identificação/previsão de mudanças no mercado de ações. Por sua vez, um ponto de interesse é uma localização específica que alguém pode achar útil ou interessante, como um restaurante ou um estádio de futebol.

Em relação à detecção de eventos, em [Gomide et al. 2011] os autores analisaram como a epidemia de Dengue é refletida no Twitter e em que medida essa informação pode ser usada na vigilância dessa doença. Os autores mostraram que o Twitter pode ser usado para prever, espacial e temporalmente, epidemias de dengue. Eles analisam como dados do Twitter refletem epidemias com base em quatro dimensões: volume, localização, tempo e percepção do público. Especificamente, os autores estudam como os usuários se referem à

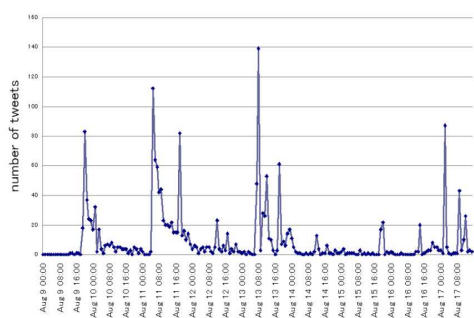


Figura 2.23. Número de *tweets* relacionados com terremotos [Sakaki et al. 2010b].

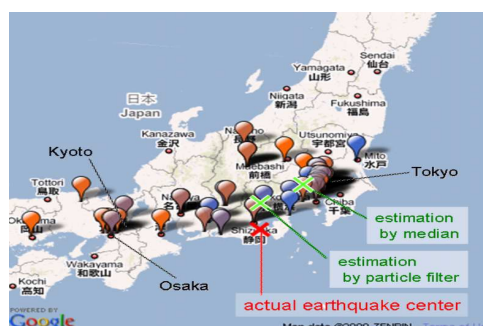


Figura 2.24. Estimativa da localização do terremoto [Sakaki et al. 2010b].

dengue no Twitter com análise de sentimentos [Gonçalves et al. 2013] e usam o resultado para focar em apenas *tweets* que, de alguma forma, expressam experiência pessoal sobre a dengue. Em seguida, os autores construíram um modelo baseado em regressão linear [Yan 2009] para a previsão do número de casos de dengue. O resultado dessa pesquisa é aplicado no projeto chamado Observatório da Dengue²³.

Em [Sakaki et al. 2010b] os autores estudaram a interação em tempo real de acontecimentos no Twitter, por exemplo, terremotos, e propuseram um algoritmo para monitorar mensagens no Twitter para detectar a ocorrência de eventos. Para demonstrar a eficácia de seu método, os autores construíram um sistema de aviso de terremoto no Japão, que foi capaz de detectar 96% dos terremotos relatados pela Agência Meteorológica do Japão (JMA) com escala de intensidade sísmica de 3 ou mais. A notificação foi capaz de ser entregue mais rápida do que os avisos que são transmitidos pela JMA. A abordagem utilizada é um classificador de *tweets* com base em características, tais como: palavras-chave em um *tweet*; o número de palavras; e o seu contexto. Depois disso, os autores produziram um modelo espaço-temporal probabilístico para o evento alvo que pode encontrar o centro e a trajetória do local do evento.

Em os autores [Bollen et al. 2011] estudaram se os estados coletivos de humor derivados de mensagens do Twitter são correlacionados com o valor da bolsa Dow Jones ao longo do tempo. Seus resultados indicam que é possível obter uma boa precisão na previsão das mudanças diárias de alta e queda dos valores de fechamento dessa bolsa de valores. Isso é possível escolhendo dimensões de humor específicos, mas não todos os que foram considerados.

Além de eventos que tendem a acontecer esporadicamente, toda cidade possui um conjunto de áreas que desperta um maior interesse dos residentes ou visitantes, as aqui denominadas *pontos de interesse* (PDI). Dentre os PDIs mais visitados, podemos mencionar os pontos turísticos da cidade. No entanto, nem todos os PDIs de uma cidade são pontos turísticos. Por exemplo, uma área de bares pode ser bastante popular entre os residentes da cidade, mas sem atrativos para turistas. Além disso, PDIs são dinâmicos, ou seja, áreas que são populares hoje podem não ser mais amanhã. Assim, uma aplicação que emerge naturalmente a partir da análise de dados de algumas RSPs, por exemplo para compartilhamento

²³<http://www.observatorio.inweb.org.br/dengue>.

de fotos ou localização, é a identificação de PDIs. Isso é possível porque cada foto ou *check-in* representa, implicitamente, um interesse de um indivíduo em um determinado instante. Com isso, quando muitas fotos de um determinado local são compartilhadas dentro de um certo intervalo de tempo, esse local pode ser um PDI.

Uma vantagem de usar RSPs para identificar pontos de interesse na cidade é que podemos obter resultados robustos a mudanças dinâmicas. Ou seja, pelo fato das RSPs fornecerem dados dinâmicos, elas podem capturar automaticamente as alterações nos interesses das pessoas ao longo do tempo, ajudando a identificar rapidamente as áreas que por ventura se tornem um PDI (por exemplo, devido à abertura de um novo restaurante) ou que deixem de ser populares.

A identificação de pontos de interesse em uma cidade foi investigada em [Crandall et al. 2009], onde os autores mostraram como inferir a localização de uma foto sem usar os dados geoespaciais. Na mesma direção, em [Kisilevich et al. 2010] os autores usaram fotos geolocalizadas para analisar e comparar eventos temporais que aconteceram em uma cidade, e também para classificar locais turísticos.

Além disso, em [Silva et al. 2013b] nós também apresentamos uma técnica para identificar PDIs e, a partir deles, identificar pontos turísticos. A técnica considera que cada par i de coordenadas (longitude, latitude) $(x, y)_i$ está associada a um ponto p_i , que representa um dado compartilhado, e.g. uma foto. Nós começamos calculando distância geográfica entre cada par de pontos (p_i, p_j) (usando a fórmula de Haversine [Sinnott 1984]) e agrupamos todos os pontos p_i que estão próximos uns dos outros (utilizando um método de agrupamento hierárquico aglomerativo, usando como critério de ligação: “complete-linkage” [Sørensen 1948, Kaufman and Rousseeuw 2009]). Para capturar os PDIs, usamos um modelo nulo para excluir grupos que possam ter sido gerados por situações aleatórias (ou seja, movimentos de pessoas aleatórias), e, portanto, não refletem a dinâmica da cidade. Para identificar os grupos, analisamos o número de compartilhamento de dados em cada um deles e usamos métodos estatísticos simples. Em seguida, separamos os pontos turísticos dos PDIs assumindo que turistas possuem rotas conhecidas na cidade (mais detalhes em [Silva et al. 2013b]).

Quando aplicada para a cidade de Belo Horizonte considerando dados do Foursquare e Instagram, essa técnica foi capaz de encontrar a maioria dos seus PDIs e pontos turísticos. Os resultados também mostram que diferentes RSPs podem fornecer dados complementares, pois nenhuma RSP encontrou todos os pontos turísticos. Tais diferenças podem refletir mudanças na cidade durante o intervalo de tempo em que um *dataset* específico foi coletado. Por exemplo, durante a coleta do *dataset* Instagram1, Belo Horizonte não estava recebendo jogos de futebol. Isso explica por que o estádio de futebol não foi identificado como um PDI utilizando esse *dataset*. Por outro lado, a análise de um *dataset* do mesmo sistema coletado mais recentemente (Instagram2), identificou corretamente o estádio como um ponto turístico importante da cidade. Isso ilustra como os dados de RSPs podem capturar automaticamente alterações da dinâmica da cidade, sendo úteis para detectar locais incomuns e populares, bem como descobrir possivelmente padrões inesperados.

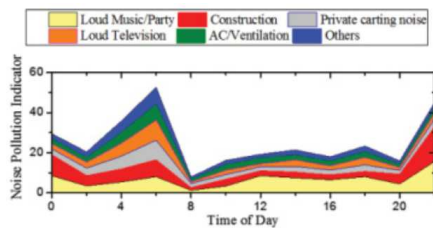
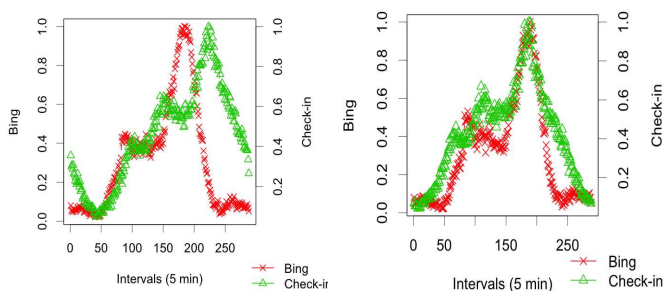


Figura 2.25. Ruídos de diferente categorias na Times Square [Zheng et al. 2014b].



(a) Intervalo de 5 minutos (b) Intervalo de 5 minutos com shift

Figura 2.26. Frequência de trânsito intenso em diferentes intervalos de tempo em um dia típico (segunda–sexta) [Tostes et al. 2014].

2.6.5. Problemas das Cidades

A coleta de dados sobre problemas que as cidades enfrentam pode ser facilitada com o uso de RSPs como o Colab.re. Essa RSP permite aos usuários criar, visualizar e compartilhar problemas de diversas naturezas sobre a cidade. Além desse exemplo, existem outras RSPs para o monitoramento de questões específicas do meio ambiente urbano, como o nível de ruído. Por exemplo, NoiseTube [Maisonneuve et al. 2009], como apresentamos anteriormente.

Com base no NoiseTube, D’Hondt e Stevens [D’Hondt et al. 2013] conduziram um experimento para mapear os níveis de ruído em Antwerp, Bélgica. Um dos objetivos era avaliar a qualidade dos mapas de ruído obtidos por sensoriamento participativo, em comparação com os mapas de ruído oficiais baseados em simulação. Para isso, foram realizados vários experimentos de calibração, investigando diversos aspectos dos padrões de ruído. Os autores foram capazes de construir mapas de ruído com uma margem de erro de 5dB, o que é comparado com mapas de ruído oficiais baseados em simulação.

Além dessas iniciativas, desde 2001, Nova Iorque disponibilizou uma plataforma chamada 311 para permitir que as pessoas reclamem de problemas da cidade usando um aplicativo móvel. Cada reclamação está associada a um local, data e hora, e, em alguns casos, informação detalhada da reclamação, como música alta ou barulho de construção (para os problemas de ruído). Usando os dados do serviço 311, em [Zheng et al. 2014b] os autores inferem a situação de ruído (consistindo de um indicador de poluição sonora), em diferentes momentos do dia para cada região de Nova Iorque. De acordo com o indicador de poluição sonora, é possível verificar a composição de ruído de um determinado local mudando ao longo do tempo (por exemplo, Time Square), como mostrado na Figura 2.25.

Os autores modelaram a situação de ruído de Nova Iorque com um tensor tridimensional, em que as três dimensões representam regiões, categorias de ruído e intervalos de tempo, o que permite recuperar a situação do ruído em toda a cidade. A informação de ruído não só pode facilitar a qualidade de vida de um indivíduo (por exemplo, ajudar a encontrar um lugar tranquilo para se estabelecer), mas também auxiliar os responsáveis governamentais no combate à poluição sonora.

Focados nos problemas de trânsito da cidade, em [Tostes et al. 2014] os autores estu-

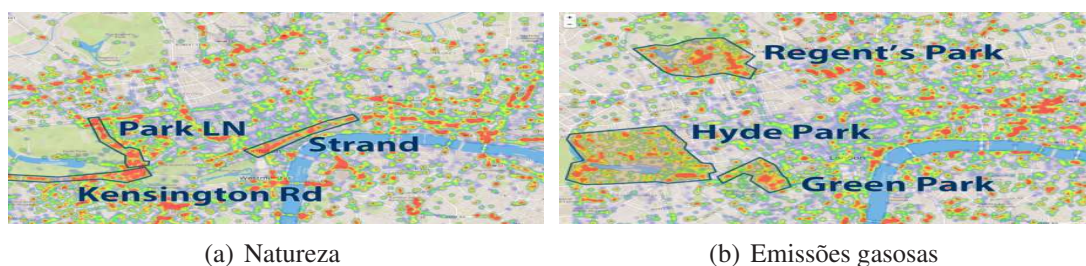


Figura 2.27. Mapas de cheiros em Londres [Quercia et al. 2015].

daram a seguinte pergunta: é possível utilizar dados de RSPs como uma característica para predição de trânsito intenso? Os autores observaram que os dados de RSP, especialmente os provenientes do Instagram e Foursquare, são surpreendentemente bastante correlacionados com trânsito intenso e que podem ser utilizados para desenvolver modelos de previsão de congestionamento mais eficientes. Para tratar a questão, os autores desenvolveram uma *Web crawler* para a coleta de condições de tráfego do Google Maps e do Bing Maps. De posse desses dados, bem como com dados de RSPs, os autores estudaram a cidade de Nova Iorque, mais especificamente a área de Manhattan. Conforme o estudo demonstra, os dados de RSP são bastante correlacionados com trânsito intenso, como mostrados na Figura 2.26. Ao comparar as Figuras 2.26a e 2.26b, podemos ver que a distribuição de trânsito intenso e a distribuição de *check-ins* durante os dias de semana são bastante semelhantes, no entanto as curvas são deslocadas no eixo x por um valor que pode ser calculado de acordo com uma equação proposta no estudo. Essa descoberta é surpreendente e sugere que dados de RSPs podem refletir as condições reais de tráfego. Em [Tostes et al. 2013] os autores propuseram também um modelo de previsão de tráfego utilizando uma regressão logística [Freedman 2009].

MacKerron and Mourato [MacKerron and Mourato 2013] estudaram como o ambiente local afeta a felicidade das pessoas. Para realizar esse estudo, os autores utilizaram uma aplicação para *smartphones* que tinha a finalidade de permitir aos usuários informarem o seu grau de humor, localização GPS e o nível de ruído do ambiente. Analisando mais de 3 milhões de registros de 45.000 pessoas no Reino Unido, os autores constataram que, em média, os participantes são significativamente mais felizes em ambientes externos, em contato com a natureza, do que em ambientes urbanos.

Em [Quercia et al. 2015] os autores exploraram a possibilidade de utilizar dados compartilhados em RSPs para mapear os cheiros percebidos em diversas regiões da cidade. Os resultados encontrados são promissores e mostram que essa pode ser uma nova forma para classificar áreas de acordo com o seu cheiro mais característico. Para realizar esse estudo, os autores consideraram dados do Instagram, Flickr e Twitter. Eles combinaram *tags* e *tweets* com as palavras de um “dicionário de cheiro” já existente. Em seguida, analisaram essas ocorrências na cidade. A Figura 2.27 mostra, utilizando um *heatmap*, a ocorrência de dados referentes a cheiros de natureza e emissão gasosa em Londres, Figuras 2.27a e 2.27b, respectivamente. É possível notar que o cheiro de natureza é fortemente observado em parques, e o cheiro de emissão gasosa em ruas com tráfego intenso.

2.6.6. Discussão

As RSPs oferecem informações atualizadas sobre locais, bem como opiniões e preferências de seus usuários. Além disso, elas têm o potencial de tratar as questões acima mencionadas em tempo (quase) real, atingindo um elevado número de regiões do globo. Nesta seção mostramos vários estudos que servem como exemplos de como trabalhar com dados de RSPs. As informações obtidas por esses estudos podem ser úteis para o desenvolvimento de serviços e aplicações mais inteligentes. Por exemplo, entender o padrão de comportamento em determinados locais na cidade, bem como a identificação de comportamentos fora do padrão esperado pode ser muito útil para o planejamento de carga de uma rede celular urbana. Estudos que visam oferecer soluções para desafogar a transmissão de dados móveis (*mobile data offloading*) podem ter grandes benefícios ao utilizar essas informações como uma ferramenta para diminuir surpresas em demandas atuais, bem como novas demandas que podem surgir, já que a cidade está em constante mudanças. Várias outras oportunidades, bem como os desafios associados a elas, são discutidas na Seção 2.8.

2.7. Compilação de Técnicas e Ferramentas

Nesta seção discutimos as principais técnicas utilizadas nos trabalhos exemplificados nas seções anteriores, bem como algumas das tecnologias e ferramentas comumente utilizadas para a análise de dados urbanos. O objetivo não é fazer uma revisão completa da literatura. No entanto, acreditamos que os apontadores mostrados aqui podem ser bastante úteis no desenvolvimento de novas aplicações e serviços na área de computação urbana.

2.7.1. Algoritmos de Agrupamento

A realização de agrupamentos pode ser muito útil para a análise de dados urbanos, permitindo oferecer novos serviços por meio de agrupamento de características espaciais. A seguir, apresentaremos os principais algoritmos de agrupamento mencionados anteriormente.

Agrupamento baseado em densidade. Um exemplo de algoritmo popular dessa classe é o DBSCAN (*Density-based spatial clustering of applications with noise*) [Ester et al. 1996]. Algumas definições: a densidade é referente ao número de pontos dentro de um raio específico (ϵ); Um *core point* tem um número mínimo de pontos especificados pelo usuário (*minPts*) dentro do raio (ϵ); Um *border point* fica localizado na vizinhança de um *core point*; Um *noise point* é qualquer ponto que não se classifica como *core point* nem como *border point*.

A Figura 2.28 ilustra as definições mencionadas do DBSCAN, onde *minPts* = 3. O ponto *A* é um *core point*, pois pelo menos três pontos são vizinhos dele em um raio ϵ . Existem vários outros *core points* e como todos são todos acessíveis a partir de um outro *core point*, eles formam um único *cluster*. Os pontos *B* e *C* não são *core points*, mas são chamados de *border points*, porém são acessíveis a partir de *A* (que é um *core point*) e, portanto, também pertencem ao *cluster*. O ponto *N* é um *noise point*, pois não é um *core point* nem um *border point*.

Ideia do algoritmo: selecionar arbitrariamente um ponto *p*. Identificar todos os pontos densamente conectados a *p* com relação aos parâmetros ϵ e *minPts*. Se *p* é um *core point*, um *cluster* é formado. Se *p* é um *border point* e não há pontos densamente conecta-

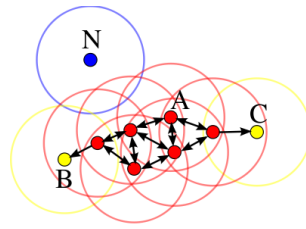


Figura 2.28. Diagrama ilustrativo do DBSCAN²⁵.

dos a p o DBSCAN visita o próximo ponto do conjunto de dados. O processo continua até que todos os pontos tenham sido avaliados.

Agrupamento baseado em particionamento. Algoritmos de agrupamento desta classe constroem várias partições e as avalia usando algum critério. As partições são criadas a partir da segmentação de um conjunto de dados em um conjunto de k clusters. O objetivo é encontrar uma partição de k clusters que otimiza o critério de particionamento escolhido.

O k – means [Hartigan and Wong 1979] é um algoritmo bastante popular para agrupamento baseado em particionamento. Esse algoritmo divide um conjunto de N amostras em k clusters disjuntos. Em linhas gerais, o algoritmo possui três passos. O primeiro é escolher os centroides iniciais, por exemplo, selecionando k amostras aleatórias do conjunto N . Após a inicialização, o k – means considera um laço de repetição contendo dois outros passos: atribuir cada amostra a um centroide mais próximo; e criar novos centroides calculando o valor médio de todas as amostras designadas a cada centroide anterior. Essas duas etapas se repetem até que o centroide não se mova significativamente.

Agrupamento hierárquico. O agrupamento hierárquico é um método que visa criar uma hierarquia de clusters. Estratégias de agrupamento hierárquico geralmente caem em dois tipos: (i) Aglomerativos: esta é uma abordagem “bottom up”, ou seja, cada observação começa em seu próprio cluster, e a cada passo combina-se clusters com alguma característica comum; e (ii) Divisivos: esta é uma abordagem “top down”, ou seja, todas as observações começam em um cluster, e divisões são realizadas de forma recursiva quando se desce na hierarquia, até que cada cluster tenha somente registros semelhantes.

A fim de decidir quais clusters devem ser combinados (por métodos aglomerativos), ou quando um cluster deve ser dividido (por métodos divisivos), é necessária uma medida de dissimilaridade entre os conjuntos de observações. Na maior parte dos métodos de agrupamento hierárquico, isto é alcançado pelo uso de uma medida apropriada de distância entre pares de observação e um critério de ligação (*linkage criteria*). Esse critério de ligação especifica a dissimilaridade dos conjuntos em função das distâncias entre pares de observações nos conjuntos. Um exemplo de critério de ligação é o *complete-linkage* [Sørensen 1948, Kaufman and Rousseeuw 2009]: $D_{AB} = \max \{ d(a, b) : a \in A, b \in B \}$, onde d é a métrica de distância escolhida, por exemplo, distância euclidiana, e A e B são conjuntos de observações. Por exemplo, valores de D_{AB} abaixo de um limite predefinido resultariam na aglomeração de A e B .

Agrupamento espectral. Em linhas gerais, o agrupamento espectral usa a similaridade

²⁵<https://en.wikipedia.org/wiki/File:DBSCAN-Illustration.svg>.

dade entre os dados para definir os *clusters*. Para isso dois objetos matemáticos são usados: grafos de similaridade e grafos laplacianos (*graph Laplacians*). No grafo de similaridade cada vértice v_i do grafo representa um dado x_i do conjunto de dados, e dois vértices são conectados se a similaridade s_{ij} entre os dados correspondentes é positiva ou maior do que um certo limiar, com s_{ij} sendo o peso das arestas. A ideia é achar partições no grafo tal que arestas de diferentes *clusters* tenha peso baixo, enquanto que arestas dentro do mesmo *cluster* tenham pesos elevados [Luxburg 2007].

Os grafos laplacianos são as principais ferramentas para o agrupamento espectral. Diferentes algoritmos espectrais existem, cada um utiliza um grafo laplaciano descrito por von Luxburg. A ideia principal por trás do algoritmo espectral é aumentar as propriedades de agrupamento dos dados em uma nova representação de maneira que os *clusters* sejam trivialmente detectados utilizando, por exemplo, algoritmos de como o $k - means$, como nos exemplos de [Luxburg 2007].

Outros métodos de agrupamento, bem como mais detalhes sobre as técnicas de agrupamento mencionadas aqui podem ser encontrados nos livros: [Zaki and Meira Jr 2014, Kaufman and Rousseeuw 2009].

2.7.2. Regressões

Em estatística, regressão é uma técnica que permite explorar e inferir a relação de uma variável dependente (variável de resposta) com variáveis independentes específicas (variáveis explanatórias). A regressão linear foi o primeiro tipo de regressão a ser estudado rigorosamente, bem como ser usado extensivamente em aplicações práticas, incluindo a predição de valores. A regressão linear é uma equação para se estimar a condicional (valor esperado) de uma variável y , dados os valores de algumas outras variáveis x . Com isso, em geral, a regressão linear trata da questão de se estimar um valor condicional não esperado [Yan 2009].

Para se estimar o valor esperado usa-se a equação: $Y_i = \alpha + \beta X_i + \varepsilon_i$, sendo que Y_i é a variável explicada; α é uma constante, que representa a interceptação da reta com o eixo vertical; β é outra constante, que representa o declive (coeficiente angular) da reta; X_i é uma variável explicativa (independente), que representa o fator explicativo na equação; ε_i é a variável que inclui todos os fatores residuais mais os possíveis erros de medição [Yan 2009].

Outra técnica utilizada nos trabalhos citados neste minicurso é a regressão logística [Freedman 2009]. A regressão logística é um tipo de regressão que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica a partir de uma série de variáveis explicativas contínuas e/ou binárias.

Em comparação com as técnicas conhecidas em regressão, em especial a regressão linear, a regressão logística distingue-se principalmente pelo fato da variável resposta ser categórica. Mais detalhes sobre outros modelos de regressão, bem como os que mencionamos aqui podem ser encontrados nos livros [Harrell 2013, Chatterjee and Hadi 2015, Freedman 2009].

2.7.3. Modelos Probabilísticos para Tópicos

Modelos probabilísticos para tópicos são usados, tipicamente, para entender e estruturar grandes coleções de documentos. O modelo de tópico mais comum é o modelo de alocação latente de Dirichlet (LDA – *Latent Dirichlet Allocation*), que utiliza a abordagem bayesiana para aprender a estrutura latente de temas que compreendem cada um dos documentos. Esse modelo vem sendo utilizado em muitos trabalhos de novas áreas, incluindo a computação urbana.

O LDA se baseia em um modelo generativo assumindo que cada um dos documentos em particular na coleção é uma mistura de temas. Num modelo LDA típico, um documento de texto é representado como um conjunto de palavras, onde cada palavra pertence a um ou mais tópicos ocultos. Assim, cada documento pode ser descrito ao considerar o quanto as palavras dentro dele se relacionam com os vários tópicos ocultos e, cada tópico oculto pode ser descrito pelos termos que são mais fortemente associados a ele. Por exemplo, um documento sobre a abertura de um novo restaurante italiano pode conter as palavras “restaurante” e “jantar”, associadas ao Tópico 1, e as palavras “pizza” e “spaghetti”, associadas ao Tópico 2. Nesse caso, o LDA nos daria informações sobre a forma como o documento é relacionado aos dois tópicos e, com isso, poderíamos entender sobre o que esses dois tópicos dizem respeito considerando as palavras que estão associadas a eles. Por exemplo, o Tópico 2 é provável que seja sobre “alimentos” ou “comida italiana”.

Uma característica bastante interessante do modelo LDA é o mínimo de intervenção humana requerida para sua aplicação. O modelo LDA é capaz de descobrir tópicos relacionados a documentos e estabelecer links entre documentos, mesmo quando não temos nenhuma informação anterior sobre estes tópicos. Além disso, os documentos não são obrigados a ser rotulados com tópicos ou palavras-chave na inicialização do modelo LDA.

2.7.4. Redução de Dimensionalidade

O objetivo dessa classe de técnicas é sumarizar os dados que contêm muitas variáveis por um conjunto menor de variáveis compostas derivadas a partir do conjunto original. Uma das motivações para o uso destas técnicas é a compressão dos dados, o que permite economizar espaço em disco e memória, bem como aumentar o desempenho de processos de aprendizado de máquina. Outra motivação é proporcionar uma maneira melhor de visualizar resultados que possuem um grande número de características (dimensões).

A técnica análise de componentes principais (*principal components analysis* - PCA [Jolliffe 2002]), é uma técnica estatística amplamente utilizada para a redução de dimensão não supervisionada. O objetivo é encontrar k vetores $u^{(1)}, u^{(2)}, \dots, u^{(k)} \in \mathbb{R}^n$ para projetar os pontos de dados de modo a minimizar o erro de projeção. Em outras palavras, dado um conjunto de pontos de dados em um espaço dimensional, o objetivo é projetá-los em um espaço de dimensão menor, preservando o máximo possível de informações. A Figura 2.29 ilustra uma projeção ortogonal dos dados para uma dimensão menor: 2-D para 1-D. As linhas pontilhadas entre os pontos originais (fundo branco) e os pontos projetados na linha sólida (fundo preto) representam os erros de projeção que devem ser minimizados. Podemos utilizar o PCA, por exemplo, para encontrar a melhor aproximação planar de dados três dimensões, ou a melhor aproximação de doze dimensões dos dados com 10^4 dimensões.

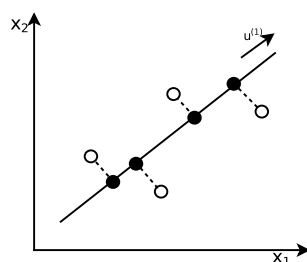


Figura 2.29. Exemplo de projeção ortogonal dos dados para uma dimensão menor: 2-D para 1-D.

Implementações para PCA podem ser encontradas em várias ferramentas. No R²⁶, as funções *princomp*²⁷ e *prcomp*²⁸ podem ser usadas para PCA. Em Matlab²⁹/Octave³⁰ a função *princomp*³¹ calcula os componentes principais.

2.7.5. Análise de Sentimento

A análise de sentimento (também conhecida como mineração de opinião) refere-se ao uso de processamento de linguagem natural, análise de texto e linguística computacional para identificar e extrair informações subjetivas de materiais textuais. De um modo geral, a análise de sentimento tem como objetivo determinar a atitude de um autor com relação a algum tema ou a polaridade contextual global de um documento. A atitude pode ser o julgamento ou avaliação do autor, o seu estado afetivo (ou seja, o estado emocional quando estava escrevendo), ou a sua comunicação emocional pretendida (ou seja, o efeito emocional que se deseja ter no leitor) [Reis et al. 2015, Gonçalves et al. 2013].

A tarefa básica na análise de sentimento é classificar a polaridade da opinião expressa pelo autor em positiva, negativa ou neutra. Existem vários métodos que permitem a análise de sentimentos. No entanto, o método SentiStrength [Thelwall et al. 2010] é o exemplo aqui ilustrado, já que é bastante utilizado, possui boas taxas de precisão e foi desenvolvido para textos curtos (tamanho comumente encontrado em redes de sensoriamento participativo) [Gonçalves et al. 2013].

O SentiStrength atribui um sentimento positivo de força 1 (nenhum sentimento positivo) a 5 (sentimento positivo muito forte) e um sentimento negativo de força -1 (nenhum sentimento negativo) a -5 (sentimento negativo muito forte) para cada texto. Uma parte principal do SentiStrength é uma lista de várias palavras associadas a uma determinada força de sentimento positivo ou negativo. Por exemplo, “bom” possui valor 3 e “medo” possui valor -4, assim a frase “Eu estava com medo, mas foi bom” pode resultar 3 na escala positiva e -4 na escala negativa. Além disso, existem regras especiais para lidar com negações, perguntas, palavras de reforço (por exemplo, muito), *emoticons*, e uma série de outros casos especiais. Se for necessário um único número global para a força sentimento então o número positivo pode ser adicionado ao número negativo para dar uma pontuação no intervalo de -4 a 4

²⁶<http://www.r-project.org>.

²⁷<http://stat.ethz.ch/R-manual/R-patched/library/stats/html/princomp.html>.

²⁸<http://stat.ethz.ch/R-manual/R-patched/library/stats/html/prcomp.html>.

²⁹<http://www.mathworks.com/products/matlab>.

³⁰<http://www.gnu.org/software/octave>.

³¹<http://octave.sourceforge.net/statistics/function/princomp.html>.

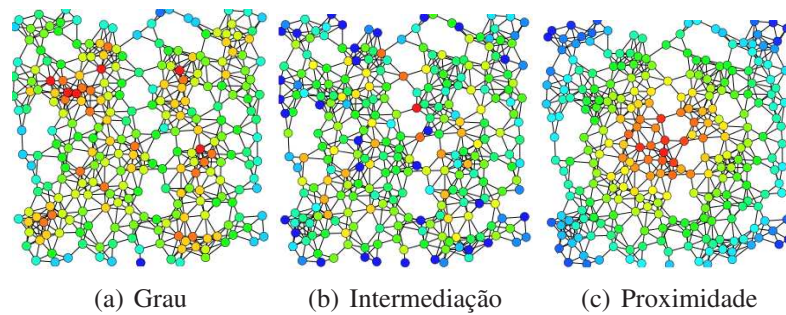


Figura 2.30. Resultados de centralidade considerando várias métricas para o mesmo grafo³³.

[Thelwall et al. 2010].

2.7.6. Teoria de Grafos/Redes

Estruturas que podem ser representadas por grafos/redes estão em toda parte e muitos problemas de interesse prático podem ser formulados como questões sobre certos grafos [Newman 2010]. Os “grafos de transição urbana” ($G(V, E)$) representam um exemplo de grafo bastante informativo sobre a dinâmica da cidade e do comportamento social urbano. Esse tipo de particular de grafo representa, por exemplo, um conjunto V de locais na cidade (denominados vértices) e um conjunto E de pares não ordenados de V que representam a movimentação dos usuários na cidade (as chamadas arestas).

A teoria de grafos é bastante rica, oferecendo várias métricas e conceitos, como métricas de centralidade. Em particular, essas métricas determinam a importância relativa de um vértice ou aresta no grafo, que no contexto de grafos de transição urbana representam o quanto um determinado local é importante. Algumas das medidas de centralidade que são amplamente utilizadas na análise de grafos: centralidade de grau (*degree centrality*), centralidade de intermediação (*betweenness centrality*) e centralidade de proximidade (*closeness centrality*) [Newman 2010].

A centralidade de grau é definida como o número de ligações incidentes sobre um vértice, ou seja, seu grau. Em um grafo de transição urbana, um vértice com grau elevado indica um local onde as pessoas podem chegar e sair com uma alta probabilidade. Assim, a centralidade de grau é uma boa medida para identificar lugares populares da cidade. Estes locais podem ser vistos como “*hubs*” da cidade.

A centralidade de intermediação mede a importância de um vértice dentro de um grafo (existe também a intermediação das arestas, que não é discutida aqui). A centralidade de intermediação quantifica o número de vezes que um vértice age como ponte ao longo do caminho mais curto entre dois outros vértices s e t . A intermediação de um vértice v pode ser calculada assim: $C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$, onde, σ_{st} é o número total de caminhos curtos desde o vértice s ao vértice t e $\sigma_{st}(v)$ é o número desses caminhos que passam por v . Essa métrica pode ser útil, por exemplo, para indicar os locais mais interessantes para atuarem como pontes para disseminar informações entre diferentes lugares ou regiões de lugares (conjunto de lugares) na cidade.

³³<https://upload.wikimedia.org/wikipedia/commons/thumb/1/14/Centrality.svg/600px-Centrality.svg.png>.

A centralidade de proximidade está relacionada com a proximidade de um vértice para todos os outros vértices na rede, ou seja, o número de arestas que separam um vértice a partir dos outros. No contexto de divulgação de informações, quanto maior a proximidade de um local, maior é a probabilidade de que a informação a ser disseminada a partir desse local alcance toda a rede na menor quantidade de tempo. Na perspectiva de um grafo de transição urbana, a centralidade de proximidade pode indicar locais estratégicos para instalar centros públicos para anúncio de informações à população. Uma ilustração do resultado de todas as métricas mencionadas aplicadas a um mesmo grafo é mostrada na Figura 2.30.

Existem várias outras métricas e conceitos de teoria de grafos além dos que foram mencionadas no texto do minicurso. Uma revisão detalhada das principais métricas e conceitos pode ser encontrada em [Newman 2010, Newman 2003, Easley and Kleinberg 2010]

2.7.7. Validações

Dados de RSPs podem possuir várias limitações. Em primeiro lugar, eles podem refletir o comportamento dos cidadãos que utilizam as RSPs: usuários que tendem a ter menos de 50 anos, donos de *smartphones* e moradores urbanos [Brenner and Smith 2013, Duggan and Smith 2014]. Além disso, os usuários podem não compartilhar dados em todos os seus destinos, por exemplo, hospitais e hotéis. Em segundo lugar, os dados podem ser baseados em uma amostra limitada de dados. Isso significa que podemos ter apenas uma amostra das atividades realizadas. Fatores externos, tais como más condições meteorológicas, podem afetar o número total de dados que são coletados para alguns lugares, especialmente locais ao ar livre. Por isso, antes de tirar conclusões com dados de RSPs, é necessário realizar uma comparação com dados obtidos de maneira tradicional (*offline*). Existem várias fontes de dados disponíveis na Web que permitem realizar validações, algumas delas foram ilustradas em [Barbosa et al. 2014].

Para investigar a precisão do método para a identificação das fronteiras culturais [Silva et al. 2014c], os autores compararam os resultados obtidos com os resultados do World Values Surveys (considerado o mais importante da área disponível publicamente), que utilizou dados coletados de forma tradicional: questionários. Além disso, em [Silva et al. 2013c] os autores utilizaram dados do TripAdvisor (um popular *website* de turismo), para verificar se os resultados dos pontos turísticos encontrados eram relevantes na cidade estudada. Outro exemplo pode ser encontrado em [Cranshaw et al. 2012], onde os autores entrevistaram moradores da cidade estudada para validar os resultados obtidos.

2.7.8. Tratamentos Estatísticos

No estudo de propriedades de grafos, um modelo nulo (*null model*) é um grafo que corresponde a um grafo específico em algumas das suas propriedades estruturais, mas que também é considerado um exemplo de um grafo aleatório. O modelo nulo é usado, geralmente, como um termo de comparação para verificar se o grafo em questão apresenta alguma característica, como a estrutura de uma comunidade, ou não. Nem todos os modelos nulos são iguais. A criação de cada um depende de quais propriedades serão preservadas no grafo original e, também, qual é a hipótese nula sendo feita. Nas técnicas City Image e identificação de POIs, mencionadas anteriormente, foram utilizados modelos nulos que mantinham os vértices de grafos de transição urbana e distribuíam o mesmo número de arestas originais de forma

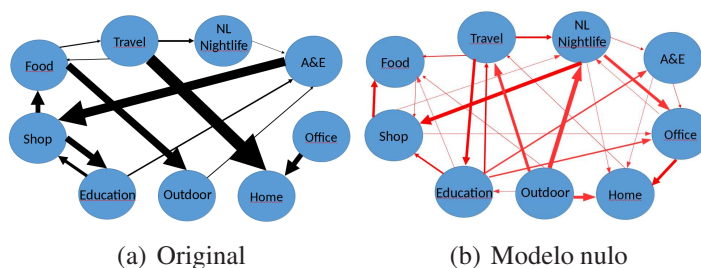


Figura 2.31. Grafo de transição urbana original e um modelo nulo.

aleatória (procedimento ilustrado na Figura 2.31, onde a espessura da aresta representa o seu peso). Esses modelos serviram para identificar transições que poderiam ser geradas de forma aleatória, bem como identificar transições populares. Além disso, este procedimento é importante porque pode nos impedir de usar dados que não têm relação com os fenômenos que estamos interessados.

Outro procedimento necessário para comparar propriedades de diferentes regiões (por exemplo, cidades), é realizar uma normalização dos dados [Freedman 2009]. Além disso, também é interessante verificar a razão de dados por habitantes, valor que, idealmente, deve ser similar para todas as regiões estudadas. Existem várias outras técnicas para realizar validações e tratamento estatístico de dados de RSPs para estudar o funcionamento de cidades e o comportamento urbano. A principal mensagem aqui é ter em mente que esses procedimentos são extremamente necessários e devem ser usados sempre que possível.

2.7.9. Aterfatos para o Estudo de Mobilidade

Nesta seção, apresentamos mais detalhes sobre algumas das principais técnicas que foram utilizadas para o estudo de mobilidade em trabalhos citados anteriores.

Raio de Giro. O raio de giro mede a frequência e o quão longe um usuário se move. Um baixo raio de giro normalmente indica um usuário que viaja principalmente localmente (com poucos compartilhamentos de dados de longa distância), enquanto um alto raio de giro indica um usuário com muitos compartilhamento de dados de longa distância [Cheng et al. 2011]. O raio de giro de um usuário pode ser formalizado como:

$$r_g = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{r}_i - \mathbf{r}_{cm})^2}$$
, onde n é o número de dados do usuário, e $\mathbf{r}_i - \mathbf{r}_{cm}$ é a distância entre um determinado dado \mathbf{r}_i e centro de massa do usuário \mathbf{r}_{cm} (que é uma média simples da localização de todos os dados).

Modelo de gravidade. Na sua formulação mais simples, o modelo de gravidade (*Gravity Model* [Zipf 1946]) afirma que a interação $T_{i,j}$ entre dois locais i e j é proporcional ao produto das suas populações P_i e P_j sobre a sua distância $d_{i,j}$:

$$T_{i,j} = k \frac{P_i^\alpha P_j^\beta}{d_{i,j}^\gamma}$$
, onde os expoentes α , β , γ e o fator de escala k são parâmetros ajustáveis, praticamente escolhidos de modo a ajustar os dados empíricos que estão sendo modelados. Ao longo dos anos, este modelo simples foi ampliado e refinado em uma variedade de maneiras, e sua aplicação tem ocorrido além do domínio de transportes, o que mostra o seu potencial. Por exemplo, ele foi usado para modelar telefonemas interurbanos

[Krings et al. 2009] e a propagação de doenças infecciosas [Balcan et al. 2009].

Fórmula de Haversine. A fórmula de Haversine é uma importante equação usada em navegação, fornecendo distâncias entre dois pontos de uma esfera a partir de suas latitudes e longitudes, podendo ser calculada como:

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$
, onde d é a distância entre os dois pontos (ao longo de um grande círculo, *great circle*, da esfera); r é o raio da esfera (raio da Terra = 6372,8 km); ϕ_1, ϕ_2 : latitude do ponto 1 e latitude do ponto 2 e λ_1, λ_2 : longitude do ponto 1 e longitude do ponto 2.

Existem várias outras técnicas e modelos que auxiliam o estudo de mobilidade urbana utilizando dados de RSPs, além dos trabalhos citados, mais informações podem ser encontradas em outras áreas que também realizam estudos neste campo, como geografia, planejamento urbano e física.

2.7.10. Tecnologias e Ferramentas para a Análise de Dados

A análise dos dados urbanos é fundamental, pois para utilizar esses dados é necessário conhecer suas propriedades. O objetivo desta seção é apresentar algumas das tecnologias e ferramentas comumente utilizadas para essa finalidade.

Quando se trata de análise de dados, a linguagem Python tanto quanto as linguagens Matlab/Octave ou R são relativamente fáceis para começar a codificar. A seguir mostramos algumas das vantagens de cada uma destas ferramentas.

Quando usar as linguagens Matlab ou R: Matlab e R possuem uma longa e confiável história, uma comunidade forte e com uso na indústria. Isto significa que se pode contar com o apoio online de outros usuários, caso precisar de ajuda ou tenha dúvidas sobre a utilização da linguagem. Além disso, há uma grande variedade de pacotes extras disponíveis publicamente. Quando é necessário fazer uma análise estatística pesada ou gráficos, Matlab ou R são as melhores opções. Operações matemáticas típicas como multiplicação de matrizes funcionam de forma bem simples e direta, bem como a visualização de dados.

Quando usar a linguagem Python: Se é necessário realizar uma limpeza e formatação dos dados, ou obter dados de *websites*, arquivos ou outras fontes de dados, a melhor escolha é Python. Em geral, devido à facilidade de transformar ideias em código e pela simplicidade de manipular *strings*, Python é uma das principais linguagens para a construção de *scripts* para o gerenciamento de dados. Como ilustramos anteriormente, existem várias APIs desenvolvidas para a linguagem Python que facilitam a coleta de dados de RSPs.

Outras ferramentas úteis na análise de dados de RSPs:

- **Estudo de grafos/redes:** Uma boa opção é o NetworkX³⁴. NetworkX é um pacote para Python que permite a criação, manipulação e estudo de grafos de forma bem simples;
- **Visualização de grafos:** Para essa finalidade, duas boas opções são o Gephi³⁵ e o Cy-

³⁴<https://networkx.github.io>.

³⁵<http://gephi.org>.

toscape³⁶. Ambos são grátis, e além de possibilitar a construção de boas visualizações, também oferecem suporte básico para a análise do grafo;

- **Manipulação de dados:** Como mencionamos anteriormente, muitas fontes de dados urbanos oferecem um volume muito grande de dados (*Big Data*). Com isso, o uso de bancos de dados NoSQL [Pokorny 2013], como o MongoDB³⁷, se tornam interessantes, pois são projetados para oferecer escalabilidade na manipulação de um volume grande de dados;
- **Análises estatísticas e visualização de dados (não espaciais) em Python:** Assim como as linguagens Matlab ou R, a linguagem Python também pode ser utilizada para realizar análises estatísticas e visualização de dados. Para que a linguagem Python tenha funcionalidades similares à linguagem Matlab ou R é necessário a instalação de pacotes específicos, por exemplo: NumPy³⁸, SciPy³⁹, Matplotlib⁴⁰ e Scikit-learn⁴¹;
- **Processo de geocodificação (*geocoding*):** Processo de conversão de endereços (por exemplo, "1600 Amphitheatre Parkway, Mountain View, CA") em coordenadas geográficas (por exemplo, latitude 37.423021 e longitude -122.083739), que podem ser usadas para a análise espacial de dados urbanos. O Google⁴², Bing Maps⁴³, Yahoo⁴⁴ e Mapquest⁴⁵ oferecem APIs para essa finalidade. Existem bibliotecas para linguagens de programação que auxiliam nessa tarefa, por exemplo, a biblioteca para Python Geopy⁴⁶;
- **Verificar existência de lei de potência (*Power law*):** Como mostramos anteriormente, ao analisar dados de RSPs podemos querer verificar se uma determinada distribuição de dados pode ser explicada por uma lei de potência. Para isso, existe um conjunto de ferramentas bastante útil disponibilizado em: <http://www.santafe.edu/~aaronc/powerlaws>. Mais detalhes sobre este processo, pode ser obtido também em [Alstott et al. 2014];
- **Análise e visualização de dados especiais:** Existem várias ferramentas para essa finalidade, no entanto mencionamos aqui apenas algumas delas. Uma das possibilidades é utilizar um Sistema de Informação Geográfica (GIS - *Geographic Information System* [Sutton et al. 2009]). Uma aplicação GIS, por exemplo o QGIS⁴⁷, auxilia no processo de manipulação de dados espaciais, possibilitando a visualização de informações em um mapa e a realização de análises espaciais diversas. Além disso, existem várias bibliotecas em diversas linguagens de programação que permitem a manipulação de

³⁶<http://www.cytoscape.org>.

³⁷<https://www.mongodb.com>.

³⁸<http://www.numpy.org>.

³⁹<http://www.scipy.org>.

⁴⁰<http://matplotlib.sourceforge.net>.

⁴¹<http://scikit-learn.org>.

⁴²<https://developers.google.com/maps/documentation/geocoding/?hl=pt-br>.

⁴³<https://msdn.microsoft.com/en-us/library/ff701715.aspx>.

⁴⁴<https://developer.yahoo.com/boss/placefinder>.

⁴⁵<http://www.mapquestapi.com/geocoding>.

⁴⁶<https://github.com/geopy/geopy>.

⁴⁷<http://www.qgis.org>.

Tabela 2.1. Comparando diferentes alternativas para visualizar e manipular dados espaciais.

	GIS (por exemplo, QGIS)	Python	R
Instalação e Configuração do ambiente de trabalho	Fácil instalação. Interfaces gráficas relativamente intuitivas. Disponível para os principais sistemas operacionais (SOs).	Geralmente possui dependências de pacotes na instalação. Independente de SO.	Depende da instalação de pacotes, processo que é geralmente fácil. Independente de SO.
Análise de dados	Bom para iniciantes.	Tende a ser a plataforma estatística preferida para cientistas com elevado conhecimento de programação. Oferece flexibilidade.	Tende a ser a plataforma estatística preferida para cientistas de áreas diversas que usam R. A curva de aprendizado é a mais baixa. Oferece flexibilidade.
Criação de um mapa	Opção mais fácil.	Requer vontade de aprender fundamentos de programação.	Bom para cientistas que já usam R.

dados espaciais. Por exemplo, o pacote `ggmap` [Kahle and Wickham 2013] baseado em R disponibiliza um conjunto de funções para visualizar dados e modelos espaciais em cima de mapas estáticos a partir de várias fontes online (por exemplo, Google Maps). Várias bibliotecas da linguagem Python também oferecem suporte para essa tarefa. Por exemplo, a biblioteca `Cartopy`⁴⁸ facilita a visualização de dados espaciais em mapas estáticos, já a biblioteca `Folium` facilita a visualização de dados espaciais em um mapa interativo, com o suporte da biblioteca `Leaflet`⁴⁹. Especificamente sobre *heatmaps*, a biblioteca Python `Heatmaps` (`jjguy`)⁵⁰, oferece *heatmaps* compatíveis com Google Earth e a Python `Heatmaps` (`sethoscope`)⁵¹, oferece *heatmaps* baseados no `OpenStreetMap`⁵². Várias outras bibliotecas baseadas em Python para trabalhar com dados espaciais pode ser encontrada aqui: [Jung 2015]. Como temos muitas opções para realizar as mesmas tarefas, a Tabela 2.1 visa apresentar mais algumas características das alternativas mencionadas com o intuito auxiliar numa eventual escolha de uso.

2.8. Desafios

Esta seção apresenta diversos desafios sobre tópicos de pesquisa atuais relacionados com computação urbana utilizando dados de redes de sensoriamento participativo.

⁴⁸<http://scitools.org.uk/cartopy>.

⁴⁹<http://leafletjs.com>.

⁵⁰<http://jjguy.com/heatmap>.

⁵¹<http://www.sethoscope.net/heatmap>.

⁵²<https://www.openstreetmap.org>.

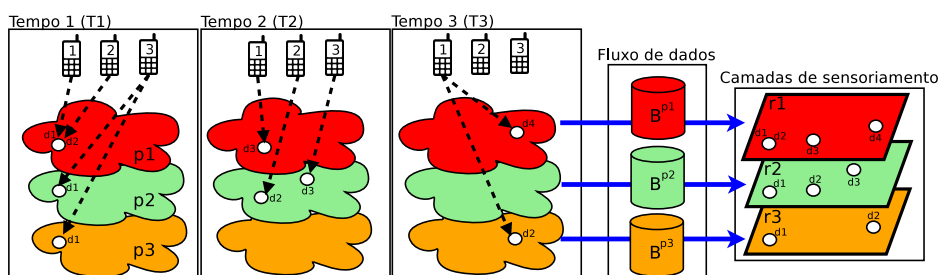


Figura 2.32. Ilustração do compartilhamento de dados em três RSPs ao longo do tempo, resultando em camadas de sensoriamento [Silva et al. 2014b].

2.8.1. Camadas de Sensoriamento

Uma camada de sensoriamento consiste de dados descrevendo aspectos específicos de uma localização geográfica. O conceito de camada de sensoriamento é bastante amplo: ele representa dados, com seus atributos, provenientes de uma determinada fonte de dados, por exemplo uma RSP particular. Cada RSP fornece acesso aos dados relacionados a certo aspecto de uma região geográfica predefinida (por exemplo, condições de tráfego, fotos de locais, etc.), e, com isso, cada RSP distinta pode ser representada como uma camada de sensoriamento [Silva et al. 2014b].

Além de RSPs, outras fontes de dados podem ser: dados disponíveis na Web não gerados por usuários, como a condição climática fornecida pela empresa Clima Tempo⁵³; ou dados de redes de sensores sem fio tradicionais. Discutimos aqui o conceito de camadas de sensoriamento para as RSPs. No entanto todos os conceitos discutidos podem ser utilizados para outras fontes de dados associadas a regiões geográficas predefinidas, com as adaptações necessárias.

A Figura 2.32 ilustra o conceito de camadas de sensoriamento. Essa figura mostra dados compartilhados em três RSPs diferentes (p1, p2 e p3), por três usuários distintos em diferentes instantes de tempo. Como discutimos na Seção 2.3, esses dados devem ser coletados (por exemplo, usando uma API) e processados, passo que inclui as tarefas de análise e padronização dos dados. Cada plano na figura representa uma camada de sensoriamento para uma região específica, por exemplo o centro de Belo Horizonte, com dados provenientes de três fontes distintas. Com isso, as camadas de sensoriamento ilustradas são: *check-ins* (r1), proveniente, por exemplo, do Foursquare; *alertas de tráfego* (r2), proveniente, por exemplo, do Waze; e *fotos de lugares* (r3), proveniente do Instagram, por exemplo.

O uso de camadas de sensoriamento de forma independente pode ser muito útil. Por exemplo, uma camada contendo informações de trânsito pode possibilitar a identificação em tempo real de rodovias com acidentes e buracos, cuja detecção é difícil com sensores tradicionais, mas torna-se mais factível quando os usuários participam do processo de sensoriamento. Os exemplos mencionados na Seção 2.6 também contribuem para esse ponto. No entanto, a grande motivação é realizar uma análise conjunta de múltiplas camadas para a construção de aplicações mais sofisticadas.

Sabemos que uma queixa comum dos habitantes das grandes cidades é o congestio-

⁵³<http://www.climatempo.com.br>.

namento. Com isso, uma aplicação que emerge naturalmente é uma que possui o objetivo de inferir as causas de congestionamento, passo fundamental para tratar o problema. Esta não é uma tarefa fácil de realizar, e o resultado pode variar entre diferentes cidades, uma vez que dependem de aspectos geográficos, culturais, econômicos, dentre outros. No entanto, a análise conjunta de diferentes camadas de sensoriamento na cidade poderia contribuir para essa aplicação. Por exemplo, poderíamos cruzar as informações fornecidas pelas camadas alertas de tráfego, *check-ins* e fotos de lugares. A primeira camada fornece dados em tempo (quase) real sobre onde estão acontecendo congestionamentos, a segunda fornece dados sobre os tipos de lugares localizados nas áreas dos congestionamentos, com isso é possível entender melhor as áreas de interesse, por exemplo, identificando o tipo da área e, finalmente, através da análise da camada fotos de lugares nós podemos obter evidência visual do que acontece em tempo real próximo das áreas durante os congestionamentos. Ao analisar conjuntamente dados destas três camadas podemos detectar, por exemplo, carros bloqueando cruzamentos, e inferir as possíveis causas disso. Obviamente, outras camadas podem também ser utilizadas, tal como a condição do clima.

Há vários desafios ao lidar com dados de várias camadas simultaneamente, como os descritos a seguir.

1. **Combinação de dados:** A fim de realizar a combinação de dados nós temos que certificar que estes são consistentes em todas as camadas. Esta é uma condição obrigatória para a correta extração de informações. Por exemplo, para combinar dados compartilhados pelo mesmo usuário em diferentes camadas pode ser um problema em RSPs, pois o mesmo usuário pode participar em diferentes camadas com diferentes identificadores. Vamos supor que queremos combinar dados de um mesmo usuário compartilhados na camada *check-ins* e na camada fotos de lugares. Como os dados dessas camadas são de sistemas independentes os usuários possuem identificações diferentes. Uma forma de tentar contornar esse problema é verificar outros sistemas com o intuito de mapear o ID do usuário de uma camada em outra. Sabemos, por exemplo, que os usuários do Foursquare e Instagram tendem a ser também usuários do Twitter. Dessa forma, a chave do processo de combinação poderia ser a identificação usada no Twitter.

Além disso, os sistemas de computação urbana são, tipicamente, obrigados a responder rapidamente a consultas dos usuários (por exemplo, para prever condições de tráfego). Sem técnicas de gerenciamento de dados que permitem organizar várias camadas de sensoriamento heterogêneas, torna-se impossível realizar processos de extração de conhecimento de forma rápida. Esses são apenas alguns dos desafios relacionados a esse tópico.

2. **Validade dos dados:** Diferentes camadas podem se referir a dados válidos para diferentes intervalos de tempo. Isso é natural porque algumas fontes de dados fornecem dados em tempo (quase) real, outras não. Por exemplo, um alerta no Waze refere-se a uma situação de trânsito que pode não existir cinco minutos mais tarde. No entanto, dados do censo geralmente são válidos por um grande intervalo de tempo, meses ou anos, até o próximo censo ser publicado. Temos de estar cientes de todas essas questões ao projetar novas aplicações.

2.8.2. Dinâmica Temporal

Um aspecto pouco explorado nas análises dos dados de RSPs é o temporal. Nesse contexto, a análise de características temporais permite aprimorar tais aplicações, bem como gerar novas oportunidades de pesquisa [Gao et al. 2013, Yuan et al. 2013]. A maioria dos estudos encontrados na literatura consideram que os dados compartilhados por usuários formam RSPs estáticas, sendo a dinâmica temporal negligenciada. Isso pode acarretar em perda de informações importantes. Por exemplo, enquanto duas regiões de uma cidade podem apresentar comportamento similar nos dados agregados durante um dia, elas podem ter diferenças quando uma perspectiva temporal é considerada na análise das atividades mais populares em cada turno.

Para ilustrar algumas iniciativas nesse sentido, em [Zhang et al. 2013] os autores analisaram atividades urbanas a partir de dados do Foursquare considerando a dinâmica temporal, a partir da divisão dos dados em períodos do dia. Os autores identificaram, por exemplo, que a atividade “comida” pode não ser tão ativa no período da tarde, mas sim, nos períodos da manhã e noite, de forma que na visão agregada não se percebe essa diferença. Essa abordagem é interessante para mostrar que determinadas atividades são pertinentes em um determinado período do dia, mas quando analisadas de forma agregada podem não ser relevantes ou podem não capturar o real comportamento dos usuários. Outro exemplo é a técnica City Image que apresentamos na Seção 2.6.1. No exemplo da Figura 2.19, a perspectiva temporal utilizada é a divisão dos dados em dias de semana/final de semana durante o dia e a noite, a partir disso foi feita uma análise dos dados por partições. Podemos perceber com o auxílio dessas imagens que existe variação significativa entre dia e noite nas duas cidades analisadas. Além disso, a imagem agregada (sem considerar partições) é bastante diferente das desagregadas, como foi mostrado em [Silva et al. 2012].

Os trabalhos descritos anteriormente fornecem indícios das vantagens da utilização de informação temporal dos dados obtidos de RSPs. No entanto, se por um lado investigar a dinâmica temporal de uma RSP é uma oportunidade para obtenção de informações mais próximas da realidade do comportamento da rede, por outro, surgem novos desafios ao adicionarmos uma dimensão temporal ao estudo, como os descritos a seguir:

1. **Janelas de tempo:** Trabalhos que analisam a questão temporal geralmente fragmentam os dados em intervalos de tempo (e.g., manhã, tarde e noite) denominados janelas. No entanto, a definição adequada do tamanho da janela é um problema, pois é necessário definir um tamanho de janela que capture dinâmicas relevantes. Nesse caso, existem inúmeras oportunidades para novas abordagens que consideram janelas com tamanhos flexíveis;
2. **Modelagem:** Geralmente, dados de uma RSP são representados como um conjunto de entidades, por exemplo, usuários ou PDIs (pontos de interesses), e suas relações (e.g., transições ou comunicação). Como a contribuição desses dados pode variar muito ao longo do tempo um modelo baseado em grafos estáticos pode não ser suficiente para capturar essa dinamicidade. Por exemplo, dados obtidos a partir do Foursquare possuem informações espaço-temporais, tais como posicionamento dos usuários e os momentos de interação. Portanto, um desafio é modelar a dinâmica espaço-temporal a fim de entender melhor, por exemplo, diversos aspectos dessa participação dos usuá-

rios. Nesse sentido, grafos temporais [Kostakos 2009] surgem como alternativa promissora que pode ser utilizada para entendimento da dinâmica espaço-temporal. Em um grafo temporal, as relações entre as entidades podem ser modeladas como arestas que podem ser criadas e destruídas ao longo do tempo. Por exemplo, entender aspectos temporais de interações entre usuários com certos locais na cidade. Dessa forma, utilizando grafos temporais para modelar uma RSP, podemos aplicar tanto conceitos de grafos (e.g., componentes conectados e caminhamento) como métricas de centralidade (e.g., centralidade de intermediação e centralidade de proximidade) para auxiliar no entendimento da dinâmica das RSPs [Nicosia et al. 2013].

2.8.3. Mecanismos de Incentivos para a Aquisição de Dados

Um ponto fundamental para as RSPs é a colaboração dos usuários, pois as aplicações em uma RSP dependem de que os usuários estejam dispostos a coletarem, processarem e transmitir os dados sensorizados [Lee and Hoh 2010]. A colaboração entre os participantes de uma RSP reflete diretamente na qualidade e quantidade dos dados sensorizados e, consequentemente, na melhoria dos serviços oferecidos com esses dados.

No entanto, como estas aplicações consomem recursos do dispositivo do usuário, o mesmo pode ser relutante em contribuir com a rede. Diversos são os motivos que podem fazer um usuário usufruir, porém não colaborar com a RSP, tais como poupar bateria, evitar gastos com a transmissão de dados, ou mesmo por questões de privacidade [Lee and Hoh 2010].

Nos últimos anos, foram propostos dezenas de mecanismos de incentivo e realizados diversos experimentos para entender o comportamento cooperativo. Alguns desses mecanismos podem recompensar o participante por meio de pagamentos reais, virtuais ou outros prêmios. Outros mecanismos visam transformar a tarefa de sensoriamento em uma tarefa mais prazerosa e estimulante para o usuário, adicionando elementos comuns em jogos, como elementos de disputa entre os usuários (mecanismos conhecidos como baseados em *gamificação*).

Existem mecanismos que visam permitir que o usuário participe da decisão sobre a tarefa que irá realizar e sobre o pagamento que irá receber da RSP. Outros visam melhorar a qualidade dos dados obtidos e minimizar os custos com sensoriamento. Podemos mencionar também mecanismos em que o usuário negocia com a plataforma o valor da recompensa pelos dados sensorizados antes de enviá-los e, ainda, os que a plataforma decide quanto irá pagar pelos dados já enviados pelo usuário.

Um mecanismo de incentivo é eficiente se ele recruta mais participantes para a RSP e mantém esses participantes ativos no sistema. Com isso alguns dos desafios relacionados aos mecanismos de incentivos para as RSPs são:

1. **Custos:** Para que o desenvolvimento de mecanismos de incentivo monetários sejam eficientes, deve-se considerar os custos para a plataforma da RSP e os ganhos para o participante da rede. Esses mecanismos utilizam um custo máximo para a plataforma RSP que será pago aos participantes ativos da rede. No entanto, encontrar e decidir um valor que minimize o custo para plataforma e, ao mesmo tempo, motive o usuário requer investigações futuras [Gao et al. 2015].

2. **Validação das propostas:** A maioria das propostas de mecanismos de incentivo utilizam uma validação teórica ou pequenos experimentos controlados. No entanto, estes experimentos podem não prever com alta precisão a participação dos usuários ao longo do tempo na plataforma. No caso de mecanismos baseados em *gamificação*, embora diversas RSPs de sucesso no mercado utilizem este conceito, aplicar uma estratégia com sucesso prévio em uma nova RSP não é garantia que funcionará. Talvez exista alguns elementos que funcionem para determinados tipos de RSPs e outros não.

2.8.4. Qualidade de Dados

A qualidade de dados é um tópico amplamente estudado pela comunidade científica. No entanto, ainda existem desafios únicos para controlar a qualidade dos dados compartilhados quando se lida com contribuições de usuários ubíquos.

Em geral, como discutimos anteriormente, os dados coletados de RSPs são, após processados, utilizados para a extração de informações contextuais, que são fundamentais para os sistemas sensíveis ao contexto [Dey and Abowd 2000].

Alguns dos principais desafios relacionados à qualidade dos dados em uma RSP são:

1. **Erros de leitura:** Um desafio que pode afetar a precisão dos dados das RSPs são possíveis erros de leitura de equipamentos. Por exemplo, um GPS pode estar mal calibrado e gerar dados cuja imprecisão está além do limite aceitável para este tipo de dado. Embora alguns erros possam parecer totalmente toleráveis, dependendo dos requisitos de uma aplicação, é possível que os limites mais restritos de precisão sejam fundamentais para sua correta operação.
2. **Ausência de estrutura:** Os dados compartilhados em RSPs, em alguns casos, são de texto livre, não apresentando uma estrutura semântica nem codificadas. Essa liberdade dada aos usuários permite que eles postem o que querem, mesmo informações incorretas, e em diferentes formatos. Por exemplo, um usuário poderia descrever um acidente em outra língua ou utilizando gírias através de algum *microblogging* como o Twitter. Com isso, o processamento dos dados se torna complexo e suscetível a erros, uma vez que há a possibilidade de dados distintos serem confundidos como um mesmo dado, ou ainda a duplicidade de dados, isto é, dados idênticos serem identificados como distintos devido a diferenças no preenchimento dos campos.
3. **Poluição dos dados:** Este desafio diz respeito à possibilidade dos dados estarem incorretos devido a um comportamento malicioso dos usuários [Coen-Porisini and Sicari 2012, Mashhadi and Capra 2011]. Podemos encontrar comportamentos maliciosos em várias esferas sociais, e o mesmo também pode ocorrer nas RSPs. Por exemplo, usuários de sistemas para compartilhamento de alertas de trânsito, como o Waze, podem gerar falsos alertas de congestionamento ou acidentes, com o intuito de incentivar os demais usuários a não utilizar determinadas vias de seu trajeto. Este comportamento malicioso poderá ocasionar em falsos positivos, por exemplo, na detecção de eventos.

2.8.5. Outros Desafios

Outra questão importante é lidar com um grande volume de dados que as RSPs podem oferecer, impondo desafios para armazenamento, processamento e indexação em tempo real usando ferramentas de gerenciamento de banco de dados tradicionais ou aplicações de processamento de dados. Isso faz com que a oferta de serviços em tempo real usando uma rede de sensoriamento participativo seja um desafio. Para resolver esta questão, precisamos de métodos para armazenar, mover e processar de forma eficaz grandes quantidades de dados. Novos paradigmas algorítmicos devem ser projetados, bem como técnicas de mineração de dados específicas devem ser criadas de acordo com esses novos paradigmas. Outros métodos devem contemplar abordagens de engenharia de dados para grandes redes com milhões ou bilhões de nós/arestas, incluindo compressão eficaz, pesquisa e métodos para casamento de padrões [Giannotti et al. 2012].

Felizmente, a pesquisa sobre os desafios de grandes dados é muito ativa, e recentemente fez grandes avanços, por exemplo, com base em plataformas paralelas (como o Hadoop⁵⁴), para o processamento de um grande volume de dados.

Podemos ainda mencionar que as RSPs são muito dinâmicas. Usuários contam com RSPs, como o Twitter ou Waze, para transmitir seus dados sensorizados. De posse desses dados podemos extrair conhecimento. Sistemas, como o Waze, por sua vez, podem ser realimentados com esses conhecimentos obtidos e, a partir disso, eles podem fornecer informações úteis para os usuários. Esses conhecimentos também podem ser gerados por aplicativos de terceiros. Para exemplificar, na Seção 2.6 é descrito um exemplo de aplicação que permite a identificação de regiões de interesse de uma cidade. Após usar este aplicativo, os usuários podem optar por mudar o seu comportamento, como para visitar preferencialmente áreas populares, o que pode vir a afetar o número de dados compartilhados nesses locais. Isto dá uma ideia do dinamismo de uma rede de sensoriamento participativo e os desafios que surgem nessas condições.

Além desses desafios, existe ainda a questão da privacidade do usuário. Essa questão é bastante ampla e está presente em muitas camadas do sistema. Privacidade de dados em sistemas de mídia social atualmente tem sido discutida em vários estudos, como em [Pontes et al. 2012, Toch et al. 2010, Brush et al. 2010].

2.9. Oportunidades

Nesta seção ilustramos algumas das oportunidades relacionadas com o estudo de sociedades urbanas utilizando dados de RSPs.

Detecção/previsão de trânsito: De forma geral, dados de RSPs são pouco explorados em modelos de detecção/previsão de trânsito. Alguns dos trabalhos mais próximos dessa direção são: [Silva et al. 2013e, Tostes et al. 2014]. Tostes et al. analisaram condições de trânsito em relação aos dados sensorizados por duas RSPs, Foursquare e Instagram. Como vimos na Seção 2.5 os sensores de RSPs fornecem informações importantes para o melhor entendimento da dinâmica das cidades. Por exemplo, uma mensagem geolocalizada, seja no Foursquare, Instagram ou Twitter, pode ser utilizada para melhor entendermos as condições de trânsito, como foi mostrado em [Tostes et al. 2014] (trabalho discutido na Se-

⁵⁴<http://hadoop.apache.org>.

ção 2.6.5). Além disso, imagine que um usuário faça um *check-in* em casa e depois vá para o trabalho. Quando chegar no trabalho, por algum motivo, ele faz outro *check-in*. Independente se for na mesma rede social ou não, existe uma informação intrínseca no intervalo de tempo entre esses *check-ins* que consiste no desempenho do trânsito. Se o trânsito estiver mais congestionado, esse intervalo entre *check-ins* será maior do que o tempo de viagem sem congestionamento, que é facilmente calculado pela distância e velocidade máxima das vias. Além disso, os autores também levantaram várias questões nessa direção: (i) como coletar dados de mapas online em tempo real?; (ii) é possível utilizar dados de RSPs como uma característica para previsão de trânsito intenso?

Modelagem de camadas de sensoriamento: Temos ainda oportunidades com relação à modelagem das camadas de sensoriamento, pois numa mesma camada as entidades podem possuir relações distintas entre elas. Para ilustrar essa oportunidade considere a camada de *check-ins*. Como ilustramos anteriormente, essa camada pode representar a mobilidade urbana considerando a relação entre lugares e pessoas, sendo útil para entender, por exemplo, a frequência de transição entre diferentes lugares (entidades). Outra possibilidade é modificar a modelagem do problema, para, por exemplo, estudar as preferências de pessoas. Nesse caso, a entidade a ser analisada passa a ser o usuário. Note que dados de uma mesma camada podem ser modelados de formas distintas para responder perguntas diferentes. Alguns arcabouços, como o apresentado em [Silva et al. 2014b], oferecem suporte básico para essa questão. No entanto, existem várias oportunidades para desenvolver extensões desse arcabouço a fim de oferecer serviços mais sofisticados.

Múltiplas estratégias de mecanismos de incentivo: A maioria das propostas para incentivar a colaboração nas RSPs focam em apenas uma estratégia. Porém, como observado em [Reddy et al. 2010], a utilização de mais de uma estratégia simultaneamente pode apresentar melhores resultados. Esses autores concluem que os incentivos funcionaram melhor quando os pagamentos (recompensas) foram combinados com outros fatores, como o altruísmo do usuário e quando havia competição entre os participantes. Além disso, mostraram que um pagamento justo para todos os participantes os mantinham mais motivados do que micro pagamentos.

Confiabilidade de um determinado usuário: Dados gerados por usuários mais confiáveis provavelmente terão maior qualidade. Uma possível direção nesse sentido está relacionada com a identificação de padrões comportamentais dos usuários das RSPs. Como apresentado na Figura 2.10, quando grandes quantidades de dados são agregadas, é possível, claramente, identificar padrões de comportamento no compartilhamento de dados durante diferentes dias da semana. Assumindo que este conhecimento prévio seria uma referência do comportamento esperado dos usuários de uma dada RSP, uma possibilidade seria comparar o comportamento de um determinado usuário com este padrão de referência. Por exemplo, usuários que possuem um padrão de compartilhamento muito distinto dos demais poderia representar um usuário não confiável (e.g., um robô malicioso).

Essa abordagem discutida acima, pode ser caracterizada como uma espécie de filtragem colaborativa [Adomavicius and Tuzhilin 2005]. Esta é uma estratégia utilizada por sistemas de recomendação quando não se possui um conhecimento prévio do usuário ao qual deverá se recomendar um item. Por exemplo, utilizando as preferências de outros usuários similares a este, assumindo de que suas preferências também sejam similares.

Estudo de áreas desprivilegiadas: Regiões que fornecem uma pequena quantidade de dados em comparação com outras regiões de uma mesma cidade pode indicar uma falta de acesso à tecnologia por parte da população, uma vez que o uso de RSPs geralmente necessita *smartphones* e planos de dados 3G ou 4G, que são caros em alguns países. Os resultados preliminares sobre o uso de RSPs nesses cenários demonstram boas oportunidades para a visualização de fatos interessantes. Por exemplo, analisando os dados para a cidade do Rio de Janeiro, mostrados na Figura 2.33, observa-se que é comum encontrar áreas muito pobres ao lado de áreas ricas. Repare que a atividade de sensoriamento é pequena nas áreas indicadas como áreas pobres. Esta informação pode ser útil de diversas formas, como para nortear melhores políticas públicas nessas áreas. A mesma informação pode ser obtida usando métodos tradicionais, tais como questionários, mas nessa possível nova maneira podemos ser capazes de obter esses dados de forma automática e mais barata usando RSPs. Para esse propósito, os algoritmos semelhantes ao proposto em [Cranshaw et al. 2012] poderiam ser aplicados.

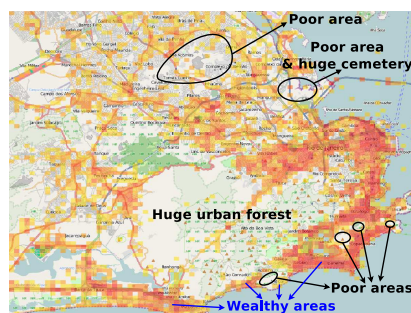


Figura 2.33. Exemplo de possível oportunidade para classificar áreas considerando a falta de dados [Silva et al. 2013a].

Classificação de áreas: Existem oportunidades de classificar áreas quando se considera conjuntamente o tempo e o local onde os *check-ins* são executadas. Pode ser possível visualizar multidões em uma cidade em tempo real. Além disso, os seres humanos possuem padrões sazonais, devido às suas rotinas. Esta sazonalidade tem um grande potencial para aplicações de previsão, uma vez que é muito provável que as pessoas repitam periodicamente suas atividades. Nós acreditamos que existem muitas oportunidades para previsão considerando o ritmo circadiano das pessoas, possibilitando a previsão, por exemplo, de multidões. Este tipo de informação é valiosa em muitos cenários, como em serviços para cidades inteligentes para evitar o tráfego em determinadas áreas e oferecer rotas alternativas para os usuários. Por exemplo, em [Hsieh et al. 2012] os autores propuseram um modelo que considera o tempo para recomendar rotas baseado em informações de RSPs para compartilhamento de localizações.

Além disso, com a utilização de dados sensorizados pelas pessoas através de RSPs é possível classificar áreas de diversas maneiras. Algumas delas foram discutidas aqui, como com relação ao cheiro, ruído e aspectos visuais. Isso pode ser útil para diversas novas aplicações e serviços. Um exemplo seria uma nova ferramenta de sugestão de rotas que sugere o menor caminho que é também o mais olfativamente agradável (as pessoas que praticam corrida de rua podem querer evitar ruas com fortes níveis de emissão de gases).

Grafos de transição urbana: Figura 2.34 mostra arestas ponderadas e nós impor-

tantes (50 maiores pesos de arestas e graus de nó) para Belo Horizonte, Cidade do México, Nova Iorque e Tóquio (considerando grafos de transição urbana, como os apresentados na seção 2.6.1, só que os nós agora são localidades individuais, ao invés de categorias de locais). Estrelas representam locais bastante visitados, setas pretas representam as arestas e círculos pretos representam *self-loops*. Quanto maior o símbolo, maior o valor. Repare que para Belo Horizonte e Cidade do México a maioria das arestas importantes são *self-loops* e de baixa distância, o que implica que as pessoas tendem a realizar atividades no bairro onde elas estão. Por outro lado, para Nova Iorque e Tóquio, cidades que são conhecidas por seus rápidos sistemas de transporte público, existe a tendência de algumas arestas ponderadas pesadas de longa distância ao longo de ligações de transporte público. Repare o potencial para serviços que dão suporte ao planejamento urbano.

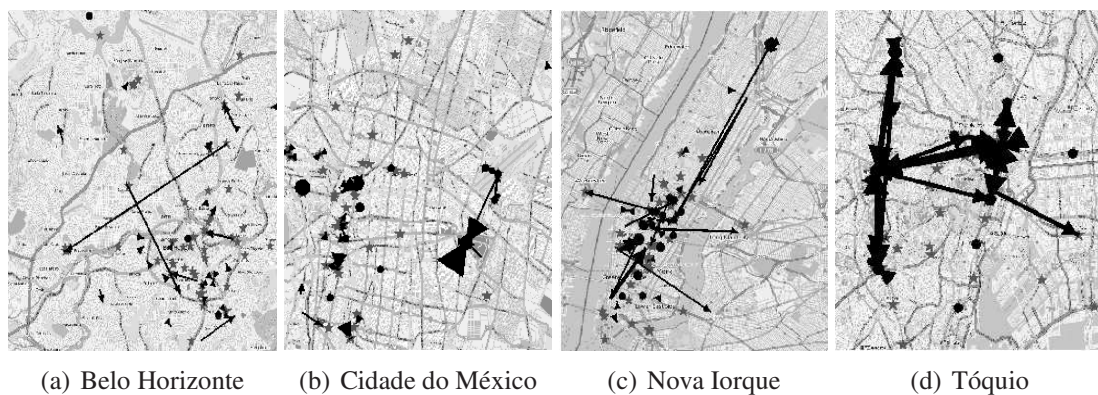


Figura 2.34. Arestas e nós importantes nos grafos de transição urbana de diversas cidades [Silva et al. 2013a].

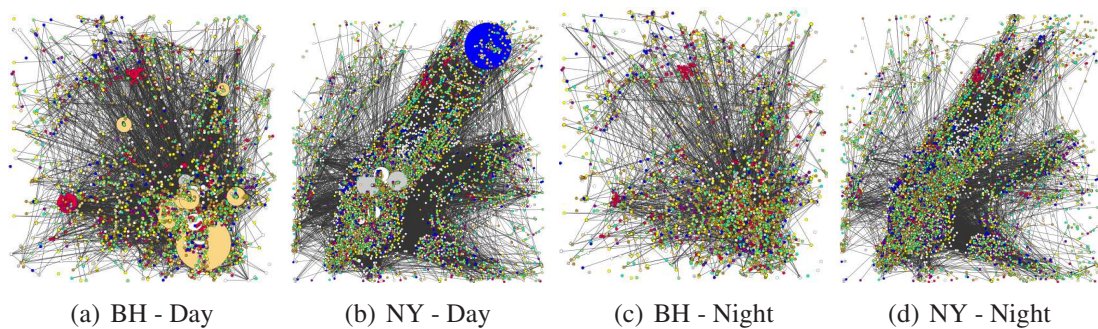


Figura 2.35. Betweenness dos nós de grafos de transição urbana para Nova Iorque e Belo Horizonte [Silva et al. 2014d].

Nessa mesma direção, podemos estudar métricas de centralidade nessa rede. Por exemplo, na Figura 2.35, mostramos os valores de centralidade de intermediação para os nós da rede. Cada cor é relacionada com uma categoria de local, e o tamanho do símbolo reflete a proporção do valor calculado. Esta abordagem pode ser utilizada para apoiar diversas aplicações, por exemplo, se for verificado um fluxo incomum e constante de pessoas entre dois locais de negócios independentes em uma cidade, os proprietários poderiam assinar um acordo comercial para aumentar suas receitas, como fazendo propaganda entre suas empresas.

Além disso, a técnica City Image, mencionada anteriormente, pode ser expandida para considerar subcategorias de locais, em vez de categorias principais. Como os dados de RSPs são altamente enviesados, algumas das transições mais populares entre as subcategorias devem ser bons indicadores da dinâmica da cidade. Essa técnica pode ser útil como uma forma de medir a distância entre duas cidades, permitindo a comparação de cidades e um agrupamento mundial que poderia ser interessante para sistemas de recomendação.

2.10. Conclusão

Neste minicurso, discutimos o conceito de computação urbana. Mostramos a relevância da área e motivamos a construção de novas aplicações para tratar questões relacionadas com a dinâmica da cidade e do comportamento social urbano. Discutimos também o sensoriamento urbano com redes de sensoriamento participativo. Mostramos que as RSPs oferecem oportunidades sem precedentes de acesso a dados de sensoriamento em escala planetária, dados que nos ajudam a entender melhor sociedades urbanas.

Estudamos também as principais técnicas utilizadas em trabalhos da área de computação urbana, bem como algumas das tecnologias e ferramentas comumente utilizadas para a análise de dados urbanos. Além disso, apresentamos diversos desafios sobre tópicos de pesquisa atuais relacionados com computação urbana utilizando dados de RSPs. Ressaltamos também várias oportunidades relacionadas ao uso de dados de RSPs em novos serviços e aplicações da área de computação urbana.

Agradecimentos

Gostaríamos de agradecer a fundamental ajuda de Pedro O. S. Vaz de Melo e Jussara M. Almeida para a realização deste trabalho. Gostaríamos de agradecer também aos alunos: Vinícius Mota, João Borges Neto, Clayson Celes, Felipe Cunha, Anna Ribeiro, Ana Ferreira e Virginia Kesting. Este trabalho é financiado parcialmente com o apoio das agências: CNPq, CAPES, FAPEMIG e Fundação Araucária.

Referências

- [Adomavicius and Tuzhilin 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749.
- [Alstott et al. 2014] Alstott, J., Bullmore, E., and Plenz, D. (2014). powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS ONE*, 9(1):e85777.
- [Balcan et al. 2009] Balcan, D., Colizza, V., Gonçalves, B., Hu, H., Ramasco, J. J., and Vespignani, A. (2009). Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences*, 106(51):21484–21489.
- [Barbosa et al. 2014] Barbosa, L., Pham, K., Silva, C., Vieira, M. R., and Freire, J. (2014). Structured open urban data: understanding the landscape. *Big data*, 2(3):144–154.
- [Barth 1969] Barth, F. (1969). *Ethnic groups and boundaries: the social organization of culture difference*. Scandinavian university books. Little, Brown.
- [Benevenuto et al. 2011] Benevenuto, F., Almeida, J. M., and Silva, A. S. (2011). Explorando redes sociais online: Da coleta e análise de grandes bases de dados às aplicações. *Proc. of SBRC'11*, pages 63–94.
- [Blei et al. 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

- [Bollen et al. 2011] Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.
- [Brenner and Smith 2013] Brenner, J. and Smith, A. (2013). 72% of online adults are social networking site users. <http://goo.gl/HTgNy3>.
- [Brockmann et al. 2006] Brockmann, D., Hufnagel, L., and Geisel, T. (2006). The scaling laws of human travel. *Nature*, 439(7075):462–465.
- [Brush et al. 2010] Brush, A. B., Krumm, J., and Scott, J. (2010). Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Proc. of Ubicomp '10*, pages 95–104, Copenhagen, Denmark. ACM.
- [Burke et al. 2006] Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., and Srivastava, M. B. (2006). Participatory sensing. In *Proc. of Workshop on World-Sensor-Web (WSW'06)*, pages 117–134, Boulder, USA.
- [Burt 1992] Burt, R. S. (1992). *Structural Holes: The Social Structure of Competition*. Harvard University Press.
- [CENS/UCLA] CENS/UCLA. *Participatory Sensing / Urban Sensing Projects*. <http://research.cens.ucla.edu/>.
- [Cha et al. 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. (2010). Measuring user influence in twitter: The million follower fallacy. In *Proc. of ICWSM'10*, Washington, USA.
- [Chatterjee and Hadi 2015] Chatterjee, S. and Hadi, A. S. (2015). *Regression analysis by example*. John Wiley & Sons.
- [Cheng et al. 2011] Cheng, Z., Caverlee, J., Lee, K., and Sui, D. Z. (2011). Exploring Millions of Footprints in Location Sharing Services. In *Proc. of ICWSM'11*, Barcelona, Spain.
- [Cho et al. 2011] Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proc. KDD '11*, pages 1082–1090, San Diego, USA. ACM.
- [Clauset et al. 2009] Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703.
- [Coen-Porisini and Sicari 2012] Coen-Porisini, A. and Sicari, S. (2012). Improving data quality using a cross layer protocol in wireless sensor networks. *Comput. Netw.*, 56(17):3655–3665.
- [Crandall et al. 2009] Crandall, D. J., Backstrom, L., Huttenlocher, D., and Kleinberg, J. (2009). Mapping the world's photos. In *Proc. of WWW '09*, pages 761–770, Madrid, Spain. ACM.
- [Cranshaw et al. 2012] Cranshaw, J., Schwartz, R., Hong, J. I., and Sadeh, N. (2012). The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City. In *Proc. of ICWSM'12*, Dublin, Ireland.
- [Dey and Abowd 2000] Dey, A. K. and Abowd, G. D. (2000). Towards a Better Understanding of Context and Context-Awareness. In *Proc. of CHI 2000 Workshops*, The Hague, The Netherlands.
- [D'Hondt et al. 2013] D'Hondt, E., Stevens, M., and Jacobs, A. (2013). Participatory noise mapping works! an evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring. *Pervasive and Mobile Computing*, 9(5):681–694.
- [Duggan and Smith 2014] Duggan, M. and Smith, A. (2014). Social media update 2013. <http://goo.gl/JhuiOG>.
- [Easley and Kleinberg 2010] Easley, D. and Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.
- [Eric Paulos and Townsend 2004] Eric Paulos, K. A. and Townsend, A. (2004). Ubicomp in the urban frontier. In *Workshop at Ubicomp'04*, Nottingham, UK.
- [Ester et al. 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of KDD-96*, Portland, USA.

- [Freedman 2009] Freedman, D. A. (2009). *Statistical models: theory and practice*. Cambridge University Press.
- [Ganti et al. 2011] Ganti, R., Ye, F., and Lei, H. (2011). Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39.
- [Gao et al. 2015] Gao, H., Liu, C., Wang, W., Zhao, J., Song, Z., Su, X., Crowcroft, J., and Leung, K. (2015). A survey of incentive mechanisms for participatory sensing. *Communications Surveys Tutorials, IEEE*, PP(99):1–1.
- [Gao et al. 2013] Gao, H., Tang, J., Hu, X., and Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. In *Proc. of RecSys '13*, pages 93–100, Hong Kong, China.
- [Giannotti et al. 2012] Giannotti, F., Pedreschi, D., Pentland, A., Lukowicz, P., Kossmann, D., Crowley, J., and Helbing, D. (2012). A planetary nervous system for social mining and collective awareness. *The Eur. Phys. Jour. Special Topics*, 214(1):49–75.
- [Gomide et al. 2011] Gomide, J., Veloso, A., Jr., W. M., Almeida, V., Benevenuto, F., Ferraz, F., and Teixeira, M. (2011). Dengue surveillance based on a computational model of spatio-temporal locality of twitter. In *Proc. of WebSci'11*, Evanston, USA.
- [Gonzalez et al. 2008] Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L. (2008). Understanding individual human mobility patterns. *Nature*, 453(7196):779–782.
- [Gonçalves et al. 2013] Gonçalves, P., Araújo, M., Benevenuto, F., and Cha, M. (2013). Comparing and combining sentiment analysis methods. In *Proceedings of the 1st ACM Conference on Online Social Networks (COSN'13)*, Boston, USA.
- [Harrell 2013] Harrell, F. E. (2013). *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer Science & Business Media.
- [Hartigan and Wong 1979] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108.
- [Hochman and Schwartz 2012] Hochman, N. and Schwartz, R. (2012). Visualizing instagram: Tracing cultural visual rhythms. In *Proc. of ICWSM'12*, pages 6–9, Dublin, Ireland. AAAI.
- [Hsieh et al. 2012] Hsieh, H.-P., Li, C.-T., and Lin, S.-D. (2012). Exploiting large-scale check-in data to recommend time-sensitive routes. In *Proc. of UrbComp '12*, pages 55–62, Beijing, China. ACM.
- [Instagram 2014] Instagram (2014). Instagram today: 200 million strong. <http://blog.instagram.com/post/80721172292/200m>.
- [Jolliffe 2002] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, second edition.
- [Joseph et al. 2012] Joseph, K., Tan, C. H., and Carley, K. M. (2012). Beyond local, categories and friends: clustering foursquare users with latent topics. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 919–926, Pittsburgh, USA. ACM.
- [Jung 2015] Jung, M. (2015). *Essential Python Geospatial Libraries*. Independent Group. https://github.com/SpatialPython/spatial_python/blob/master/packages.md.
- [Kahle and Wickham 2013] Kahle, D. and Wickham, H. (2013). ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161.
- [Karamshuk et al. 2013] Karamshuk, D., Noulas, A., Scellato, S., Nicosia, V., and Mascolo, C. (2013). Geospotting: Mining online location-based services for optimal retail store placement. In *Proc. of KDD '13*, pages 793–801, Chicago, Illinois, USA. ACM.
- [Kaufman and Rousseeuw 2009] Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.
- [Kindberg et al. 2007] Kindberg, T., Chalmers, M., and Paulos, E. (2007). Guest editors' introduction: Urban computing. *IEEE Pervasive Computing*, 6(3):18–20.

- [Kisilevich et al. 2010] Kisilevich, S., Krstajic, M., Keim, D., Andrienko, N., and Andrienko, G. (2010). Event-based analysis of people’s activities and behavior using flickr and panoramio geotagged photo collections. In *Proc. of Conf. on Inf. Vis.*, pages 289–296, London, UK. IEEE.
- [Kostakos 2009] Kostakos, V. (2009). Temporal graphs. *Physica A: Statistical Mechanics and its Applications*, 388(6):1007–1023.
- [Kostakos and O’Neill 2008] Kostakos, V. and O’Neill, E. (2008). {City ware: Urban Computing to Bridge Online}. *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*.
- [Krings et al. 2009] Krings, G., Calabrese, F., Ratti, C., and Blondel, V. D. (2009). Urban gravity: a model for inter-city telecommunication flows. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):L07003.
- [Lane et al. 2010] Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. (2010). A survey of mobile phone sensing. *Comm. Mag., IEEE*, 48(9):140–150.
- [Lane et al. 2008] Lane, N. D., Eisenman, S. B., Musolesi, M., Miluzzo, E., and Campbell, A. T. (2008). Urban sensing systems: Opportunistic or participatory? In *Proc. of HotMobile ’08*, pages 11–16, Napa Valley, California. ACM.
- [Lee and Hoh 2010] Lee, J.-S. and Hoh, B. (2010). Dynamic pricing incentive for participatory sensing. *Pervasive and Mobile Computing*, 6(6):693–708.
- [Loureiro et al. 2003] Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., Mini, R. A., Nakamura, E. F., and Figueiredo, C. M. S. (2003). Redes de sensores sem fio. *Proc. of SBRC’03*, pages 179–226.
- [Luxburg 2007] Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.
- [MacKerron and Mourato 2013] MacKerron, G. and Mourato, S. (2013). Happiness is greater in natural environments. *Global Environmental Change*, 23(5):992–1000.
- [Maisonneuve et al. 2009] Maisonneuve, N., Stevens, M., Niessen, M. E., and Steels, L. (2009). Noisetube: Measuring and mapping noise pollution with mobile phones. In *Information Technologies in Environmental Engineering*, pages 215–228. Springer.
- [Martine et al. 2007] Martine, G., Marshall, A., et al. (2007). State of world population 2007: unleashing the potential of urban growth. In *State of world population 2007: unleashing the potential of urban growth*. UNFPA.
- [Mashhadi and Capra 2011] Mashhadi, A. J. and Capra, L. (2011). Quality Control for Real-time Ubiquitous Crowdsourcing. In *Proc. of UbiCrowd’11*, pages 5–8, Beijing, China.
- [Nazir et al. 2008] Nazir, A., Raza, S., and Chuah, C.-N. (2008). Unveiling facebook: A measurement study of social network based applications. In *Proc. of IMC ’08*, pages 43–56, Vouliagmeni, Greece.
- [Newman 2010] Newman, M. (2010). *Networks: an introduction*. Oxford University Press, Inc.
- [Newman 2003] Newman, M. E. J. (2003). The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256.
- [Ng et al. 2002] Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856.
- [Nguyen and Szymanski 2012] Nguyen, T. and Szymanski, B. K. (2012). Using location-based social networks to validate human mobility and relationships models. *arXiv preprint arXiv:1208.3653*.
- [Nicosia et al. 2013] Nicosia, V., Tang, J., Mascolo, C., Musolesi, M., Russo, G., and Latora, V. (2013). Graph metrics for temporal networks. In *Temporal Networks*, pages 15–40. Springer.
- [Noulas et al. 2011a] Noulas, A., Scellato, S., Mascolo, C., and Pontil, M. (2011a). An Empirical Study of Geographic User Activity Patterns in Foursquare. In *Proc. of ICWSM’11*, Barcelona, Spain.

- [Noulas et al. 2011b] Noulas, A., Scellato, S., Mascolo, C., and Pontil, M. (2011b). Exploiting Semantic Annotations for Clustering Geographic Areas and Users in Location-based Social Networks. In *Proc. of the Fifth Int'l Conf. on Weblogs and Social Media (ICWSM'11)*, Barcelona, Spain. AAAI.
- [Paulos and Goodman 2004] Paulos, E. and Goodman, E. (2004). The familiar stranger: anxiety, comfort, and play in public places. In *Proc. of CHI'04*, pages 223–230, Vienna, Austria. ACM.
- [Poblete et al. 2011] Poblete, B., Garcia, R., Mendoza, M., and Jaimes, A. (2011). Do all birds tweet the same?: characterizing twitter around the world. In *Proc. of CIKM*, pages 1025–1030, Glasgow, UK. ACM.
- [Pokorny 2013] Pokorny, J. (2013). Nosql databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1):69–82.
- [Pontes et al. 2012] Pontes, T., Magno, G., Vasconcelos, M., Gupta, A., Almeida, J., Kumaraguru, P., and Almeida, V. (2012). Beware of what you share: Inferring home location in social networks. In *Proc. of ICDMW*, pages 571–578, Brussels, Belgium.
- [Quercia et al. 2012] Quercia, D., Capra, L., and Crowcroft, J. (2012). The social world of twitter: Topics, geography, and emotions. In *Proc. of ICWSM'12*, Dublin, Ireland.
- [Quercia et al. 2014] Quercia, D., Schifanella, R., and Aiello, L. M. (2014). The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media, HT '14*, pages 116–125, New York, NY, USA. ACM.
- [Quercia et al. 2015] Quercia, D., Schifanella, R., Aiello, L. M., and McLean, K. (2015). Smelly maps: The digital life of urban smellscapes. In *Proc. of ICWSM'15*, Oxford, UK.
- [Reddy et al. 2010] Reddy, S., Estrin, D., Hansen, M., and Srivastava, M. (2010). Examining micro-payments for participatory sensing data collections. In *Proc. of Ubicomp '10*, pages 33–36, Copenhagen, Denmark. ACM.
- [Reis et al. 2015] Reis, J., Benevenuto, F., Vaz de Melo, P., Prates, R., Kwak, H., and An, J. (2015). Breaking the news: First impressions matter on online news. In *Proceedings of the 9th International AAAI Conference on Web-Blogs and Social Media*, Oxford, UK.
- [Sakaki et al. 2010a] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010a). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of WWW'10*, pages 851–860, Raleigh, USA.
- [Sakaki et al. 2010b] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010b). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of WWW'10*, pages 851–860, Raleigh, USA. IW3C2.
- [Shannon 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell system tech. jour.*, 27.
- [Silva et al. 2014a] Silva, T., Vaz De Melo, P., Almeida, J., and Loureiro, A. (2014a). Large-scale study of city dynamics and urban social behavior using participatory sensing. *Wireless Communications, IEEE*, 21(1):42–51.
- [Silva et al. 2014b] Silva, T. H., Vaz de Melo, P., Almeida, J., Viana, A., Salles, J., and Loureiro, A. (2014b). Participatory Sensor Networks as Sensing Layers. In *Proc. of SocialCom'14*, Sydney, Australia.
- [Silva et al. 2012] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., and Loureiro, A. A. F. (2012). Visualizing the invisible image of cities. In *Proc. IEEE CPScom'12*, pages 382–389, Besancon, France.
- [Silva et al. 2013a] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., and Loureiro, A. A. F. (2013a). Challenges and opportunities on the large scale study of city dynamics using participatory sensing. In *Proc. of IEEE ISCC'13*, pages 528–534, Split, Croatia.
- [Silva et al. 2013b] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., and Loureiro, A. A. F. (2013b). Uma Fotografia do Instagram: Caracterização e Aplicação. In *Proc. of SBRC'13*, Brasília, Brazil.
- [Silva et al. 2014c] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., Musolesi, M., and Loureiro, A. A. F. (2014c). You are What you Eat (and Drink): Identifying Cultural Boundaries by Analyzing Food & Drink Habits in Foursquare. In *Proc. of ICWSM'14*, Ann Arbor, USA.

- [Silva et al. 2013c] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., Salles, J., and Loureiro, A. A. F. (2013c). A picture of Instagram is worth more than a thousand words: Workload characterization and application. In *Proc. of DCOSS'13*, pages 123–132, Cambridge, USA.
- [Silva et al. 2013d] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., Salles, J., and Loureiro, A. A. F. (2013d). A comparison of foursquare and instagram to the study of city dynamics and urban social behavior. In *Proc. of UrbComp'13*, pages 1–8, Chicago, USA.
- [Silva et al. 2014d] Silva, T. H., Vaz de Melo, P. O. S., Almeida, J. M., Salles, J., and Loureiro, A. A. F. (2014d). Revealing the city that we cannot see. *ACM Trans. Internet Technol.*, 14(4):26:1–26:23.
- [Silva et al. 2013e] Silva, T. H., Vaz de Melo, P. O. S., Viana, A., Almeida, J. M., Salles, J., and Loureiro, A. A. F. (2013e). Traffic Condition is more than Colored Lines on a Map: Characterization of Waze Alerts. In *Proc. of SocInfo'13*, pages 309–318, Kyoto, Japan.
- [Sinnott 1984] Sinnott, R. W. (1984). Virtues of the Haversine. *Sky and Telescope*, 68(2):159+.
- [Sørensen 1948] Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, 5(4).
- [Srivastava et al. 2012] Srivastava, M., Abdelzaher, T., and Szymanski, B. (2012). Human-centric sensing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):176–197.
- [Sutton et al. 2009] Sutton, T., Dassau, O., and Sutton, M. (2009). *A gentle introduction to gis*.
- [Thelwall et al. 2010] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558.
- [Toch et al. 2010] Toch, E., Cranshaw, J., Drielsma, P. H., Tsai, J. Y., Kelley, P. G., Springfield, J., Cranor, L., Hong, J., and Sadeh, N. (2010). Empirical models of privacy in location sharing. In *Proc. of Ubicomp'10*, pages 129–138, Copenhagen, Denmark. ACM.
- [Tostes et al. 2013] Tostes, A. I. J., Duarte-Figueiredo, F., Assunção, R., Salles, J., and Loureiro, A. A. F. (2013). From data to knowledge: City-wide traffic flows analysis and prediction using bing maps. In *Proc. of ACM UrbComp'13*, Chicago, USA.
- [Tostes et al. 2014] Tostes, A. I. J., Silva, T. H., Duarte-Figueiredo, F., and Loureiro, A. A. F. (2014). Studying traffic conditions by analyzing foursquare and instagram data. In *Proc. of ACM PE-WASUN'14*, Montreal, Canada.
- [Yan 2009] Yan, X. (2009). *Linear regression analysis: theory and computing*. World Scientific.
- [Youyou et al. 2015] Youyou, W., Kosinski, M., and Stillwell, D. (2015). Computer-based personality judgments are more accurate than those made by humans. *Nat. Acad. of Sci.*, 112(4):1036–1040.
- [Yuan et al. 2013] Yuan, Q., Cong, G., Ma, Z., Sun, A., and Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. In *Proc. of SIGIR '13*, pages 363–372, Dublin, Ireland. ACM.
- [Zaki and Meira Jr 2014] Zaki, M. J. and Meira Jr, W. (2014). *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.
- [Zhang et al. 2013] Zhang, K., Jin, Q., Pelechris, K., and Lappas, T. (2013). On the importance of temporal dynamics in modeling urban activity. In *Proc. of UrbComp'13*, pages 7:1–7:8, Chicago, Illinois.
- [Zheng et al. 2014a] Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014a). Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38.
- [Zheng et al. 2014b] Zheng, Y., Liu, T., Wang, Y., Zhu, Y., Liu, Y., and Chang, E. (2014b). Diagnosing new york city's noises with ubiquitous data. In *Proc. of UbiComp'14*, pages 715–725, Seattle, Washington. ACM.
- [Zheng et al. 2009] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Proc. of WWW'09*, pages 791–800, Madrid, Spain. ACM.

- [Zheng et al. 2012] Zheng, Y.-T., Zha, Z.-J., and Chua, T.-S. (2012). Mining travel patterns from geotagged photos. *ACM Trans. Intell. Syst. Technol.*, 3(3):56:1–56:18.
- [Zipf 1946] Zipf, G. K. (1946). The $p_1 \propto p_2^d$ hypothesis: On the intercity movement of persons. *American sociological review*, pages 677–686.

Capítulo

3

Teoria da Computação: Uma Introdução à Complexidade e à Lógica Computacional

Celina M. H. de Figueiredo e Luís C. Lamb
Universidade Federal do Rio de Janeiro e Universidade Federal do Rio Grande do Sul

Abstract

Computing theory is an abstract and mathematical research area, motivated by the challenges of computer science practice. Perhaps the main objective of theoretical computer science is to explore and understand the nature of computing, contributing to development of efficient and effective methodologies. This course introduces the basics of two fundamental subareas of theoretical computer science: computational complexity and computational logic. Such areas have significantly contributed to the development of computer science, not only from a foundational, but also from an applied perspective.

Resumo

A teoria da computação é uma área de pesquisa abstrata e matemática, e é motivada pelos desafios da prática da computação. O objetivo da teoria da computação é explorar e entender a natureza da computação, de modo a oferecer metodologias eficientes e corretas. Este curso apresenta uma introdução à duas subáreas fundamentais da teoria da computação: complexidade e lógica computacional. Estas duas áreas têm contribuído significativamente para o desenvolvimento da Ciência da Computação, não somente do ponto de vista fundamental, mas também sob uma perspectiva de aplicações.

3.1. Uma Introdução à Complexidade (por Celina M. H. de Figueiredo)

“Resolver ou Verificar?” é uma pergunta que vale um milhão de dólares. No ano 2000, o Instituto Clay para Matemática distinguiu sete problemas considerados centrais para o progresso da matemática, chamando-os de Os Problemas do Milênio. A solução de cada problema corresponde a um prêmio de um milhão de dólares. Um dos sete problemas selecionados é um problema de Teoria da Computação: existe pergunta cuja resposta pode ser verificada rapidamente mas cuja resposta requer muito tempo para ser encontrada?

Esse Problema do Milênio, conhecido como P versus NP, é o problema central na área de Complexidade Computacional, onde tentamos classificar a dificuldade dos problemas de acordo com a eficiência das possíveis soluções através de algoritmos computacionais. Discutiremos o problema P versus NP, através da classificação de problemas desafiadores em polinomial ou NP-completo e na definição de classes de problemas que admitem um resultado de dicotomia, onde cada problema da classe é classificado como polinomial ou NP-completo.

3.1.1. O Problema do Milênio

O computador é um aliado imprescindível para resolver os complexos problemas que surgem em biologia, química, física, economia, áreas nas quais pesquisadores se dedicam à modelagem e à simulação de problemas de larga escala com uso de computadores. Porém, há problemas que têm resistido à habilidade dos programadores. À medida que resolvemos problemas cada vez maiores e mais complexos por meio de enorme poder computacional e algoritmos engenhosos, os problemas resistentes e desafiadores ganham destaque. Nesse sentido, a teoria que propõe o problema do milênio ajuda a entender as limitações computacionais fundamentais. Uma questão de interesse teórico explica a dificuldade prática de problemas formulados em toda a comunidade científica.

No ano 2000, o Instituto Clay para Matemática distinguiu sete problemas considerados centrais para o progresso da matemática, chamando-os de Os Problemas do Milênio. A solução de cada problema corresponde a um prêmio de um milhão de dólares. Um dos sete problemas selecionados é um problema de Teoria da Computação: existe pergunta cuja resposta pode ser verificada rapidamente mas cuja resposta requer muito tempo para ser encontrada? Esse Problema do Milênio, conhecido como P versus NP, é o problema central na área de Complexidade Computacional, onde tentamos classificar a dificuldade dos problemas de acordo com a eficiência das possíveis soluções através de algoritmos computacionais [22].

O problema P versus NP pode ser informalmente resumido na pergunta “Resolver ou Verificar?”, que é ilustrada pela seguinte estória:

Em 1903, o matemático americano Frank Cole provou que o número $2^{67} - 1 = 147573952589676412927$ não é primo exibindo a fatoração $193707721 \times 761838257287$. É simples (embora tedioso se feito manualmente) calcular $2^{67} - 1$, calcular o produto $193707721 \times 761838257287$ e verificar que dão o mesmo número. Já encontrar essa fatoração é difícil. Cole disse que ele levou três anos trabalhando aos domingos.

“Resolver ou Verificar?” é uma pergunta que vale um milhão de dólares.

Duas campanhas: vacinação e pavimentação

Imagine que uma campanha de vacinação no estado do Rio de Janeiro precisa visitar cada uma das capitais dos seus 50 municípios. Por restrições de custo, a equipe responsável, saindo da cidade do Rio de Janeiro, precisa visitar cada uma das outras 49 cidades e retornar à cidade de partida, realizando um circuito que usa as rodovias do estado e que visita

cada capital de município exatamente uma vez. Será que tal circuito pode ser realizado no estado do Rio de Janeiro? Este circuito corresponde ao problema do Ciclo Hamiltoniano, formulado por William Hamilton em 1856, e conhecido também como o problema do Caixeiro Viajante. Uma condição suficiente para que tal circuito do vacinador exista é que cada cidade tenha muitas rodovias do estado que passem por ela mas esta condição não é necessária. Uma condição necessária é a inexistência de cidade gargalo, isto é, uma cidade tal que todas as rotas que conectam outras duas cidades usando rodovias do estado precisam passar por esta cidade gargalo mas esta condição não é suficiente. No estado do Rio de Janeiro, o município de Cabo Frio é um gargalo que isola o município de Búzios e impossibilita um circuito do vacinador. O município de Angra é outro gargalo e isola o município de Paraty.

Imagine agora que, ao invés de um vacinador que saindo da cidade sede da campanha visita num circuito um conjunto de cidades passando por cada cidade visitada uma única vez e retornando à cidade de partida, você considera uma campanha para recuperar a pavimentação das rodovias do estado. Por restrições de custo, a equipe responsável, saindo da cidade sede da campanha, realiza um circuito que percorre cada rodovia exatamente uma vez e retorna à cidade sede de partida. Será que tal circuito pode ser realizado no estado do Rio de Janeiro? Observe que o circuito do vacinador visita cada cidade do estado exatamente uma vez enquanto que o circuito do pavimentador percorre cada rodovia do estado exatamente uma vez. Possivelmente o vacinador deixa de percorrer alguma rodovia do estado enquanto que possivelmente o pavimentador visita uma cidade do estado mais de uma vez. O circuito do vacinador corresponde a uma permutação do conjunto das cidades do estado enquanto que o circuito do pavimentador corresponde a uma permutação do conjunto das rodovias do estado. Ambos problemas buscam a solução, o circuito desejado, num espaço com um número enorme de circuitos viáveis, da ordem de $50!$ circuitos, onde consideramos todas as possíveis permutações. É impraticável resolver o problema através de um algoritmo de força bruta que testa cada uma das permutações candidatas. Buscamos uma propriedade matemática do problema que reduza drasticamente este espaço exponencial de busca. Leonhard Euler resolveu definitivamente o problema do pavimentador ao publicar em 1736 o que consideramos o primeiro artigo científico da área de Teoria dos Grafos. Neste artigo, foi apresentada uma propriedade que caracteriza, uma propriedade que é ao mesmo tempo necessária e suficiente, para a existência de tal circuito do pavimentador: o número de rodovias que passam por cada cidade é par. Euler reduziu drasticamente o espaço exponencial de busca constituído de todas as possíveis permutações das rodovias. A poderosa condição de Euler fornece a solução após a fácil verificação dos graus das cidades.

Em Matemática e em Computação, a Teoria dos Grafos é a área que estuda estruturas matemáticas que modelam relações binárias entre objetos de um conjunto. Um grafo é definido por um conjunto de objetos chamados vértices e um conjunto de arestas que ligam pares de vértices. Euler em 1736 modelou o problema do pavimentador através de um problema em Teoria dos Grafos: cada cidade corresponde a um vértice e cada rodovia entre duas cidades é representada por uma aresta que liga os dois vértices correspondentes. No estado do Rio de Janeiro temos 50 cidades capitais de municípios correspondendo a 50 vértices e temos rodovias que correspondem às arestas conectando estes vértices. A cidade capital de Itaguaí tem 7 rodovias que passam por ela, e corres-

ponde a um vértice de grau ímpar no grafo das cidades capitais e rodovias do estado, o que segundo o Teorema de Euler impossibilita a existência de um circuito do pavimentador no estado. Por outro lado, caso encontremos um outro estado onde todas as cidades tenham grau par, esta condição fácil e poderosa de Euler garante que tal estado admite um circuito do pavimentador.

Para o problema do pavimentador, podemos decidir se o circuito existe ou não através da condição simples encontrada por Euler: basta conferir a paridade de cada cidade. Para o problema do vacinador, temos algumas condições apenas necessárias, temos outras condições apenas suficientes, mas ainda não temos uma condição que caracterize e resolva o problema, e não sabemos se tal condição existe! Para o problema do vacinador, hoje nos resignamos a buscar a resposta examinando o enorme espaço exponencial de busca constituído de todas as possíveis permutações das cidades!

Por outro lado, caso alguém persistente ou com muita sorte apresente um circuito do vacinador, um circuito que visita cada cidade exatamente uma vez, é fácil verificar que este circuito satisfaz a restrição: basta conferir que cada cidade é visitada uma vez e que cidades consecutivas no circuito são de fato conectadas por uma rodovia.

Será que o problema do vacinador é mais difícil que o problema do pavimentador? Será que existem problemas cuja solução pode ser verificada facilmente mas cuja solução não pode ser encontrada facilmente? Este é o desafio central para a Teoria da Computação.

Problemas P, NP e NP-completo

Vejamus novo exemplo. Considere uma turma em um colégio, digamos com 40 alunos, cada qual com certo número de amigos dentro da turma. O problema do Emparelhamento procura dividir aqueles alunos em 20 pares, cada par formado por alunos que sejam amigos. Note que o número de conjuntos contendo 20 pares de alunos quaisquer, não necessariamente amigos, é igual a $40!/(20!2^{20})$, aproximadamente 2^{78} , ou $3,2 \times 10^{23}$, um número de 24 algarismos! Segundo essa estimativa, temos algo como 3 mil vezes mais maneiras distintas de dividir nossa turma em 20 pares de alunos do que grãos de areia no mundo. O problema do emparelhamento é um problema fundamental na área de complexidade computacional. É um problema cuja solução sugeriu a definição formal adotada para computação eficiente. Um algoritmo é eficiente se a sua execução consome um número de passos que cresce como um polinômio no tamanho dos dados da entrada. O problema do emparelhamento admite algoritmo que o resolve em n^3 passos, para uma turma de n alunos. Para n igual a 40, observe a drástica redução do exponencial 2^{78} , muito maior que 2^{40} , para o polinomial 40^3 , que dá um número de passos, ou operações básicas, próximo a 64 mil. Usamos o conceito de polinomial como sinônimo de tratável, viável, eficiente, em contraste com exponencial, sinônimo de inviável.

Considere algumas variações do problema do emparelhamento: o problema de se dividir a turma em grupos contendo 3 amigos mútuos em cada grupo; o problema de se dividir a turma em 3 grupos contendo apenas amigos mútuos em cada grupo; problema de organizar os alunos numa única mesa redonda de forma que apenas amigos sentem-se lado a lado (aqui reencontramos disfarçado o problema do vacinador, na turma de 40 alunos, existem $40!$ alocações possíveis numa mesa redonda); No caso de cada uma das variações

do problema, ainda não conhecemos algoritmo que executa um número polinomial de passos. Cada uma das variações do problema define um problema matemático desafiador, tema de pesquisa avançada na área da Matemática chamada Combinatória. O desafio é encontrar alguma propriedade que corresponda a um atalho polinomial para buscar rapidamente no espaço exponencial das possíveis soluções. Estes problemas desafiadores compartilham uma propriedade: dada uma candidata a solução — um emparelhamento em grupos de três alunos compatíveis, uma alocação numa única mesa redonda, uma partição em três grupos de alunos compatíveis — podemos aprovar a candidata através de um algoritmo.

A questão central para Computação é: quão eficientemente um problema pode ser resolvido através de um algoritmo? Do ponto de vista computacional, distinguimos problemas fáceis e difíceis usando o conceito de algoritmo polinomial. Consideramos fáceis os problemas que podem ser resolvidos através de um algoritmo cujo número de passos cresce com uma potência fixa do número de símbolos usados para especificar a entrada do problema. Por outro lado, consideramos difíceis os problemas para os quais qualquer possível algoritmo consome um número extremamente grande de passos até retornar a resposta. Muitos problemas candidatos a problemas difíceis compartilham a propriedade: encontrar a solução parece difícil mas verificar a solução é fácil. A intuição diz que encontrar a solução tem que ser mais difícil do que simplesmente verificar a solução mas nem sempre a intuição é um bom guia para a verdade. Para estes problemas candidatos a problemas difíceis, não conhecemos ainda algoritmos polinomiais e não sabemos provar a inexistência de algoritmo polinomial. A Teoria da Complexidade Computacional considera estes problemas desafiadores, estes problemas que resistem à classificação em fácil ou difícil, como o problema do vacinador, dentro de uma única classe contendo problemas igualmente difíceis, igualmente desafiadores.

Em Teoria da Computação, chamamos de P a classe dos problemas que podem ser resolvidos através de um algoritmo polinomial, um algoritmo cujo número de passos seja uma potência fixa no tamanho dos dados do problema. E chamamos de NP a classe dos problemas onde toda candidata a solução pode ser aprovada ou rejeitada em tempo polinomial. A condição de verificar parece bem menos restritiva que a de encontrar a solução. A igualdade $P = NP$ significaria que todo problema que tem solução que pode ser verificada rapidamente tem também solução que pode ser encontrada rapidamente. Para estudar a possível igualdade $P = NP$, os pesquisadores definiram o conjunto dos problemas igualmente difíceis entre si, e pelo menos tão difíceis quanto qualquer problema em NP. São os chamados problemas NP-completo. O problema do vacinador é um problema NP-completo. Caso alguém consiga um algoritmo de tempo polinomial para resolver o problema do vacinador, terá na verdade resolvido um problema que vale 1 milhão de dólares porque terá provado que $P = NP$.

Problemas desafiadores em Teoria dos Grafos

Um grafo é *perfeito* quando todo subgrafo induzido admite uma coloração própria de seus vértices (i.e., vértices adjacentes têm cores diferentes) que usa o mesmo número de cores que o número de vértices de uma de suas cliques (um conjunto de vértices mutua-

mente adjacentes). Os *grafos perfeitos* definem um rico tópico de pesquisa que pode ser visto como a interseção de três áreas: Teoria dos Grafos, Complexidade Computacional e Otimização Combinatória.

Claude Berge [5] introduziu a classe dos grafos perfeitos no início dos anos 1960, como uma classe de grafos interessante sob três aspectos, cada um com um problema aberto desafiador: uma classe definida a partir dos problemas de otimização combinatória NP-difíceis COLORAÇÃO MÍNIMA e CLIQUE MÁXIMA; uma classe cujo problema de reconhecimento (i.e., decidir se um dado grafo é perfeito) não se sabia sequer classificar como em NP; uma classe que sugeria uma caracterização por uma família auto-complementar infinita de subgrafos induzidos proibidos. Grötschel, Lovász e Schrijver [24] apresentaram em 1979 um algoritmo polinomial para a coloração mínima dos grafos perfeitos usando o método do elipsóide para programação linear, e deixaram em aberto o problema de encontrar um algoritmo polinomial de natureza puramente combinatória para colorir otimamente os grafos perfeitos. Johnson em 1981 na sua coluna *The NP-completeness – an ongoing guide* [28] propôs o problema de reconhecimento para a classe dos grafos perfeitos. Cameron [8] provou em 1982 que esse problema estava em CoNP. A prova de que a caracterização por subgrafos proibidos proposta por Berge estava correta e o algoritmo polinomial para o reconhecimento dos grafos perfeitos só apareceriam respectivamente em 2006 e em 2005 [12, 11]. O algoritmo polinomial de natureza puramente combinatória para colorir otimamente os grafos perfeitos ainda permanece como um desafio [37]. Descrevemos a seguir como se deu a classificação da complexidade computacional de dois problemas centrais, abertos por décadas, um em polinomial e o outro em NP-completo.

Partição Assimétrica O problema da partição assimétrica foi proposto por Chvátal em 1985 no contexto de decomposições para grafos perfeitos [13]: dado um grafo G , o seu conjunto de vértices admite uma partição em quatro partes não vazias A, B, C, D tal que todo vértice em A é adjacente a todo vértice em B e todo vértice em C é não-adjacente a todo vértice em D ? Chvátal propôs a partição assimétrica como uma generalização de decomposições bem estudadas: corte estrela, corte clique e conjunto homogêneo, e acertou ao afirmar que a partição assimétrica teria um papel determinante na prova da conjectura de Berge para grafos perfeitos. No grafo G , o conjunto de vértices $A \cup B$ é um corte cuja remoção desconecta as partes não vazias C e D , e no complemento do grafo G , o conjunto de vértices $C \cup D$ é um corte cuja remoção desconecta as partes não vazias A e B . Dessa propriedade auto-complementar, vem o nome partição assimétrica, o conjunto $A \cup B$ é chamado *corte assimétrico*, e a partição A, B, C, D é chamada *partição assimétrica*.

O primeiro algoritmo polinomial para testar se um grafo admite uma partição assimétrica, um algoritmo recursivo de complexidade $O(n^{100})$, na verdade resolve um problema mais geral, a partição assimétrica com listas [15]. A versão com listas de um problema de partição tem como entrada, além do grafo cujo conjunto de vértices será particionado, uma lista de partes permitidas, para cada um dos vértices. Na versão com listas, as partes podem ser vazias. Muitos problemas combinatórios em grafos podem ser formulados como um problema de partição dos vértices em subconjuntos de acordo com restrições internas ou externas em relação às adjacências: conjunto homogêneo, corte

clique, coloração de vértices, grafo *split*, entre outros. Feder, Hell, Klein e Motwani descreveram uma dicotomia para partições em quatro partes na versão com listas, onde classificaram cada um dos problemas dessa classe em quasi-polinomial ou NP-completo [14]. Um algoritmo de tempo quasi-polinomial tem complexidade de pior caso $O(n^{c \log t n})$, onde c e t são constantes positivas e n é o número de vértices do grafo de entrada. O problema da partição assimétrica foi classificado nessa ocasião como quasi-polinomial, uma indicação de que o problema era de fato polinomial. Posteriormente, o algoritmo polinomial que desenvolvemos para partição assimétrica estabeleceu a complexidade do problema de Chvátal. O nosso algoritmo recursivo para partição assimétrica com listas recursivamente define problemas de partição assimétrica com listas menores. O número de subproblemas $T(n)$ encontrados satisfaz recorrências da forma $T(n) \leq 4T(9n/10)$, o que fornece um tempo de execução $O(n^{100})$, um desafio para a noção aceita de que solúvel em tempo polinomial é o mesmo que solúvel eficientemente na prática.

Grafo Clique O problema de reconhecimento dos grafos clique foi proposto por Roberts e Spencer em 1971 no contexto de grafos de interseção [39]. Um *conjunto completo* de um grafo $H = (V, E)$ é um subconjunto de V que induz um subgrafo completo. Uma *clique* é um conjunto completo maximal. Um grafo é um *grafo clique* se ele é o grafo de interseção dos conjuntos completos maximais de algum grafo. Foram necessários quase 40 anos para que a prova de NP-completude fosse estabelecida [1]. Não é evidente que o problema de reconhecimento pertença a NP, já que o número de completos maximais em um grafo pode ser exponencial no seu número de vértices. Considere por exemplo o grafo que consiste de um emparelhamento induzido. O grafo contém $2^{n/2}$ conjuntos independentes maximais, e portanto o seu complemento é um grafo com $2^{n/2}$ conjuntos completos maximais. A propriedade de Helly tem sido muito estudada com o objetivo de classificar a complexidade do problema de reconhecimento de grafos clique. Uma família de conjuntos $\mathcal{F} = (F_i)_{i \in I}$ é *intersectante* se a interseção de cada dois de seus membros é não vazia. A *interseção total* de \mathcal{F} é o conjunto $\bigcap \mathcal{F} = \bigcap_{i \in I} F_i$. A família \mathcal{F} tem a *propriedade de Helly*, se qualquer subfamília intersectante tem interseção total não vazia. Roberts e Spencer nesse seu artigo fundamental caracterizaram os grafos clique em termos de uma cobertura das arestas do grafo por conjuntos completos que satisfaz a propriedade de Helly, o que forneceu uma prova de que o problema de reconhecimento dos grafos clique está em NP. Szwarcfiter caracterizou os grafos clique-Helly e apresentou um algoritmo polinomial para o seu reconhecimento [44]. O problema de reconhecimento dos grafos clique foi seguidamente proposto em vários livros [7, 32, 36, 45]. Após a nossa surpreendente prova de NP-completude [1], continuamos aprofundando o estudo do problema de reconhecimento dos grafos clique para a classe dos grafos *split*. A idéia era buscar um problema que separasse em termos de complexidade as classes cordal e *split*. É conhecido que um grafo cordal tem um número linear de conjuntos completos maximais. Embora a classe dos grafos *split* seja precisamente a interseção dos grafos cordais e dos complementos de grafos cordais, são raros os problemas que separam as classes cordal e *split*, i.e., problemas que são NP-completos para cordais mas polinomiais para *split*, segundo o *The NP-completeness – an ongoing guide* de Johnson [29] e o livro de Spinrad [42]. Estabelecemos recentemente que o reconhecimento dos grafos clique *split* é NP-completo [2].

classe de grafos	VERTEXCOL	EDGECOL	MAXCUT
perfect	P	N	N
chordal	P	O	N
split	P	O	N
strongly chordal	P	O	O
comparability	P	N	O
bipartite	P	P	P
permutation	P	O	O
cographs	P	O	P
proper interval	P	O	O
split-proper interval	P	P	P

Tabela 3.1. N: NP-completo, P: polinomial, O: aberto

3.1.2. O Guia para Problemas Desafiadores

Na sua famosa série de colunas sobre NP-completude, Johnson [28] atualiza o seu livro seminal *The NP-completeness – an ongoing guide* [23]. Nos referimos mais precisamente à coluna sobre *graph restrictions and their effect* [29]. O objetivo é identificar problemas interessantes e classes de grafos interessantes estabelecendo o conceito de separação de complexidade. Uma classe C é dita *classe de grafos separadora* quando C separa a complexidade de dois problemas π and σ : π é NP-completo quando consideramos como entrada para π grafos da classe C , mas σ é polinomial quando consideramos como entrada para σ grafos da classe C . Um problema π é dito *problema separador de complexidade* quando π separa duas classes de grafos $\mathcal{A} \subset \mathcal{B}$: π é polinomial quando consideramos como entrada para π grafos da classe \mathcal{A} , mas π é NP-completo quando consideramos como entrada para π grafos da classe \mathcal{B} .

A coluna sobre *graph restrictions and their effect* [29] possui uma seção dedicada aos grafos perfeitos e aos grafos de interseção. A Tabela 3.1 é uma subtabela da tabela apresentada naquela coluna onde selecionamos algumas linhas correspondentes a classes de grafos e algumas colunas correspondentes a problemas, e mantivemos a mesma notação.

É bem conhecido na área de Teoria dos Grafos que os grafos perfeitos constituem uma classe de grafos onde o problema coloração de vértices é polinomial. Por outro lado, o problema coloração de arestas é NP-completo quando restrito à classe dos grafos perfeitos. Na verdade, a subclasse dos grafos de comparabilidade é uma classe de grafos separadora [27] porque separa os problemas coloração de vértices e coloração de arestas. Além disso, coloração de arestas é um problema separador de complexidade já que separa a classe dos grafos de comparabilidade da classe dos grafos bipartidos. É notável que existam poucas classes de grafos onde a complexidade de coloração de arestas é estabelecida, e é surpreendente que, desde 1985, várias classes de grafos para as quais Johnson mencionou na sua coluna como “aparentemente aberto, mas possivelmente fácil de resolver”, permanecem desafiadores, 30 anos depois. A classe dos cografos, uma classe de grafos muito restrita e estruturada definida ao proibirmos o caminho induzido por quatro vértices P_4 , admite apenas uma solução parcial [40]. A classe de grafos split-proper inter-

val merece atenção. Ortiz, Maculan, e Szwarcfiter [34] caracterizaram os grafos que são split e proper interval, e a caracterização permitiu que coloração de arestas [34] e corte máximo [6] sejam polinomiais para esta classe. Observamos que, dois problemas clássicos: coloração de arestas e corte máximo permanecem abertos para a classe dos grafos de intervalo, uma classe de grafos de interseção muito estudada [42]. Observe que existem classes de grafos onde coloração de vértices é NP-completo e coloração de arestas é polinomial, por exemplo, grafos com um vértice universal [35]. Possivelmente mais interessante é a existência de classes de grafos onde coloração de arestas é NP-completo, mas coloração total — onde colorimos todos os elementos do grafo, vértices e arestas — é polinomial [30].

split versus cordal O fato de que as classes split e cordal concordam em relação à complexidade dos três problemas: coloração de vértices, coloração de arestas e corte máximo, sugere um padrão que tem sido investigado na literatura.

Um grafo é cordal, quando todo ciclo com pelo menos quatro vértices admite uma corda, e um grafo é split quando o seu conjunto de vértices pode ser particionado em um conjunto independente e uma clique. Os grafos split constituem uma subclasse dos grafos cordais muito estruturada, já que são exatamente os grafos tais que tanto o grafo quanto o seu complemento são ambos cordais.

Johnson, na sua coluna sobre *graph restrictions and their effect* [29], declarou que não conhecia problema que separasse as classes split e cordal do aspecto de complexidade. Spinrad, em seu livro [42] vinte anos depois, ao relatar os resultados de complexidade para classes de grafos, atualiza os problemas separadores de complexidade em relação às classes split e cordal: clique máxima e coloração de vértices são polinomiais para ambos; enquanto conjunto dominante, corte máximo e ciclo hamiltoniano são NP-completos para ambos. Existem poucos problemas separadores de complexidade em relação às classes split e cordal: empacotamento de triângulos e pathwidth, para os quais o problema é NP-completo para cordal mas polinomial para split. Spinrad [42] informa que os grafos split estão no núcleo dos algoritmos e dos resultados de dificuldade para os grafos cordais. Problemas separadores de complexidade em relação às classes split e cordal são raros. Recentemente, tentamos o problema grafo clique, mas obtivemos que o problema permanece NP-completo para grafos split [1, 2]. Possivelmente, a classe dos grafos planares fornecerá uma classe onde o problema grafo clique seja polinomial [3]. Outro problema que continua aberto é coloração de arestas, e os resultados parciais para as classes cordal, intervalo e split sugerem que o problema coloração de arestas é desafiador mesmo restrito a classes muito estruturadas [16, 34].

grafos sem ciclos com corda única Trotignon e Vušković [46] estudaram a classe C dos grafos sem ciclos com corda única. A motivação principal para investigar a classe foi encontrar um teorema que revelasse a estrutura desses grafos, um tipo de resultado que não é muito frequente na literatura. Dada uma estrutura proibida, por exemplo, ciclo com corda única, procuramos investigar que consequência traz para a estrutura do grafo, e que problemas desafiadores tornam-se polinomiais. O teorema é do tipo: todo grafo na classe C pode ser obtido a partir de grafos básicos de C aplicando uma série de operações

que “identificam” grafos de C . Outra propriedade investigada é a de família de grafos χ -limitada, introduzida por Gyárfás [25], como uma generalização natural da propriedade que define os grafos perfeitos. Uma família de grafos \mathcal{G} é χ -limitada com função χ -limitada f se, para todo subgrafo induzido G' de $G \in \mathcal{G}$, temos $\chi(G') \leq f(\omega(G'))$, onde $\chi(G')$ é o número cromático de G' e $\omega(G')$ denota o tamanho da clique máxima em G' . Os grafos perfeitos satisfazem: para todo subgrafo induzido G' de G na classe dos grafos perfeitos, temos $\chi(G') = \omega(G')$, a função χ -limitada f é a função identidade. Por outro lado, o teorema dos grafos perfeitos estabelece que um grafo G é perfeito se e somente se G e o complemento de G não admitem um ciclo ímpar sem cordas e com pelo menos 5 vértices. A pesquisa nesta área é dedicada principalmente a entender para que escolhas de subgrafos induzidos proibidos, a família resultante de grafos é χ -limitada; veja [38] para um *survey* sobre o assunto. Pelo Teorema de Vizing para coloração de arestas, a classe dos grafos de linha de grafos simples é uma classe de grafos χ -limitada com função χ -limitada $f(x) = x + 1$ (este limite superior especial é chamado o *limite de Vizing*) e, além disso, os grafos de linha são caracterizados por nove subgrafos induzidos proibidos [47]. A classe C dos grafos sem ciclos com corda única é também χ -limitada com o limite de Vizing [46]. Em [46] os seguintes resultados são obtidos para os grafos na classe C : um algoritmo $O(nm)$ para a coloração dos vértices, um algoritmo $O(n + m)$ para clique máxima, um algoritmo $O(nm)$ para o reconhecimento da classe, e a prova de NP-completude para conjunto independente máximo.

Consideramos a complexidade do problema de coloração de arestas para a classe C [31]. O problema de coloração de arestas, também conhecido como índice cromático, é o problema de determinar o menor número $\chi'(G)$ de cores necessárias para colorir as arestas do grafo G . Também investigamos subclasses obtidas da classe C quando proibimos o ciclo sem cordas com 4 vértices e quando proibimos o ciclo sem cordas com 6 vértices. As Tabelas 3.2 e 3.3 resumem os resultados obtidos, mostrando que classes com muita estrutura ainda definem problemas difíceis e desafiadores. Nas tabelas, denotamos por C a classe dos grafos sem ciclo com corda única e por Δ o grau máximo no grafo.

classe de grafos	$\Delta = 3$	$\Delta \geq 4$	regular
grafos em C	NP-completo	NP-completo	NP-completo
grafos em C sem C_4	NP-completo	polinomial	polinomial
grafos em C sem C_6	NP-completo	NP-completo	NP-completo
grafos em C sem C_4 , sem C_6	polinomial	polinomial	polinomial

Tabela 3.2. Dicotomia de complexidade NP-completo versus polinomial para a coloração de arestas dos grafos sem ciclo com corda única.

A classe inicialmente investigada em [31] foi a classe C dos grafos sem ciclo com corda única. Para os objetivos do trabalho, um grafo G é *básico* se G é completo, um ciclo sem cordas com pelo menos 5 vértices, um grafo fortemente 2-bipartido, ou um subgrafo induzido do grafo de Petersen ou do grafo de Heawood; e G não possui um vértice de articulação, um corte próprio com 2 vértices ou uma junção própria. Após provar que coloração de arestas é NP-completo para grafos em C , consideramos os grafos na subclasse $C' \subset C$ que não têm C_4 . Ao proibir um ciclo sem cordas com 4 vértices C_4 , evitamos a decomposição por junção própria, uma decomposição difícil para coloração

de arestas [4, 40, 41], o que significa que cada grafo da classe C' que não é básico pode ser decomposto através de um vértice de articulação ou através de um corte próprio com 2 vértices. Para a classe C' , estabelecemos então uma dicotomia: coloração de arestas é NP-completo para grafos de C' com grau máximo 3 e é polinomial para grafos de C' com grau máximo *não* 3. Adicionalmente, determinamos uma condição necessária para um grafo $G \in C'$ com grau máximo 3 ser Classe 2, isto é, ter índice cromático $\chi'(G) = \Delta(G) + 1 = 4$. A condição necessária é ter um grafo P^* — subgrafo do grafo de Petersen — como um bloco básico na árvore de decomposição. Consequentemente, se proibimos ambos C_4 e C_6 , o índice cromático dos grafos sem ciclo com corda única pode ser determinado em tempo polinomial. Os resultados alcançados em [31] podem ser relacionados com várias áreas da pesquisa em coloração de arestas, como observamos a seguir.

A primeira observação diz respeito ao resultado de dicotomia encontrado para a classe class C' . A dicotomia é muito interessante porque esta é a primeira classe de grafos para a qual coloração de arestas é NP-completo para grafos com um dado grau máximo fixo Δ mas é polinomial para grafos com grau máximo $\Delta' > \Delta$. Além disso, a classe C' é a primeira classe de grafos interessante onde coloração de arestas é NP-completo em geral, mas é polinomial quando restrito a grafos regulares. É interessante observar que a classe de grafos C' é uma classe de grafos com poucos grafos regulares — somente o grafo de Petersen, o grafo de Heawood, os grafos completos e os ciclos sem cordas.

A segunda observação diz respeito à conjectura de Chetwynd e Hilton. Uma ferramenta importante para identificar classes de grafos que são Classe 2 é o conceito de *sobrecarregado* [17]. Um grafo é dito *sobrecarregado* se satisfaz $|E| > \Delta(G)\lfloor |V|/2 \rfloor$, uma condição suficiente para que $\chi'(G) = \Delta(G) + 1$. Por exemplo, o ciclo sem cordas com 5 vértices é um grafo sobrecarregado. Uma condição suficiente mais geral é: G é *subgrafo sobrecarregado* se G possui o mesmo grau máximo que o grafo e G é sobrecarregado. Grafos que são subgrafo sobrecarregados são Classe 2 [17] e podemos verificar em tempo polinomial se um grafo é subgrafo sobrecarregado [33]. Para algumas classes de grafos, ser subgrafo sobrecarregado é equivalente a ser Classe 2. Exemplos de tais classes são: grafos com vértice universal [35], grafos multipartidos completos [26], e grafos split com grau máximo ímpar [9]. A conjectura de Chetwynd e Hilton [10] diz que um grafo $G = (V, E)$ com $\Delta(G) > |V|/3$ é Classe 2 se e somente se é subgrafo sobrecarregado. De fato, para a maioria das classes para as quais o problema de coloração de arestas é resolvido em tempo polinomial, a equivalência “Class 2 = Subgrafo Sobrecarregado” é válida. É notável que a maioria destas classes é composta de grafos cujo grau máximo é alto — sempre superior a um terço do número de vértices. Portanto, para estas classes, a equivalência “Class 2 = Subgrafo Sobrecarregado” — e o consequente algoritmo polinomial para o problema de coloração de arestas — seriam consequência direta da Conjectura de Chetwynd e Hilton para grafos subgrafo sobrecarregados, caso a conjectura seja válida. Neste sentido, a classe C' investigada em [31, 46] apresenta um grande interesse: para grafos em C' não há limite na relação entre “número de vértices versus grau máximo”; mesmo assim, se o grau máximo não é 3, a equivalência “Class 2 = Subgrafo Sobrecarregado” é válida. Portanto, a classe dos grafos em C' com grau máximo não 3 é uma classe de grafos que não encaixa nas hipóteses da Conjectura de Chetwynd e Hilton para grafos subgrafo sobrecarregados, mas para a qual coloração de arestas é ainda solúvel em tempo polinomial através da equivalência “Class 2 = Subgrafo Sobrecarregado”.

classe de grafos	$k \leq 2$	$k \geq 3$
grafos k -partidos	polinomial	NP-completo

Tabela 3.3. Dicotomia de complexidade NP-completo versus polinomial para a coloração de arestas de grafos multipartidos.

A terceira observação é relacionada com o estudo dos *snarks* [43]. Um *snark* é um grafo cúbico sem pontes com índice cromático igual a 4. Para evitar casos triviais, snarks são em geral restritos a grafos com cintura pelo menos 5 e não contêm 3 arestas cuja remoção resulta em um grafo desconexo, com cada componente não trivial. O estudo dos snarks é relacionado fortemente ao famoso e histórico Teorema das Quatro Cores, para a coloração de mapas. Através de um resultado de [31], o único snark não trivial que *não* possui ciclo com corda única é o grafo de Petersen.

A quarta observação diz respeito ao problema de determinar o índice cromático de um grafo k -partido, isto é, um grafo cujo conjunto de vértices pode ser particionado em k conjuntos independentes. O problema de determinar o índice cromático de um grafo k -partido, já estava classificado, é polinomial para $k = 2$ [28, 29], e para grafos multipartido completos [26]. A prova de NP-completude do índice cromático dos grafos na classe C implica que coloração de arestas é NP-completo para grafos k -partidos que são r -regulares, para cada $k \geq 3$, $r \geq 3$ [31].

3.1.3. Notas Bibliográficas

Indicamos algumas publicações da autora para o leitor interessado. Para um público amplo, a autora possui um artigo de divulgação [18] na revista Ciência Hoje: “Resolver ou Verificar? Uma pergunta que vale um milhão de dólares”. Para o público especialista na área de Teoria dos Grafos e Complexidade Computacional, a autora reuniu as suas contribuições principais e apontou problemas em aberto relacionados no artigo *survey* [19]: “The P vs. NP-complete dichotomy of some challenging problems in graph theory”. A autora tem dois textos introdutórios anteriores na Jornada de Atualização em Informática (JAI), um texto sobre Emparelhamento em Grafos [20] e um texto sobre Coloração em Grafos [21].

No JAI 1999, em co-autoria com Jayme Szwarcfiter, escreveu sobre “Emparelhamento em Grafos — Algoritmos e Complexidade”. Encontrar um emparelhamento máximo em um grafo é um problema clássico no estudo de algoritmos, com muitas aplicações: designação de tarefas, determinação de rotas de veículos, problema do carteiro chinês, determinação de percursos mínimos, e muitos outros. O artigo histórico de Edmonds que descreveu um algoritmo $O(n^4)$ para o problema geral de emparelhamento, na verdade, introduziu a noção de algoritmo de tempo polinomial. Implementações mais eficientes do algoritmo de Edmonds foram apresentadas por vários pesquisadores como Lawler (algoritmo $O(n^3)$), Hopcroft e Karp (algoritmo $O(m\sqrt{n})$ para o caso bipartido), Micali e Vazirani (algoritmo $O(m\sqrt{n})$ para o caso geral, o mais eficiente até o momento). O texto apresenta uma introdução ao problema de emparelhamento em grafos, e um estudo de seus algoritmos e complexidade. Apresentamos algoritmos eficientes para os quatro problemas relacionados com encontrar um emparelhamento de cardinalidade máxima ou de peso máximo, em grafos bipartidos ou em grafos quaisquer. Estes quatro

problemas são todos casos particulares do problema de emparelhamento de peso máximo em grafos quaisquer, mas é interessante considerá-los em ordem crescente de dificuldade por razões de ordem didática [20].

No JAI 1997, em co-autoria com João Meidanis e Célia Mello, escreveu sobre “Coloração em Grafos”. O texto é uma introdução à coloração em grafos. Consideramos dois tipos de problemas de coloração em grafos: coloração de vértices e coloração de arestas. Aplicamos a estes problemas técnicas clássicas de coloração tais como: algoritmos gulosos, decomposição e alteração estrutural. Consideramos classes de grafos para as quais tais técnicas resolvem estes problemas de coloração eficientemente. O estudo de coloração, um tópico básico em Teoria dos Grafos, surgiu a partir do conhecido “problema das quatro cores”. Em 1852, Francis Guthrie questionou se todo grafo planar poderia ser colorido com quatro cores. Embora estivesse claro que quatro cores eram necessárias, a questão se referia ao número mínimo de cores, i.e., quantas cores são suficientes para colorir qualquer grafo planar. Esta questão foi respondida afirmativamente por Appel e Haken, usando computador, após mais de 100 anos. Robertson, Sanders, Seymour e Thomas mais recentemente confirmaram esta resposta fornecendo uma prova mais simples, mais aceita pela comunidade matemática, embora ainda use o computador. O problema das quatro cores corresponde à coloração dos vértices de um grafo. Mais precisamente, corresponde à coloração mínima dos vértices de um grafo. Outros problemas de coloração existem, tais como coloração das arestas, coloração total (vértices e arestas), coloração a partir de conjuntos de cores atribuídos previamente aos vértices ou às arestas de um grafo. O escopo do texto, no entanto, é mais restrito: coloração mínima de vértices e coloração mínima de arestas em classes de grafos [21].

3.2. Lógica Para Computação: Uma Introdução Curta (por Luís C. Lamb)

O entendimento de sistemas lógicos é fundamental em Ciência da Computação. A própria noção de computabilidade é originária de estudos em lógica. Diversas lógicas têm sido utilizadas e desenvolvidas em Computação desde o estabelecimento desta como ciência organizada, particularmente a partir da segunda metade do Século XX. O uso de métodos lógicos tem sido fundamentais em múltiplas subáreas da computação. Estas áreas incluem: especificação e verificação de sistemas computacionais, projeto de sistemas de bancos de dados, projeto lógico de circuitos integrados, complexidade descritiva, web semântica, inteligência artificial e sistemas distribuídos. Este capítulo apresenta uma breve introdução à lógica computacional. Apresentaremos, também, noções históricas sobre a evolução da lógica em Ciência da Computação, ilustrando com algumas aplicações em áreas relevantes.

3.2.1. Introdução: Brevíssimas Considerações Históricas

It takes an extraordinary intelligence to contemplate the obvious.
Alfred North Whitehead

Lógica é uma área organizada do domínio do conhecimento humano. É uma área de estudo caracterizada por conceitos e definições precisas. O estudo das técnicas e métodos de lógica(s) permite a melhor definição, entendimento e raciocínio sobre diversos domínios

do conhecimento.¹

Na Grécia antiga (“clássica”), os *Estóicos*² estudaram e definiram noções iniciais de lógica proposicional e, preliminarmente, de inferência. O nome de Aristóteles, fundamental na história da ciência define os silogismos (padrões de raciocínio em que premissas tem uma conclusão lógica) e o raciocínio com quantificadores (séc. IV A.C). Exemplo simples de silogismo: *Todos os lógicos são filósofos; Luís é lógico; Portanto, Luís é filósofo*; silogismos foram utilizados até a idade média, servindo, inclusive para a formalização de textos escolásticos. No período medieval, alguns lógicos se destacam, como Pierre Abélard (França), no Século XIII, e William of Ockham (Inglaterra) no Século XIV. Ockham é referenciado até hoje pelo conceito da “navalha de Ockham”, mas também introduziu princípios similares ao que conhecemos hoje como Leis de *De Morgan* em linguagem natural, além de lógicas tri-valoradas. Abélard publica o *Logica ingredientibus*, em 1121 (lógica para principiantes) e destaca-se como grande propagador do sistema de Aristóteles. Ambos fazem uso do sistema silogístico para verificação de argumentos. No mesmo período, Ramon Llull (*latim*: Lulio), no Séc. XIII, desenvolve estudos para verificar automaticamente a verdade de silogismos, desenvolvendo um sistema de raciocínio como computação. É importante ressaltar que o trabalho de Llull, de certa forma, influenciou o trabalho de Leibniz.

Durante um longo período, o conhecimento acadêmico foi pouco desenvolvido ou relatado e reportado, algumas vezes justificado pelo período da idade média no mundo ocidental. Na idade média, o ensino da lógica era parte do *trivium*: conjunto de disciplinas básicas das artes liberais: gramática, lógica e retórica. O *trivium* era base para o *quadrivium*, conjunto de disciplinas formado pela aritmética, geometria, música e astronomia. Neste período, há desenvolvimentos relevantes no oriente médio e vizinhança, particularmente na Pérsia, sub-continentes asiático (Índia) e Oriente.³ Outras contribuições significativas são de Gottfried Willhem Leibniz (1646-1716). Leibniz, grande nome da matemática, idealizou uma linguagem universal a *lingua characteristica universalis* para expressar todo conhecimento humano. Desenvolveu também o *Calculus ratiocinator*: modelo universal teórico de cálculo lógico - executado por máquinas - para derivar relações lógicas. Esta contribuição é relevante devido ao objetivo computacional. Tanto que Leibniz é referido como o “Patron saint of Computer Science” por Moshe Vardi, na *Communications of the ACM*, no Editorial de Dezembro de 2011.

¹O objetivo deste texto não é oferecer uma história completa da lógica. As considerações históricas são apresentadas, apenas, para estimular o leitor a buscar informações adicionais sobre a evolução da lógica e seu impacto na ciência da computação. O texto também *não oferece* um curso completo de lógica; apenas introduz, de forma não exaustiva, conceitos e noções básicas e - em reduzida síntese - sua relação com a ciência da computação. O leitor é convidado a consultar as referências bibliográficas citadas no texto.

²O pensamento estóico tem como pioneiro Zeno (ou Zenão) de Cítio (344-262 AC). O pensamento estóico foi predominante na filosofia ocidental até o crescimento da doutrina cristã. Para os estóicos, a lógica inclui a análise de argumentos, a retórica, gramática, teoria de conceitos, percepções, proposições, pensamentos, o que - de certa forma - poderia ser visto como epistemologia ou filosofia da linguagem. No entanto, não há espaço suficiente neste capítulo para maiores detalhes sobre o estoicismo. Uma referência na moderna “Stanford Encyclopedia of Philosophy” [48] é uma boa bibliografia inicial sobre o assunto.

³Para melhor entendimento dos estudos em lógica desenvolvidos neste período histórico recomendamos a consulta do *Handbook of Philosophical Logic e Handbook of the History of Logic*. Estas importantes obras modernas de organização do conhecimento sobre lógica pura e aplicada publicaram capítulos específicos sobre o desenvolvimento da lógica nestas regiões e nestes períodos históricos [67, 69].



Figura 3.1. Aristoteles, G. W. Leibniz

No século XIX há grandes desenvolvimentos em lógica matemática. George Boole (1815-1864) nome de forte impacto em matemática (posteriormente em ciência da computação, economia e engenharia) desenvolve a lógica proposicional, sob rigorosa formalização. Historiadores da ciência afirmam que Boole, de fato, introduz a lógica simbólica em *An Investigation of the Laws of Thought* (1854). No mesmo século, em torno de 1879, Gottlob Frege (1848-1925), no *Begriffsschrift*, formaliza a Lógica de Primeira Ordem, introduzindo definições precisas da quantificação existencial e universal em lógica matemática. Augustus De Morgan (1806-1871) formaliza a lógica das relações e define *indução matemática* de forma rigorosa - técnica hoje muito utilizada em ciência da computação - além de formalizar as leis que levam seu nome. Além desses, Charles Sanders Peirce (1839-1914) define também o raciocínio abduutivo, propõe uma definição rigorosa de indução matemática e princípios de lógica relacional, estendendo trabalho de De Morgan. Ademais, mostra que a utilização do operador *nor* permite representar todas as funções booleanas. No final do século XIX, Bertrand Russell mostra que o sistema de Frege é inconsistente. Isto leva Russell ao desenvolvimento da teoria de tipos em lógica matemática (que terá certa influência e inspiração sobre a teoria de tipos em ciência da computação, particularmente na área de linguagens de programação).



Figura 3.2. Grandes Lógicos do Século XIX: Boole, De Morgan, Frege, Peirce

Entre 1903 e 1910, Alfred North Whitehead (1861-1947) e Bertrand Russell (1872-1970) escrevem o monumental *Principia Mathematica* [99], em 3 volumes, com mais de 2 mil páginas. O objetivo era formalizar na linguagem da lógica, utilizando axiomas e regras de inferência, todo o conhecimento matemático. A partir deste período, a lógica matemática e a teoria de conjuntos estabelecem-se como fundamentos da matemática moderna.⁴

⁴O *Principia* ou PM, como também é conhecido, foi considerado um dos 100 livros de não-ficção mais importantes do Século XX pelo New York Times (<http://www.nytimes.com/library/books/042999best-nonfiction-list.html>). Whitehead e Russell foram filósofos de grande influência no Século XX. Russell foi considerado por muitos especialistas o “filósofo do século” [89]. Whitehead foi considerado genial pela famosa intelectual da primeira metade do século, Gertrude Stein, que nas primeiras frases da “Autobiografia

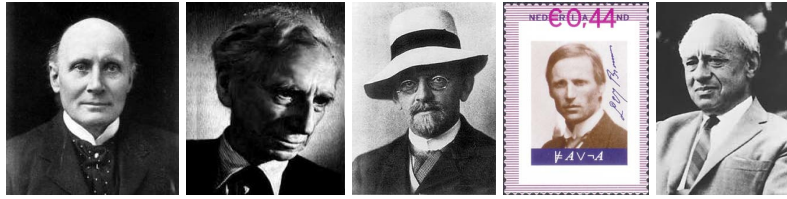


Figura 3.3. Alfred Whitehead, Bertrand Russell, David Hilbert, L.E.J. Brouwer, Alfred Tarski

Luitzen Egbertus Jan Brouwer (1881-1966), desenvolve a lógica intuicionista [86, 96]. A lógica intuicionista (ou construtiva) rejeita a lei do meio excluído em seu sistema, a eliminação da dupla negação $\neg\neg\alpha \neq \alpha$, assim como provas por contradição. Assim, as provas nesta lógica são construtivas, o que remete à relação com conceitos de computação. O isomorfismo de Curry-Howard, simplificada, relaciona provas construtivas (intuicionistas) e programas. A lógica intuicionista tem sido pesquisada como um sistema lógico que pode ser fundamental em diversas áreas da ciência da computação, incluindo na teoria de tipos de linguagens de programação.

Na década de 1920, David Hilbert (1862-1943), um dos matemáticos mais influentes do início do século XX, propôs a busca por uma nova fundamentação para a matemática. Sinteticamente, Hilbert queria mostrar (1) a completude da matemática: todas as verdades podem ser provadas. (2) a consistência da matemática: uma afirmação matemática e sua contradição não podem ser ambas provadas através de um sistema lógico. (3) a decidibilidade: existe uma forma mecânica de mostrar se uma afirmação matemática arbitrária qualquer é verdadeira ou falsa. Através de um procedimento formal, Hilbert queria mostrar que era possível construir todas as verdades matemáticas. Deste programa deriva-se o *Entscheidungsproblem* (ou “problema da decisão”) - demonstrar a (in)existência de um algoritmo para determinar se um enunciado da lógica de primeira ordem pode ser provado. Em 1922, Ludwig Wittgenstein publica o *Tractatus Logico-Philosophicus* [100]. O *Tractatus* é uma obra influente na filosofia analítica e estabeleceu o seu autor como um dos filósofos mais importantes do século XX. Nesta obra, Wittgenstein define, entre diversas conceituações notáveis, a noção de tabelas-verdade, na forma em que passam a ser utilizadas até hoje.

Na década de 1930 Kurt Gödel, Alan Turing e Alonzo Church estudam as limitações da lógica de primeira ordem.⁵ Entre as contribuições fundamentais de Gödel, citamos o seu primeiro teorema da incompletude: um sistema axiomático suficientemente poderoso para descrever a aritmética dos naturais, não é completo e o segundo teorema de Gödel: em um sistema axiomático suficientemente poderoso para descrever a aritmética dos naturais, a consistência destes axiomas não pode ser provada pelo próprio sistema. Tais resultados responderam negativamente ao programa de Hilbert, mostrando não ser possível definir um sistema axiomático correto e completo para toda matemática. É inte-

de Alice B. Toklas” (na verdade sua própria autobiografia, uma obra vanguardista na cultura ocidental) afirmou “... that only three times in my life have I met a genius ... the three geniuses of whom I wish to speak are Gertrude Stein, Pablo Picasso and Alfred Whitehead.”

⁵A contribuição de Kurt Gödel foi inclusive popularizada pelos grandes meios de comunicação da imprensa. Em 1999, Gödel foi escolhido o matemático do século pela revista *Time Magazine*.

ressante observar que estudos em decidibilidade contribuem decisivamente para o desenvolvimento da noção de computabilidade. Neste mesmo período histórico e até os anos 1950, a definição precisa de consequência lógica, a noção de verdade em uma estrutura e trabalhos fundamentais na teoria de modelos são desenvolvidos por Alfred Tarski.

Além disso, Gödel desenvolve um sistema completo e correto para a lógica de primeira ordem. Church encontra uma solução negativa ao *Entscheidungsproblem* [52]: é impossível decidir algoritmicamente se afirmações da aritmética são verdadeiras ou falsas; Church define a noção de *calculabilidade* e também introduz o cálculo- λ para estudo das funções referidas por ele como calculáveis - mas referidas por Turing como *computáveis*. Posteriormente, o cálculo- λ também passa a ter forte influência na ciência da computação, principalmente na área de linguagens de programação, a partir da década de 1960. De grande influência na comunidade de lógica, Church orientou as teses de nomes relevantes em lógica e ciência da computação como: A.M. Turing, S. Kleene, J.B. Rosser, D.S. Scott, M.O. Rabin, H. Rogers Jr., R. Smullyan e Leon Henkin.

Atualmente, em poucos textos a referência ao termo *calculabilidade* é encontrada. No famoso artigo de Turing [93], ele faz referência pioneira ao termo “computable” enquanto esclarece que o conceito de “calculable” (calculável) de Church ao estudar funções calculáveis é equivalente à sua definição de computável [93]:

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In §8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel [1]. These results have valuable applications. In particular, it is shown (§11) that the Hilbertian Entscheidungsproblem can have no solution. In a recent paper Alonzo Church [2] has introduced an idea of “effective calculability”, which is equivalent to my “computability”, but is very differently defined. Church also reaches similar conclusions about the Entscheidungsproblem [3]. The proof of equivalence between “computability” and “effective calculability” is outlined in an appendix to the present paper.

[1] Über formal unentscheidbare Satze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik*, 38(1931):173-198.

[2] An unsolvable problem of elementary number theory, *Amer. J. Math.*, 58(1936): 345-363.

[3] A note on the Entscheidungsproblem. *J. of Symbolic Logic*, 1(1936): 40-41.

Ademais, em [93] as noções de algoritmo, procedimento efetivo e computabilidade são explicitadas. A partir das contribuições de Turing, Church, Gödel e muitos outros, a Ciência da Computação passa a ser desenvolvida sob rigor científico e, fundamentalmente, lógico-matemático. As contribuições de Church e Turing levam à formulação da Tese de Church-Turing: uma função sobre números inteiros é computável se, e somente se, ela é computável em uma Máquina de Turing. Outra ideia de Turing - o programa armazenado em fita tem grande impacto, inclusive tecnológico. Este princípio muito simples influenciou von Neumann no desenvolvimento do conceito de programa armazenado em memória.

A partir do final da segunda guerra mundial, o desenvolvimento da Ciência da

Computação acontece de forma muito acelerada. O próprio desenvolvimento do transistor, no final da década de 1940, a construção dos primeiros computadores e o desenvolvimento do modelo de arquitetura de computadores conhecido como modelo de von Neumann, levam a impressionantes impactos sociais e econômicos. Ainda nesta época, Turing novamente causa impacto, ao levantar questões relacionados ao raciocínio e aprendizado de máquinas, i.e., pavimentando o caminho para o surgimento da Inteligência Artificial. Evidentemente, para formalizar raciocínio torna-se necessário o desenvolvimento de novos sistemas lógicos, “implementáveis” em computadores. A partir da década de 1960 novas lógicas são desenvolvidas em ciência da computação, tendo em vista a evolução da ciência e a necessidade de especificar, verificar e raciocinar sobre programas, sistemas e construções computacionais. Em suma, o impacto da lógica em computação foi tão significativo que muitos pesquisadores se referem à lógica como o “cálculo da computação” [72, 83].

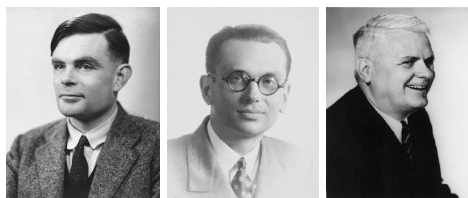


Figura 3.4. Alan Turing, FRS (1912-54), Kurt Gödel (1906-78), Alonzo Church (1903-95)

3.2.2. Lógica em Computação: Uma Breve Introdução

A seguir, apresentamos uma introdução muito básica à lógica proposicional. Há excelentes livros que recomendamos para o leitor, com extenso material sobre o assunto, inclusive sobre aspectos relacionados à Ciência da Computação. Um dos melhores textos em língua portuguesa é o livro de Corrêa da Silva, Finger e Melo [54]. Neste livro, são abordados em detalhe temas como SAT (satisfatibilidade), prova automática de teoremas e verificação de programas. Este capítulo tem como objetivo, apenas, oferecer uma breve introdução à lógica em computação. Na literatura há outras obras de grande abrangência e qualidade sobre lógica em computação, e.g. [50, 66, 76].

3.2.3. Lógica Proposicional

Lembre-se que para falar sobre algum domínio, é necessário estabelecer (isto é, definir) uma *linguagem*. Em sistemas lógicos, as noções de sintaxe, semântica, e teoria de prova (sistemas de prova) são fundamentais. A teoria de prova se refere à forma sintática de obter ou identificar as afirmações válidas da linguagem.

Também dizemos que a sintaxe se refere às regras utilizadas para construir as fórmulas de um sistema lógico. Por sua vez, a semântica se refere ao significado destas fórmulas bem-formadas (formadas de acordo com regras definidas de sintaxe), mapeando as sentenças a uma interpretação. Em ciência da computação, estes conceitos também são essenciais, tanto do ponto de vista fundamental, quanto do ponto de vista de aplicações, como em linguagens de programação. Assim, o conhecimento básico de lógica clássica torna-se útil em um grande número de aplicações. Tipicamente, a lógica clássica inclui

e tem como objeto de estudo a lógica proposicional e a lógica de predicados de primeira ordem. Iniciaremos, desta forma, nossa abordagem através de breves introduções à lógica proposicional e à lógica de predicados.

Em lógica proposicional as afirmações atômicas são *proposições*: fatos que podem ser verdadeiros/falsos em um determinado mundo possível/situação. Para representar variáveis proposicionais, por exemplo, podemos usar p, q, r, \dots ; para representar os conectivos (ditos “Booleanos”) tipicamente utilizamos $\neg, \vee, \wedge, \rightarrow$ (ou \supset)⁶. Estes símbolos de conectivos representam negação (conectivo unário), disjunção (“ou” lógico), conjunção (“e” lógico) e implicação material (“se então”, informalmente). Exemplos de formulae podem ser construídos a partir destes símbolos da linguagem proposicional: $\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$; $(p \rightarrow q) \rightarrow (\neg p \vee q)$; $(p \rightarrow q) \rightarrow p$.

Note que a lógica proposicional tem expressividade limitada. Por exemplo, a afirmação *todos os homens são mortais*, é expresso como um átomo proposicional, abstraindo-se relações e quantificações. Os conectivos “Booleanos” têm uma interpretação similar à utilizada em linguagem natural. Assim, $p \wedge q$ intuitivamente significa que p e q são interpretados como verdadeiros em uma determinada situação. Lógicas são utilizadas para aplicações, por exemplo, em representação do conhecimento (humano). Portanto, encontrar e estabelecer correspondência entre lógica(s) e linguagem natural é essencial nestas aplicações. Esta “tradução” nem sempre é simples. Lembre que a lógica clássica nos permite representar somente afirmações. A idéia fundamental na tradução é tentar extrair proposições atômicas de uma afirmação e combiná-las através de conectivos. Por exemplo, “Maria gosta de João e de Pedro” pode ser “traduzida” como “Maria gosta de João \wedge Maria gosta de Pedro” e formalizada como $(p \wedge q)$. Por sua vez, a afirmação (proposição) “Spinoza não gosta de Tomás” pode ser representada como $\neg(\text{Spinoza gosta de Tomás})$ e formalizada simplesmente como $\neg p$. Note que esta formalização proposicional abstrai relações entre objetos/pessoas das afirmações em linguagem natural. Finalmente, um exemplo envolvendo implicação material (ou um “condicional”): “Se o Internacional vence, então todos estamos felizes” pode ser traduzida como “Inter vencedor \rightarrow felizes” e formalizada $p \supset q$ ou $p \rightarrow q$. No entanto, como a linguagem natural é ambígua, encontrar uma tradução aceita universalmente nem sempre é possível: por exemplo, “Pedro será rico e famoso” não pode ser apenas traduzido como “Pedro será rico \wedge Pedro será famoso”, pois Pedro poderá ficar famoso somente após perder sua fortuna em um cassino, o que possivelmente seria melhor formalizado com uma *lógica temporal*. Algumas fórmulas são relativamente simples: Por exemplo, “*está chovendo \rightarrow Candide não sai de casa*” pode ser traduzido como “se está chovendo então Candide não sai de casa”. Entretanto, *não* devemos ler a fórmula $p \rightarrow q$ como “ p causa q ”, pois a relação de causa e efeito tem interpretação distinta da implicação material da lógica clássica.

⁶A sintaxe para o condicional $p \rightarrow q$ é hoje a mais comumente utilizada. Na notação do Principia Mathematica [99] a implicação material seria representada como $p \supset q$. No entanto, deve-se observar que existem diversos condicionais em lógica(s). O condicional da lógica clássica é conhecido como “implicação material”, sendo que $p \rightarrow q$ é logicamente equivalente a $\neg p \vee q$. No entanto, esta equivalência não é válida em outros sistemas lógicos não-clássicos, por exemplo em lógica intuicionista, na qual esta equivalência não é um teorema. Há diversas outras formalizações de condicionais: contrafatuais, condicionais estritos, causais e outros. No entanto, o escopo deste capítulo é restrito, e não exploraremos estes “condicionais não-clássicos” [51].

É claro que nem sempre existe uma correspondência “trivial” entre aquilo que é expresso através de uma linguagem natural e através de uma linguagem simbólica de uma lógica. Em computação, a fim de expressarmos as afirmações e o conhecimento sobre um determinado domínio, idealmente imaginamos que não existam ambiguidades nas interpretações de fórmulas. Na lógica clássica, trabalhamos com valores-verdade de fórmulas em situações nas quais determina-se se os componentes (átomos) de fórmulas são verdadeiras ou não. Se pensarmos num programa de computador, avaliaríamos um teste condicional (um “if”) em função dos valores das variáveis do programa. Para átomos p, q, r, \dots uma situação especificará os valores-verdade dos mesmos, i.e. se eles são verdadeiros ou falsos nesta situação. Note que os valores-verdade dos átomos podem ser diferentes em diferentes situações. O valor-verdade de uma fórmula proposicional em uma situação é definida como a seguir. É conveniente incluir \top e \perp como as fórmulas que são sempre verdadeiras e sempre falsas, respectivamente. Assim, \top e \perp denotam, também, tautologias e contradições, respectivamente. \top é verdadeiro e \perp é falso; $\neg p$ verdadeiro se p é falso, e falso se p é verdadeiro. $p \wedge q$ é verdadeiro se p, q são ambos verdadeiros; caso contrário, $p \wedge q$ é falso. $p \vee q$ é verdadeiro se um ou mais dentre p, q são verdadeiros, falso em caso contrário. $p \rightarrow q$ é verdadeiro se p é falso ou q é verdadeiro (ou ambos). Caso contrário, a implicação é falsa.⁷ As noções de *verdade*, *validade* e *equivalência* são essenciais em lógica. Uma fórmula proposicional é logicamente válida se ela é verdadeira em qualquer situação. Escreve-se $\models p$ se p é válida, para uma fórmula arbitrária p . Fórmulas proposicionais válidas são também conhecidas como *tautologias*.

Por sua vez, uma fórmula p é uma *contradição* se ela for falsa em todas interpretações. Uma fórmula proposicional é *satisfável* (ou ainda *consistente*) se ela é verdadeira em pelo menos uma situação. Duas fórmulas p, q são logicamente equivalentes se elas são verdadeiras em exatamente nas mesmas situações. Podemos escrever $p \equiv q$ neste caso. Outra noção relevante é a de consistência de um conjunto de fórmulas. Um conjunto de fórmulas bem formadas $\{A_1, A_2, \dots, A_n\}$ é consistente se existe uma interpretação de sua conjunção que é verdadeira; i.e. se valores podem ser atribuídos a todas as sentenças atômicas no conjunto de fórmulas $\{A_1, A_2, \dots, A_n\}$ que fazem com que as fórmulas sejam verdadeiras. Por exemplo, o conjunto $\{p \rightarrow q, \neg p, \neg q\}$ é consistente, enquanto que o conjunto $\{r \rightarrow s, r, \neg s\}$ não é consistente.

Dizemos que uma fórmula q é *consequência lógica* de um conjunto de fórmulas p_1, p_2, \dots, p_n , representado por $p_1, p_2, \dots, p_n \models q$ se em toda situação (interpretação) na qual p_1, p_2, \dots, p_n são todas fórmulas verdadeiras, q também é verdadeira. Note que esta é uma noção de consequência *semântica*. Podemos, também definir consequência lógica como uma noção *sintática*; neste caso, $p_1, p_2, \dots, p_n \vdash q$ significa que existe um *prova* de q a partir das fórmulas p_1, p_2, \dots, p_n , utilizando um sistema de prova (e.g. sistema de dedução natural, tableaux, etc). Uma *prova* é uma sequência (finita) de fórmulas bem-formadas que são deduzidas a partir de outras fórmulas através do uso de regras de um sistema de provas ou de axiomas. Resultados anteriormente já provados (teoremas) podem ser utilizados como *lemas* em provas. Note que um sistema de provas utiliza regras puramente

⁷Podemos expressar o significado definido acima através de *tabelas-verdade*. Ludwig Wittgenstein, no *Tractatus Logico-Philosophicus* [100] define a noção de tabelas-verdade, na forma em que passaram a ser utilizadas até hoje. Cada linha de uma tabela-verdade corresponderá a uma *interpretação* da fórmula para a qual construímos a tabela.

sintáticas. Podemos dizer que q pode ser *provado* a partir das premissas p_1, p_2, \dots, p_n . Assim, as noções de consequência lógica no nível sintático ($p_1, p_2, \dots, p_n \vdash q$) e semântico ($p_1, p_2, \dots, p_n \models q$) são claramente distintas. O conjunto de fórmulas (premissas) p_1, p_2, \dots, p_n é também chamado de teoria. A construção sintática $p_1, p_2, \dots, p_n \vdash q$ é denominada, também, de *sequente*. As fórmulas à esquerda de \vdash são chamadas de antecedente, enquanto que a fórmula à direita de \vdash é chamada de consequente. Por sua vez, *teoremas* são fórmulas que seguem logicamente de outras fórmulas válidas; são afirmações deduzidas a partir de outras afirmações (e que são, obviamente, válidas) em um cálculo formal. Formalmente, no entanto, um teorema é a última linha de uma prova. A notação $\vdash \alpha$ denota que α é um teorema. Existem diversos métodos para verificação de validade de fórmulas, incluindo: (1) tabelas-verdade, onde simplesmente verificamos mecanicamente todas as situações relevantes; têm crescimento exponencial no número de subfórmulas atômicas e são aplicáveis somente à lógica proposicional; (2) argumentação direta (utilizado por matemáticos/lógicos), que é relativamente rápida, desde que o usuário seja familiarizado com o método; (3) Sistemas de prova: sistemas de Hilbert (axiomáticos), tableaux, dedução natural e métodos automáticos (provedores automáticos de teoremas - “automated theorem provers”).

Propriedades importantes de sistemas lógicos (ditas metapropriedades) incluem a *correção* e a *completude*. Se todas as fórmulas válidas de um sistema lógico são provadas por um sistema de provas para este sistema, ele é dito *completo* em relação ao modelo semântico; enquanto que a *correção* garante que as provas obtidas por este sistema de prova efetivamente são válidas em relação ao modelo semântico. Por exemplo, para a lógica proposicional, existem tanto sistemas de prova axiomáticos, quanto sistemas de dedução natural que são *corretos e completos*.

3.2.3.1. Exemplo de Sistema de Provas: Sistema Axiomático (à la Frege-Hilbert)

Existem diversos sistemas de prova para lógica proposicional e de primeira ordem. Entre as diversas formas de apresentação de sistemas lógicos, sistemas axiomáticos estão entre as mais utilizadas. Esta apresentação também é conhecida como apresentação *à la* Frege-Hilbert, ou, no “estilo de Hilbert” [84]. O sistema axiomático é historicamente muito utilizado por filósofos e matemáticos. Tem como limitação (en. “drawback”) o fato de levar à construção de provas complexas, mesmo para experientes. Um sistema axiomático consiste de fórmulas que são definidas como axiomas e pela especificação de suas regras de inferência. Para exemplificarmos a sua utilização, apresentamos a seguinte axiomatização da lógica proposicional. Se α, β e γ são quaisquer fórmulas bem-formadas de L , então os seguintes são axiomas de L [84]; esta axiomatização utiliza apenas os conectivos \rightarrow, \neg .

(A1). $(\alpha \rightarrow (\beta \rightarrow \alpha))$

(A2). $((\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)))$

(A3). $((\neg\beta \rightarrow \neg\alpha) \rightarrow ((\neg\beta \rightarrow \alpha) \rightarrow \beta))$.

Uma regra de inferência de L : *modus ponens*: $\alpha, \alpha \rightarrow \beta \vdash \beta$

Exemplo simples: prova em L de $A \rightarrow A$.

1. $(A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$ (Instância Ax. A2)

2. $A \rightarrow ((A \rightarrow A) \rightarrow A)$ Esquema Ax. A1
3. $(A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$ 1, 2, MP
4. $A \rightarrow (A \rightarrow A)$ Esquema Ax. A1
5. $A \rightarrow A$ A partir de 3, 4 por MP

Há mais de uma forma de axiomatizar a lógica proposicional. O sistema axiomático proposto por Hilbert para a lógica proposicional inclui a regra de inferência modus ponens e os seguintes axiomas:

1. $A \rightarrow (B \rightarrow A)$
2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
3. $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$
4. $\neg\neg A \rightarrow A$
5. $A \rightarrow \neg\neg A$

Embora a apresentação axiomática seja sintética, a construção de provas neste sistema é notadamente mais difícil do que em outros sistemas de prova.

3.2.3.2. Exemplo de Sistema de Provas: Sistema de Dedução Natural

O método de dedução natural, foi proposto por Gentzen [70], e refinado por Fitch [63] em um formato onde provas e subprovas são representadas hierarquicamente. O método consiste em uma representação formal da argumentação direta, em que utilizamos regras do sistema para manipular premissas e obter conclusões. A formalização do raciocínio é feita através de regras ditas de *introdução* e de *eliminação*. Regras de *introdução* permitem introduzir em uma linha de prova uma fórmula que utilizam um determinado conetivo que está sendo “introduzido”. Regras de *eliminação* permitem a inferência de uma nova fórmula em que um conetivo foi “eliminado” (estas noções são detalhadas a seguir). Nossa representação utilizará o formato de “boxes” (caixas) que lembram a hierarquia da estrutura original no formato de árvores como Gentzen utilizava. Os livros [50, 51, 66, 76] utilizam os formatos de caixas, que - de certa forma, ressalte-se - lembram também a estrutura de um programa de computador. O método de dedução natural é muito utilizado como formalização do raciocínio utilizado em provas por especialistas; i.e. existe uma correspondência próxima com a forma de como teoremas são provados (na “prática” lógico-matemática) e como são construídas as relações entre premissas e conclusões em dedução natural. Em vez de axiomas, há regras de inferência, que descrevem o que pode ser derivado a partir de premissas e hipóteses.

Em síntese, o sistema de dedução natural tem uma série de vantagens em relação a outros sistemas, como em relação ao sistema axiomático, pois: (i) concentra-se na noção de dedução, que é a noção fundamental em lógica; (ii) evita que tenhamos de definir um

conjunto de fórmulas como axiomas; (iii) freqüentemente as regras de dedução natural são mais fáceis de serem manipuladas. (iv) de certa forma, ele espelha a forma como os seres humanos realizam inferências.

As Regras de Dedução do Sistema de Dedução Natural

A implicação material e uso do raciocínio hipotético: Nas ciências exatas o raciocínio hipotético é fundamental. Existem inúmeras publicações em filosofia da ciência sobre o tema, e neste curso apresentamos algumas formalizações elementares sobre o mesmo. Lembre-se que lógica é fundamentalmente baseada sobre o raciocínio condicional (“se... então...”) e este padrão de raciocínio constitui a espinha dorsal das derivações (algumas não triviais) obtidas através de raciocínios hipotéticos. O método de dedução natural ilustra muito bem a formalização metódica deste tipo de raciocínio, fundamental na ciência. Nas regras que manipulam a implicação material “ \rightarrow ”, a negação “ \neg ”, a disjunção “ \vee ” e nas provas por contradição, a utilização de hipóteses é essencial. Hipóteses são fórmulas utilizadas de uma forma distinta das demais fórmulas. Usa-se suposições (através de raciocínios em situações *hipotéticas*) para construir uma situação na qual esta fórmula é válida. Ou seja, *assumimos* que em uma determinada situação uma hipótese (fórmula) é válida a fim de obter alguma conclusão a partir desta hipótese (fórmula). Por que usamos hipóteses? De forma simplificada, para derivar informações adicionais (em subprovas) sobre a prova maior que estamos tentando construir.

Em dedução natural, uma das **regras para o condicional** \rightarrow faz uso de hipóteses. Assim, a *regra de introdução da implicação material* ($\rightarrow I$) é explicada a seguir: Para introduzirmos (i.e. mostrarmos) uma fórmula $\alpha \rightarrow \beta$ em uma linha de prova, assume-se α e então constrói-se a prova de β . Pode-se utilizar α e as demais fórmulas anteriormente derivadas nesta subprova. No entanto, ao obter β , a hipótese α **não** pode ser utilizada novamente, pois esta era uma hipótese adicional para provar β . Esta prova é isolada em uma caixa para indicar que a hipótese é válida apenas “localmente”. Note que hipóteses são anotadas entre colchetes “[α]”. *Eliminação da implicação material:* Para “eliminar” a implicação material em $\alpha \rightarrow \beta$ é necessário também provar α , e assim podemos derivar β . Dizemos “eliminação” por estarmos deduzindo uma fórmula β (que é o conseqüente de uma implicação $\alpha \rightarrow \beta$). Esta regra também é conhecida como *modus ponens* e é usualmente abreviada em livros de lógica como MP. A seguir, apresentamos as regras de introdução e eliminação da implicação material (representadas por $\rightarrow I$ e $\rightarrow E$, respectivamente).

1. [α] hipótese 2. \vdots 3. β	1. $\alpha \rightarrow \beta$ 2. α 3. β	$\rightarrow E(1,2)$
4. $\alpha \rightarrow \beta \rightarrow I(1,3)$		

Regras para a conjunção \wedge : *Regra de Introdução:* Para introduzir (ou para escrever) uma fórmula do tipo $\alpha \wedge \beta$ em uma prova, devemos provar α e provar β . *Regra de Eliminação:* Se provamos $\alpha \wedge \beta$ então podemos escrever α e/ou podemos também escrever β . As regras podem ser representadas como a seguir.

- | | |
|--|-------------------------------|
| 1. α | 1. $\alpha \wedge \beta$ |
| 2. \vdots | 2. $\alpha \quad \wedge E(1)$ |
| 3. β | 3. $\beta \quad \wedge E(1)$ |
| 4. $\alpha \wedge \beta \quad \wedge I(1,3)$ | |

Regras para a disjunção \vee : *Regra de Introdução:* Para provar $\alpha \vee \beta$, basta provar α ou basta provar β . (A rigor, há duas regras de introdução do \vee : uma a partir de α e outra para eliminação de β). Note que α, β são fórmulas quaisquer da linguagem proposicional. *Regra de Eliminação:* Para provar algo a partir de $\alpha \vee \beta$, deve-se prová-lo a partir da hipótese que α é válida e a partir da hipótese que β é válida. Esta construção é uma argumentação por casos.

- | | | | | | | | |
|---|---|------------------------|-----------------------|---------------------------------------|---------------------------------------|-------------|-------------|
| 1. α
2. $\alpha \vee \beta \quad \vee I(1)$ | 1. $\alpha \vee \beta$
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 2px;">2. $[\alpha]$ hipótese</td> <td style="width: 50%; padding: 2px;">5. $[\beta]$ hipótese</td> </tr> <tr> <td style="padding: 2px;">3. \vdots \langleprova\rangle</td> <td style="padding: 2px;">6. \vdots \langleprova\rangle</td> </tr> <tr> <td style="padding: 2px;">4. γ</td> <td style="padding: 2px;">7. γ</td> </tr> </table> 8. $\gamma \quad \vee E(1, 2-4, 5-7)$ | 2. $[\alpha]$ hipótese | 5. $[\beta]$ hipótese | 3. \vdots \langle prova \rangle | 6. \vdots \langle prova \rangle | 4. γ | 7. γ |
| 2. $[\alpha]$ hipótese | 5. $[\beta]$ hipótese | | | | | | |
| 3. \vdots \langle prova \rangle | 6. \vdots \langle prova \rangle | | | | | | |
| 4. γ | 7. γ | | | | | | |

Regras para (\neg) : *Introdução:* Para mostrar $\neg\alpha$, assumimos α e provamos \perp . A hipótese não pode ser utilizada fora da subprova. Note que nesta regra, usa-se também o raciocínio hipotético, e interpreta-se $\neg\alpha$ como $\alpha \rightarrow \perp$. *Eliminação:* A partir de α e de $\neg\alpha$, provamos \perp .

- | | | | | |
|--|------------------------|-------------|------------|---|
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">1. $[\alpha]$ hipótese</td> </tr> <tr> <td style="padding: 2px;">2. \vdots</td> </tr> <tr> <td style="padding: 2px;">3. \perp</td> </tr> </table> 4. $\neg\alpha \quad \neg I(1,3)$ | 1. $[\alpha]$ hipótese | 2. \vdots | 3. \perp | 1. α
2. \vdots
3. $\neg\alpha$
4. $\perp \quad \neg E(1,3)$ |
| 1. $[\alpha]$ hipótese | | | | |
| 2. \vdots | | | | |
| 3. \perp | | | | |

Entre as regras derivadas do sistema de dedução natural, podemos citar a regra de eliminação da dupla negação $\neg\neg E$: A partir de $\neg\neg\alpha$ provamos α . Esta regra é utilizada para simplificar passos de prova, mas pode ser derivada de outras regras do sistema de dedução natural. A seguir mostramos que, em lógica clássica, *qualquer fórmula pode ser obtida a partir de uma prova de \perp (também referido como falsum)*. Isto é, mostramos que $\perp \vdash A$ para qualquer fórmula A .

- | | |
|-----------------------------------|--|
| 1. \perp Premissa | |
| 2. $[\neg A]$ hipót. | |
| 3. $\perp \quad \checkmark (1)$ | |
| 4. $\neg\neg A \quad \neg I(2,3)$ | |
| 5. $A \quad \neg\neg E(4)$ | |

Ou seja, ao assumirmos uma situação na qual \perp é verdadeiro (linha 1), se $\neg A$ é verdadeiro (linha 2), então \perp é verdadeiro, o que não é possível. Portanto, temos $\neg\neg A$ (linha 4) e consequentemente, A é verdadeira nesta situação. Isto é: “qualquer coisa” (fórmula) segue de uma contradição. Na prova acima, anotamos uma linha com o símbolo “ \checkmark ”. Esta

anotação simplesmente justifica a utilização de uma fórmula já provada anteriormente (esta notação é utilizada por alguns autores, incluindo [50]).

Note que é impossível que \perp seja verdadeira. \perp só é provado através de hipóteses contraditórias, e isto acontece no meio de uma prova maior. Em uma subprova, podemos então deduzir aquilo que necessitamos/desejamos. Claramente, mostrar contradições, i.e. mostrar \perp em uma sub-prova, pode ser útil em construção de provas.

Prova por Contradição⁸ Para provar α , assume-se $\neg\alpha$ e prova-se \perp . Novamente, esta é uma regra que utiliza raciocínio condicional (hipotético).

1. $[\neg\alpha]$ hipótese 2. \vdots 3. \perp 4. α PC(1,3)
--

Regras para \perp : *Introdução:* Para provar \perp temos de provar α e provar $\neg\alpha$, para fórmulas quaisquer. Esta regra é análoga à regra $\neg E$. *Eliminação:* A partir de uma prova de \perp podemos derivar qualquer fórmula.

1. $\neg\alpha$	
2. \vdots	1. \perp
3. α	2. α $\perp E(1)$
4. \perp $\perp I(1,3)$	

Existem também regras para \top , de introdução e eliminação. \top pode ser introduzido em qualquer ponto de uma prova, pois é uma fórmula sempre válida, enquanto que se temos \top nada podemos afirmar a partir desta fórmula.

Utilização de Lemas: Lemas são fórmulas provadas (válidas), e assim denominadas quando utilizadas na construção de uma prova maior. Por exemplo, um lema muito utilizado em dedução natural para lógica clássica é a lei do meio excluído, i.e. $\alpha \vee \neg\alpha$. Este lema - em geral - deve ser utilizado em conjunto com uma eliminação de \vee , onde $\neg\alpha$ e α serão hipóteses para um raciocínio por casos.

Regras para \leftrightarrow : Lembre-se que $\alpha \leftrightarrow \beta$ é equivalente a $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$, e também é equivalente a $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. *Introdução:* Para provar $\alpha \leftrightarrow \beta$, devemos provar α e β ou devemos provar $\neg\alpha$ e $\neg\beta$. Devido à equivalência acima, também podemos derivar $\alpha \leftrightarrow \beta$ ao provarmos $(\alpha \rightarrow \beta)$ e $(\beta \rightarrow \alpha)$. *Eliminação:* A partir de $\alpha \leftrightarrow \beta$, podemos provar $(\alpha \rightarrow \beta)$ e $(\beta \rightarrow \alpha)$ e $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$. Assim como as demais regras para os conetivos, as regras para \leftrightarrow também estão representadas na Figura 3.6.

1. α	1. $\alpha \leftrightarrow \beta$
2. \vdots	2. $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$ $\leftrightarrow E(1)$
3. β	3. $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ $\leftrightarrow E(1)$
4. $\alpha \leftrightarrow \beta$ $\leftrightarrow I(1,3)$	

A seguir, apresentamos a prova de um sequente simples, representada na Figura 3.5. Na

⁸Também conhecida como redução ao absurdo (Reductio ad absurdum - RAA).

Figura 3.5. Exemplo: Prova do seqüente $p \rightarrow (q \rightarrow r) \vdash p \wedge q \rightarrow r$ através do sistema de dedução natural, ilustrada nesta figura.

1.	$p \rightarrow (q \rightarrow r)$ <i>premissa</i>
2.	$[p \wedge q]$ <i>hip.</i>
3.	p $\wedge E(2)$
4.	q $\wedge E(2)$
5.	$q \rightarrow r$ $\rightarrow E(1,3)$
6.	r $\rightarrow E(4,5)$
7.	$p \wedge q \rightarrow r$ $\rightarrow I(1,6)$

prova do seqüente $p \rightarrow (q \rightarrow r) \vdash p \wedge q \rightarrow r$, escrevemos a premissa $p \rightarrow (q \rightarrow r)$ na linha 1 da Figura 3.5. O nosso objetivo é provar $p \wedge q \rightarrow r$ (que está na linha 7). A conclusão a ser provada é uma implicação material, $\alpha \rightarrow \beta$. Para provar uma implicação material, assumimos como hipótese o antecedente (α) da implicação: portanto, na linha 2 assumimos $[p \wedge q]$ com o objetivo de mostrarmos r . Note que r ocorre como conseqüente de uma implicação, na linha 1. Assim, será necessário usar a regra de eliminação do condicional $\rightarrow E$ (modus ponens) para mostrarmos r . Isto é feito sucessivamente: primeiramente mostramos p , na linha 3 e posteriormente mostramos q na linha 4. Ao mostrarmos estes antecedentes das implicações da linha 1, finalmente podemos mostrar r , a partir de duas aplicações de modus ponens (linhas 5 e 6). Note que a prova da Figura 3.5 não é desenvolvida apenas de “cima para baixo”. Uma analogia muitíssimo simples sobre como construir uma prova nos remete à construção de caminhos entre dois pontos em um mapa. Ou seja, a construção de uma prova muitas vezes é análoga à construção de um caminho: temos de saber o ponto de saída (premissas, hipóteses) e o ponto de chegada (a conclusão, i.e. a fórmula que temos de provar). A partir destes pontos construímos o “caminho” (a prova) entre estes “pontos”, usando este raciocínio, recursivamente, até que o caminho (prova) esteja concluído.

As regras aqui apresentadas permitem que todas as fórmulas válidas da lógica clássica proposicional sejam provadas usando o sistema de dedução natural. Esta propriedade é conhecida como *completude* de um sistema de provas. Um sistema lógico é completo se, e somente se, todas as fórmulas bem formadas válidas são teoremas do sistema. Idealmente, um sistema de provas teria que permitir que todas as fórmulas válidas sejam deriváveis através das regras do sistema. Por outro lado, é desejável também que as fórmulas provadas através do sistema de prova sejam efetivamente fórmulas válidas do sistema lógico em questão. Esta propriedade é conhecida como *correção* de um sistema de provas. O sistema de dedução natural apresentado aqui é correto e completo em relação à semântica apresentada. Isto garante que o sistema de dedução natural pode ser

utilizado para verificar a validade de fórmulas proposicionais clássicas. A seção acima resume regras básicas do sistema de dedução natural para a lógica clássica proposicional. Estas regras seriam diferentes para outros sistemas lógicos. Na lógica intuicionista, por exemplo, não se admitem provas por contradição, nem eliminação da dupla negação $\neg\neg E$ e não é aceita a lei (ou o princípio) do meio excluído $\alpha \vee \neg\alpha$, considerado válido na lógica clássica.

Figura 3.6. Resumo das Regras para os Conetivos Clássicos. As linhas da tabela servem como referência. Obviamente poderíamos adotar uma numeração genérica para as linhas de prova; mas como exemplo didático, a numeração a seguir é clara.

<ol style="list-style-type: none"> 1. α 2. \vdots 3. β 4. $\alpha \wedge \beta \quad \wedge I(1,3)$ 	<ol style="list-style-type: none"> 1. $\alpha \wedge \beta$ 2. $\alpha \quad \wedge E(1)$ 3. $\beta \quad \wedge E(1)$ 			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"> <ol style="list-style-type: none"> 1. $[\alpha]$ hipótese 2. \vdots 3. β </td> <td style="padding: 2px;"> <ol style="list-style-type: none"> 1. $\alpha \rightarrow \beta$ 2. α 3. $\beta \quad \rightarrow E(1,2)$ </td> </tr> </table>	<ol style="list-style-type: none"> 1. $[\alpha]$ hipótese 2. \vdots 3. β 	<ol style="list-style-type: none"> 1. $\alpha \rightarrow \beta$ 2. α 3. $\beta \quad \rightarrow E(1,2)$ 	<ol style="list-style-type: none"> 4. $\alpha \rightarrow \beta \quad \rightarrow I(1,3)$ 	
<ol style="list-style-type: none"> 1. $[\alpha]$ hipótese 2. \vdots 3. β 	<ol style="list-style-type: none"> 1. $\alpha \rightarrow \beta$ 2. α 3. $\beta \quad \rightarrow E(1,2)$ 			
<ol style="list-style-type: none"> 1. α 2. $\alpha \vee \beta \quad \vee I(1)$ 	<ol style="list-style-type: none"> 1. $\alpha \vee \beta$ 2. $[\alpha]$ hipótese 3. \vdots <prova> 4. γ 8. $\gamma \vee E(1,2-4,5-7)$ 			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"> <ol style="list-style-type: none"> 1. $[\alpha]$ hipótese 2. \vdots 3. \perp </td> <td style="padding: 2px;"> <ol style="list-style-type: none"> 1. α 2. \vdots 3. $\neg\alpha$ </td> </tr> </table>	<ol style="list-style-type: none"> 1. $[\alpha]$ hipótese 2. \vdots 3. \perp 	<ol style="list-style-type: none"> 1. α 2. \vdots 3. $\neg\alpha$ 	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"> <ol style="list-style-type: none"> 5. $[\beta]$ hipótese 6. \vdots <prova> 7. γ </td> </tr> </table>	<ol style="list-style-type: none"> 5. $[\beta]$ hipótese 6. \vdots <prova> 7. γ
<ol style="list-style-type: none"> 1. $[\alpha]$ hipótese 2. \vdots 3. \perp 	<ol style="list-style-type: none"> 1. α 2. \vdots 3. $\neg\alpha$ 			
<ol style="list-style-type: none"> 5. $[\beta]$ hipótese 6. \vdots <prova> 7. γ 				
<ol style="list-style-type: none"> 4. $\neg\alpha \quad \neg I(1,3)$ 	<ol style="list-style-type: none"> 4. $\perp \quad \neg E(1,3)$ 			
<ol style="list-style-type: none"> 1. $\neg\alpha$ 2. \vdots 3. α 4. $\perp \quad \perp I(1,3)$ 	<ol style="list-style-type: none"> 1. \perp 2. $\alpha \quad \perp E(1)$ 			
<ol style="list-style-type: none"> 1. $\neg\neg\alpha$ 2. $\alpha \quad \neg\neg E(1)$ 	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"> <ol style="list-style-type: none"> 1. $[\neg\alpha]$ hipótese 2. \vdots 3. \perp </td> </tr> </table>	<ol style="list-style-type: none"> 1. $[\neg\alpha]$ hipótese 2. \vdots 3. \perp 		
<ol style="list-style-type: none"> 1. $[\neg\alpha]$ hipótese 2. \vdots 3. \perp 				
<ol style="list-style-type: none"> 4. $\alpha \quad PC(1,3)$ 	<ol style="list-style-type: none"> 4. $\alpha \quad PC(1,3)$ 			
<ol style="list-style-type: none"> 1. α 2. \vdots 3. β 4. $\alpha \leftrightarrow \beta \quad \leftrightarrow I(1,3)$ 	<ol style="list-style-type: none"> 1. $\alpha \leftrightarrow \beta$ 2. $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta) \quad \leftrightarrow E(1)$ 3. $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha) \quad \leftrightarrow E(1)$ 			

3.2.4. O Problema da Satisfatibilidade (SAT)

O problema da satisfatibilidade - SAT - consiste em determinar se existe uma interpretação que satisfaça uma fórmula da lógica proposicional. Dada uma fórmula α existe uma atribuição v tal que $v(\alpha) = 1$ (ou *true*)? Embora o enunciado seja “simples” e extensivamente estudado, apresenta um dos grandes desafios da Ciência da Computação. Até hoje, não há algoritmo em tempo polinomial para o problema. Steven A. Cook⁹ identificou

⁹Cook recebeu o ACM A.M. Turing Award em 1982 pelo seu trabalho pioneiro em complexidade computacional. Este prêmio, o mais importante da Ciência da Computação, tem esta denominação em reconhecimento às contribuições de Alan M. Turing.

a relevância de SAT em um clássico artigo, publicado no início da década de 1970 [53]. O artigo é intitulado *The Complexity of Theory Proving Procedures*. O próprio título já indica a forte relevância que as questões de lógica tem e sua relação com a área de complexidade computacional. Neste artigo, Cook formalizou as noções de redução em tempo polinomial e mostrou que SAT é um problema NP-completo. Neste artigo, também foi formulada famosa questão *P* versus *NP*, problema que ainda está em aberto.



Figura 3.7. Edgar (Ted) F. Codd, Stephen A. Cook

3.2.5. Lógica de Primeira Ordem

A lógica proposicional, em termos de poder de expressão, é mais “fraca” que a lógica de predicados de primeira ordem. Por exemplo, a afirmação (proposição) “um herói é alguém admirado por todos” é representada simplesmente como uma proposição em lógica proposicional, digamos p ; note que, por exemplo, o predicado “admirado” e o quantificador “todos” são abstraídos na representação. A lógica de predicados aumenta o poder de expressão da lógica proposicional ao adicionar a possibilidade de expressarmos relações entre objetos, inclusive fazendo uso de quantificadores: \forall - que representa a quantificação universal e \exists - que representa a quantificação existencial, além de outras extensões à linguagem. Símbolos de relação n -ários, símbolos de função, constantes e quantificadores sobre variáveis são introduzidos. Na sintaxe, define-se a noção de *assinatura*, que consiste que um conjunto de constantes, símbolos de função e símbolos de relação. A noção de *termo* é fundamental para nomear objetos (utilizamos constantes, variáveis, símbolos de função n -ários como termos). Exemplos de fórmulas atômicas na lógica de primeira ordem, construídas com símbolos de relação unária *homem* e de relação binária *amigo*, constantes (*Pedro*, *Celina*, *Luis*), variáveis (x, y) e símbolo de função unária f : $homem(Pedro)$, $amigo(Celina, Luis)$, $\forall x \exists y (y = f(x))$. Note que constantes e símbolos de função se referem a indivíduos, a objetos, enquanto que os símbolos de relação especificam, representam relações entre estes objetos. Esta especificação de relações entre objetos não é representada de forma precisa na lógica proposicional. Ademais, a semântica da lógica de predicados é mais complexa. Neste curso, apresentamos apenas uma noção muito simplificada desta semântica, por questões de espaço.

Para que possamos interpretar esta linguagem mais expressiva, a semântica deve considerar quantificadores, variáveis e novas fórmulas atômicas, que envolvem relações. A noção de *estrutura* ou *modelo* torna-se fundamental. Uma estrutura identifica uma coleção de objetos em seu domínio, bem como o significado dos símbolos da linguagem. A interpretação de uma constante corresponde a uma constante no domínio da estrutura (ou do modelo); a interpretação de um símbolo de relação em uma estrutura corresponde a uma relação sobre os objetos do domínio desta estrutura. Uma *sentença* é uma fórmula

sem ocorrência de variáveis livres, e.g. $\forall x \exists y (y = f(x))$; as variáveis deste exemplo estão sob o escopo dos quantificadores \forall e \exists , são ditas amarradas (ou limitadas, ou ligadas) ou ainda *bound* - em inglês. Por sua vez a fórmula $\forall x (P(x, y) \rightarrow R(x, y))$ não é uma sentença, pois a variável y é livre: y não ocorre no escopo de um quantificador \forall ou \exists . Variáveis livres são interpretadas através de uma atribuição de valor para um objeto do domínio da estrutura. De forma muito sucinta, fórmulas quantificadas existencialmente são interpretadas em uma estrutura de forma que alguma atribuição torne a fórmula verdadeira, enquanto que para fórmulas quantificadas universalmente todas as atribuições tornam a fórmula verdadeira.¹⁰

Portanto, a lógica de primeira ordem permite especificar propriedades de estruturas como (grafos, ordens parciais, grupos, bancos de dados) usando predicados (relações). As sentenças atômicas da lógica de predicados têm argumentos (*termos*) que denotam objetos e símbolos de predicados de n argumentos. A utilização de variáveis quantificadas pelo quantificador existencial ($\exists x$) e universal ($\forall x$) permite grande expressividade sobre conjuntos de objetos. Por exemplo, podemos expressar propriedades sobre grafos, $G = (V, E), E \subseteq V^2$, como *Cada nodo do grafo tem pelo menos dois vizinhos distintos*: $\forall x \exists y \exists z (\neg(y = z) \wedge E(x, y) \wedge E(x, z))$. Outra observação simples, mas poderosa é que uma *estrutura relacional é essencialmente um banco de dados relacional*. Este “insight” de Ted Codd (no final da década de 1960) posteriormente levou ao desenvolvimento de uma indústria bilionária em bancos de dados relacionais. Codd posteriormente foi agraciado com o ACM Turing Award.

3.2.5.1. Dedução Natural para Lógica de Predicados

As provas no cálculo de dedução natural para lógica de predicados são similares às provas para lógica proposicional. Introduce-se regras para os quantificadores e para o símbolo de igualdade. As regras para os conectivos são preservadas: as regras para lógica proposicional continuam válidas para a lógica de predicados. Novamente, nossa apresentação sumariza os sistemas apresentados em [51, 66, 76] que apresentam um sistema de dedução natural para lógica de predicados com estrutura hierárquica no formato de caixas. Primeiramente, temos as regras para igualdade (=). A reflexividade da igualdade permite escrever em qualquer linha de prova que $t = t$. Isto é, qualquer *termo* t é igual a si mesmo, o que na verdade é um axioma. Dada uma fórmula arbitrária α , se provamos $\alpha(t)$ e provamos que $t = u$, sendo t, u termos, então pela substituição de termos iguais = *sub* podemos mostrar $\alpha(u)$. As regras estão representadas na Figura 3.8. Exemplo: Podemos mostrar,

1.	\vdots	1.	$\alpha(t)$	provado
2.	$t = t$	2.	\vdots	
	=refl	3.	$t = u$	provado
		4.	$\alpha(u)$	=sub (1,3)

Figura 3.8. Regras de Dedução Natural para Igualdade (=)

¹⁰Por questões de espaço a apresentação formal da semântica da lógica de primeira ordem não é apresentada neste texto. No entanto, pode ser compreendida através da leitura dos excelentes livros [54, 76].

usando as regras acima que (exemplo de [76]):

$$x + 1 = 1 + x, (x + 1 > 1) \rightarrow (x + 1 > 0) \vdash (1 + x) > 1 \rightarrow (1 + x) > 0$$

1. $(x + 1) = (1 + x)$ premissa
2. $(x + 1 > 1) \rightarrow (x + 1 > 0)$ premissa
3. $(1 + x > 1) \rightarrow (1 + x > 0)$ =sub(1,2)

Regras para o Quantificador Universal (\forall)

Para qualquer fórmula $\forall x\alpha(x)$, podemos derivar $\alpha(t)$ para qualquer termo fechado t . Isto se deve ao fato de afirmações universais serem verdade para quaisquer objetos de um domínio, isto é para quaisquer termos fechados. Assim, a regra de eliminação do quantificador universal denotada por $\forall E$ pode ser representada como na Figura 3.9. A regra de introdução do quantificador universal $\forall I$ requer a utilização de raciocínio hipotético. É necessário provar, para um termo arbitrário t , ainda não utilizado em nenhuma outra linha de prova, que $\alpha(t)$ é válido. Note que nesta regra escrevemos um termo em uma linha de prova, o que não ocorre nas demais regras do sistema de dedução natural. Note que t é arbitrário, não sendo utilizado na prova posteriormente. A anotação feita à direita $t - \forall I$, ao lado do termo t na linha 1, denota este termo arbitrário t , para o qual temos de provar a fórmula $\alpha(t)$, conforme a Figura 3.9.

Regras para o Quantificador Existencial \exists

Para mostrarmos uma sentença $\exists x\alpha$ através do sistema de dedução natural, temos que mostrar que $\alpha(t)$ para um termo fechado t (que não inclui variáveis). Assim, a regra de introdução do quantificador existencial $\exists I$ é representada na Figura 3.9. A eliminação do quantificador existencial exige a utilização de uma análise de casos, através de raciocínio hipotético. Lembre-se que a quantificação existencial $\exists x\alpha(x)$ significa que α é verdadeira para pelo menos uma atribuição de x ; isto é $\exists x\alpha(x)$ pode ser interpretado como uma disjunção. Se provamos a sentença $\exists x\alpha(x)$, a partir da hipótese de que $\alpha(t)$ é verdadeiro construímos uma prova de β , então podemos afirmar que β foi derivado a partir de $\exists x\alpha(x)$. No formato hierárquico que estamos utilizando, a regra é representada como a seguir. Como exemplo, provamos o seguinte sequente: $\exists x\neg\alpha(x) \vdash \neg\forall x\alpha(x)$.

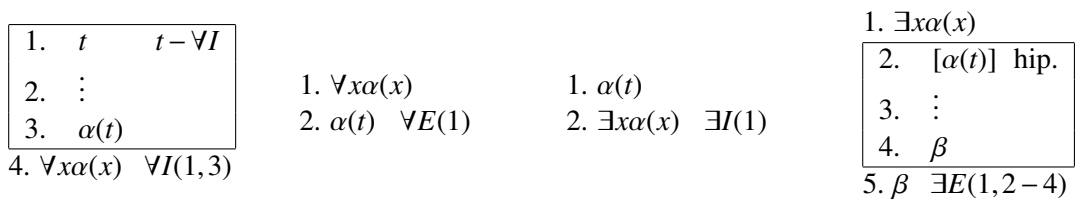


Figura 3.9. Regras de Dedução Natural para os Quantificadores \forall, \exists .

1.	$\exists x \neg \alpha(x)$ <i>premissa</i>						
2.	<table style="border: 1px solid black; width: 100%;"> <tr> <td style="width: 5%; text-align: right; vertical-align: top;">3.</td> <td style="padding: 5px;">$[\neg \alpha(t)]$ <i>hip.</i></td> </tr> <tr> <td style="text-align: right; vertical-align: top;">4.</td> <td style="padding: 5px;">$\alpha(t)$ $\forall E(2)$</td> </tr> <tr> <td style="text-align: right; vertical-align: top;">5.</td> <td style="padding: 5px;">\perp $\neg E(3,4)$</td> </tr> </table>	3.	$[\neg \alpha(t)]$ <i>hip.</i>	4.	$\alpha(t)$ $\forall E(2)$	5.	\perp $\neg E(3,4)$
3.	$[\neg \alpha(t)]$ <i>hip.</i>						
4.	$\alpha(t)$ $\forall E(2)$						
5.	\perp $\neg E(3,4)$						
6.	\perp $\exists E(1,3-5)$						
7.	$\neg \forall x \alpha(x)$ $\neg I(2,6)$						

Explicação: Para provarmos $\neg \forall x \alpha(x)$, temos $\exists x \neg \alpha(x)$ como premissa. Note que a conclusão é uma fórmula negada. Portanto, podemos usar a regra de $\neg I$ para prová-la; fazemos isto assumindo $\forall x \alpha(x)$ como hipótese para derivar uma contradição \perp . Observe que, ao assumirmos $\forall x \alpha(x)$ e tendo a premissa da linha 1 (i.e. $\exists x \neg \alpha(x)$), a hipótese $\neg \alpha(t)$ da linha 3 pode levar a uma contradição com a fórmula $\alpha(t)$ da linha 4. Assim, obtemos a conclusão.

3.2.5.2. Outros Métodos de Prova

Infelizmente, devido a restrições de espaço, a apresentação de outros métodos avançados, incluindo métodos de prova automática de teoremas é mais adequada para um curso de automação do raciocínio. Por exemplo, o princípio (ou método) da *Resolução* é uma regra de inferência do cálculo de predicados e utilizado extensivamente em inteligência artificial, devido à sua eficiência [78, 79]. O princípio da resolução foi estudado inicialmente por Davis e Putnam na década de 1960. Robinson estendeu o princípio, tornando-o mais eficiente, notadamente por fazer uso de um algoritmo de unificação (detalhes em [79]). A resolução é o mecanismo de execução/automação da linguagem Prolog. Neste método, todas as fórmulas envolvidas devem estar na forma clausal. Para quaisquer duas cláusulas C_1 e C_2 , se existe um literal L_1 em C_1 que é complementar a um literal L_2 em C_2 , então remova L_1 e L_2 de C_1 e C_2 , respectivamente, e construa a disjunção das cláusulas restantes. Ou, se α, β são cláusulas e p_i, q_j são fórmulas atômicas, então a regra é:

$$\frac{\alpha \vee p_1 \vee \dots \vee p_m \quad \beta \vee \neg q_1 \vee \dots \vee \neg q_m}{(\alpha \vee \beta)\theta}$$

onde θ é o *Unificador Mais Geral (UMG)* de todas as fórmulas p_i e q_j . O UMG é obtido por *unificação*; a unificação busca determinar se dois termos coincidem. No caso da lógica proposicional, é possível construir um provador de teoremas utilizando o princípio da resolução que é correto e completo. Uma característica essencial da resolução é o fato

deste método funcionar exclusivamente sobre sentenças em *forma clausal* - disjunções de literais (átomos ou negações de átomos). A forma clausal é utilizada por métodos automáticos de prova desde os primórdios desta área. A transformação de uma sentença em forma clausal preserva a consistência (ou inconsistência) da sentença original, e desta forma o uso de cláusulas facilita a busca por refutações. Os métodos não-clausais também fazem uso das idéias sugeridas para melhorar o desempenho de provadores de teoremas clausais.

Outro método de inferência relevante é o conhecido como tableaux analítico, desenvolvido inicialmente por Beth e aprimorado por Smullyan [54, 92]. É um método refutacional. Ou seja, para provarmos o sequente $\beta_1, \dots, \beta_n \vdash \alpha_1, \dots, \alpha_m$, afirmamos os antecedentes β_1, \dots, β_n em busca de uma contradição; se a contradição é mostrada pelo tableau, o sequente está provado.

Os assuntos relacionados à prova automática de teoremas são de alta complexidade, demandando conhecimento de diversas lógicas, algoritmos e complexidade computacional. A automação do raciocínio tem uma longa história. Ainda na década de 1950, Martin Davis construiu um programa para automatizar a aritmética de Pressburger (a teoria de primeira ordem dos naturais com adição). Allan Newell, Herbert Simon and Cliff Shaw desenvolveram o “Logic Theorist”: um programa que imitava o raciocínio humano, que também foi fundamental nos primórdios da inteligência artificial. Este sistema foi capaz de provar 38 dos 52 primeiros teoremas do *Principia Mathematica* [99]. Na década de 1960, os algoritmos de Davis-Putnam (1960) e o algoritmo DPLL (Davis-Putnam-Logemann-Loveland, 1962) foram desenvolvidos. DPLL é aplicado na resolução de problemas de satisfatibilidade (SAT). Até hoje variações e extensões do algoritmo DPLL são utilizadas pelos modernos “SAT solvers”. Para problemas de SAT em lógica de primeira ordem podemos utilizar SMT (Satisfiability Modulo Theories), que têm obtido sucesso notável na área de verificação, particularmente em aplicações em engenharia de software [60]. A área de projeto de circuitos integrados e aplicações críticas também fazem uso intensivo de métodos automáticos de prova.

3.2.6. Lógicas-Não-Clássicas

Em lógica clássica assumimos uma única situação, um único mundo possível em que fórmulas são avaliadas, isto é, em que assumem valores-verdade. Em diversos domínios do conhecimento, este modelo não é satisfatório. Em Ciência da Computação, nosso conhecimento do mundo, pode se referir a outros “mundos/estados possíveis” [72]. Raciocinar/refletir sobre conhecimento, computações e ações é fundamental em computação: como processos se coordenam para executar uma ação? O que robôs precisam saber para executar suas tarefas? O que programas (“agentes”) precisam saber sobre os outros programas para se comunicarem? As lógicas que investigam estas questões são lógicas não-clássicas: modais, temporais, epistêmicas, entre outras.

Além disso, sistemas computacionais são dinâmicos e interativos. O tempo, por exemplo, é um aspecto fundamental, assim como estados, transições, trocas de mensagens. Para aplicações em Ciência da Computação, Inteligência Artificial entre outras, estas lógicas requerem uma formalização/axiomatização distinta da realizada através de lógica clássica. A lógica clássica é monotônica, i.e. um resultado não é refutado sob

novas informações. Em inteligência artificial, no entanto, muitas vezes é necessário utilizar raciocínios não-monotônicos. Tendo em vista as características e a complexidade das aplicações em IA e computação, surgem então diversas propostas de lógicas “não-clássicas”.¹¹

Entre as lógicas originárias da Ciência da Computação podemos citar as lógicas desenvolvidas para raciocinar sobre programas, a partir dos trabalhos de Hoare, Dijkstra e Floyd [76]. Hoare e Milner também desenvolveram cálculos computacionais para expressar propriedades de concorrência e interação, que demandam novas lógicas computacionais para expressar propriedades de sistemas concorrentes [74, 85]. Dijkstra foi um pioneiro da área de verificação formal e da *derivação de programas*, método no qual um programa e sua prova são desenvolvidos concomitantemente. Uma lógica modal que



Figura 3.10. E.W. Dijkstra, C.A.R. Hoare, Robin Milner: Turing Awards (1972, 1980, 1991)

se tornou extremamente útil em ciência da computação é a lógica temporal. O raciocínio sobre tempo é fundamental em computação. Este requisito essencial do ponto de vista computacional, tem conduzido ao desenvolvimento de um considerável número de sistemas lógicos temporais. Estes desenvolvimentos iniciaram-se com o estudo de lógicas modais temporais, com forte impacto desde o trabalho de Pnueli *On the temporal logic of programs*, a partir de 1977 [87]. As lógicas temporais tem mostrado efetiva aplicabilidade na verificação de programas, verificação de hardware, teoria de bancos de dados, computação distribuída. É relevante observar que a lógica LTL (Linear Temporal Logic) proposicional tem o poder expressivo da lógica de primeira ordem sobre os naturais [68].

3.2.6.1. Lógicas Não-Clássicas: Lógicas Modais

Not ignorance, but ignorance of ignorance, is the death of knowledge.
Alfred North Whitehead

As lógicas modais formalizam modos de verdade. Em filosofia, são relacionadas à epistemologia, sendo relevantes no estudo do conhecimento. São as lógicas das verdades contingenciais, do “necessário” e “possível”. Sua origem moderna remonta a origem a Lewis [82]. Há, também, extensões para lógica modal de primeira ordem. Na sintaxe

¹¹Além disso, as lógicas não-clássicas encontram um grande respaldo da comunidade de Ciência da Computação, inclusive através de um grande número de Periódicos (Journals), Handbooks, Livros e Conferências. É também notável que um percentual significativo de pesquisadores que atuam em áreas relacionadas à lógica computacional tenham sido vencedores do ACM Turing Award. A lista de premiados inclui M. Rabin & D. Scott, C.A.R. Hoare, R. Milner, J. McCarthy, E. Codd, S. Cook, A. Pnueli, E. Clarke, A. Emerson e J. Sifakis, que realizaram trabalhos em áreas diretamente vinculadas à lógica.

das lógicas modais, adiciona-se inicialmente operadores unários à lógica clássica proposicional. Os operadores básicos \Box , \Diamond adicionados à lógica proposicional/predicados são $\Box\phi$: que lemos ϕ “é necessariamente verdadeiro”, e $\Diamond\phi$ que é lido “ ϕ é possivelmente verdadeiro” (e equivalente a $\neg\Box\neg\phi$). Em lógica modal uma proposição é necessária em um mundo ω se ela for verdadeira em todos os mundos que são possíveis em relação a este mundo ω , enquanto que uma proposição é possível em um mundo se ela for verdadeira em pelo menos um mundo possível alternativo, em relação a ω . Esta relação é formalizada através de uma relação binária de acessibilidade entre mundos possíveis. As lógicas modais foram consideradas apropriadas para estudar provas (lógica de provas), tempo, conhecimento, crenças, obrigações e outras modalidades [51, 62]. Em inteligência artificial e computação, lógicas modais estão entre os formalismos mais adequados para análise e representação de conhecimento e raciocínio sobre sistemas distribuídos e concorrentes [62, 65].

Em aplicações em epistemologia, e em Ciência da Computação, na modelagem de conhecimento em sistemas distribuídos e em inteligência artificial, $\Box\phi$ é lido como “de acordo com o meu conhecimento”, “de acordo com as leis da física”, “após o término do programa”. Notadamente a partir da década de 1950, os filósofos da lógica (e epistemologistas) Hintikka e Kripke propuseram a *semântica de mundos possíveis* [73, 81]. A partir da década de 1970, as lógicas modais passaram a ser muito utilizadas em Ciência da Computação na modelagem e o raciocínio sobre tempo, espaço, crenças, ações e conhecimento, individualmente ou através de combinações destas. Uma obra de referência que tornou as lógicas modais relativamente populares em computação (dizemos relativamente, pois o estudo de lógica computacional reúne uma comunidade científica pequena) foi o agora clássico “Reasoning about Knowledge” de Fagin, Halpern, Moses, Vardi [62]. Ademais, as propriedades de decidibilidade da lógica modal motivam seu uso em computação. Vardi mostra que elas são robustamente decidíveis, sendo adequadas, portanto, para aplicações em computação. [98].



Figura 3.11. Saul Kripke, Amir Pnueli, Moshe Vardi

A lógica modal encontra diversas aplicações. Diversas leituras (ou interpretações) das modalidades são possíveis, por exemplo:

- Modalidades Aléticas:
 - $\Box\phi$: ϕ é necessariamente verdadeiro.
 - $\Diamond\phi$: ϕ é possivelmente verdadeiro (equivalente a $\neg\Box\neg\phi$).
- Modalidades epistêmicas
 - $K_a\phi$: Agente a sabe ϕ .
 - $B_a\phi$: Agente a acredita que ϕ é verdade.

- Modalidades temporais lineares:
 - $\circ\phi$ (ou $X\phi$): ϕ será verdade no próximo ponto de tempo (estado do mundo).
 - $\square\phi$ (ou $G\phi$): ϕ será verdadeiro sempre.
 - $\diamond\phi$ (ou $F\phi$): ϕ será possivelmente (contingencialmente) verdadeiro no futuro (equivalente a $\neg\square\neg\phi$: ϕ não será sempre falso).
 - $\phi\cup\psi$: ϕ é verdadeiro até que ψ seja verdadeiro.

A axiomatização básica da lógica modal acrescenta novos axiomas à lógica proposicional. O Sistema Axiomático da lógica modal conhecido como sistema K , consiste dos axiomas abaixo e duas regras de inferência:

A1 Todas as tautologias proposicionais.

A2 (**K**) $\square(\alpha \rightarrow \beta) \rightarrow (\square\alpha \rightarrow \square\beta)$

R1 De $\alpha, \alpha \rightarrow \beta$ infira β (MP)

R2 De α infira $\square\alpha$ (Generalização)

Kripke demonstrou, na década de 1960, que o sistema K é um sistema axiomático correto e completo em relação ao modelo semântico. O axioma A2 é hoje chamado de **K** em homenagem a Kripke. Os sistemas lógicos modais ditos normais contêm o axioma **K** e a regra de generalização acima. Quanto a exemplos de fórmulas, podemos citar: $\square\square A$: é necessário que A seja necessário (ou “é conhecido que A é conhecido”); $\square\diamond A$: é necessário que A é possível; $\square A \rightarrow A$: se A é conhecido, então A é verdadeiro (ou se A é necessariamente verdadeiro, então A é verdadeiro). É importante observar que a cada propriedade da relação de acessibilidade corresponde um axioma da lógica modal. Por exemplo, a seguinte lista apresenta alguns axiomas da lógica modal e a correspondência com as correspondentes propriedades da relação binária R .

T:	$\square\alpha \rightarrow \alpha$	$\forall xR(x, x)$	(Reflexiva)
4:	$\square\alpha \rightarrow \square\square\alpha$	$\forall x, y, z(R(x, y) \wedge R(y, z)) \rightarrow R(x, z)$	(Transitiva)
B:	$\alpha \rightarrow \square\diamond\alpha$	$\forall x, y(R(x, y) \rightarrow R(y, x))$	(Simétrica)
D:	$\square\alpha \rightarrow \diamond\alpha$	$\forall x\exists yR(x, y)$	(Serial)
5:	$\diamond\alpha \rightarrow \square\diamond\alpha$	$\forall x, y, z(R(x, y) \wedge R(x, z)) \rightarrow R(y, z)$	(Euclidiana)

Assim, obtemos diferentes sistemas de lógicas modais considerando subconjuntos destes axiomas. Por exemplo:

K: A1, A2, R1, R2.

KD45: K, D, 4, 5 + A1, R1, R2.

KT4: tipicamente chamado de **S4**

KT45: tipicamente chamado de **S5**.

O axioma T corresponde à propriedade em R da reflexividade; 4 corresponde à propriedade da transitividade; B corresponde à propriedade da simetria; D corresponde à propriedade serial e 5 corresponde à propriedade euclidiana.

3.2.6.2. Semântica de Mundos Possíveis

Embora existam diversos modelos semânticos para as lógicas modais, esta síntese se concentra na semântica relacional de Kripke/Hintikka. O modelo dos mundos possíveis de

Kripke é utilizado de forma ampla, inclusive como modelo de conhecimento, para lógicas modais epistêmicas. Kripke [80, 81] definiu uma teoria de modelos baseada em “mundos possíveis” e relações de acessibilidade explícita entre mundos. Estes modelos são aplicáveis em uma classe ampla de lógicas. A noção de que mundos são “possíveis” depende do mundo atual, ou de referência. Por exemplo, em lógica modal temporal as noções de passado e futuro dependem do mundo atual. Se considerarmos um conjunto W de mundos possíveis e uma relação binária R em W , podemos ler $R(\omega_1, \omega_2)$ como ω_2 é um mundo possível tendo ω_1 como referência, ou a partir de ω_1 . Podemos considerar diversas propriedades das relações de acessibilidade. Um *frame* de Kripke é definido por uma estrutura (W, R) onde W é um conjunto de não vazio e R é uma relação em W .

A noção de relação de acessibilidade também é relevante pois sistemas diferentes de lógicas modais se caracterizam pelas propriedades das relações de acessibilidade. Por exemplo, se definirmos que a relação de acessibilidade é reflexiva, simétrica e transitiva definimos um sistema lógico conhecido como lógica modal $S5$ (ou $KT45$) [62]. De forma simplificada, podemos definir então uma Estrutura de Kripke: $M = (W, R, \nu)$, onde: W é um conjunto de mundos possíveis; $R \subseteq W^2$ é a relação de acessibilidade entre mundos; e $\nu(\omega)$ associa a cada mundo $\omega \in W$ um conjunto de variáveis proposicionais. A satisfação de fórmulas $(M, \omega) \models \alpha$ (que pode ser lida α é verdadeira em (M, ω) ou α é satisfeita em (M, ω)), pode então ser definida:

- $(M, \omega) \models p$ se, e somente se, $p \in \nu(\omega)$.
- $(M, \omega \models \neg\alpha \iff (M, \omega) \not\models \alpha$
- $(M, \omega \models \alpha \wedge \beta \iff (M, \omega) \models \alpha$ and $(M, \omega) \models \beta$
- $(M, \omega \models \alpha \vee \beta \iff (M, \omega) \models \alpha$ or $(M, \omega) \models \beta$
- $(M, \omega \models \alpha \rightarrow \beta \iff (M, \omega) \models \beta$ or $(M, \omega) \not\models \alpha$
- $(M, \omega) \models \Box\varphi \iff$ para todo $\omega_i \in W$, se $R(\omega, \omega_i)$ então $(M, \omega_i) \models \varphi$
- $(M, \omega) \models \Diamond\varphi \iff$ existe ω_i tal que $R(\omega, \omega_i)$ e $(M, \omega_i) \models \varphi$

Observe que a definição de satisfação para os conectivos clássicos reflete a definição da lógica proposicional clássica. Em cada mundo possível, a lógica proposicional clássica segue a sua interpretação usual. A lógica modal é uma extensão da lógica clássica pois um modelo da lógica modal é uma estrutura de modelos da lógica clássica interdependentes de acordo com a satisfatibilidade das fórmulas modais. A noção de *validade* também é mais geral. A validade pode ser definida em relação a um dado modelo ou em relação a uma dada classe de modelos. Diferentes funções de avaliação para um mesmo frame (W, R) podem gerar um classe C de modelos associados a (W, R) . Dizemos então que, para um modelo $M = (W, R, \nu)$ e α uma fórmula bem-formada, α é *válida* no modelo M , denotado por $M \models \alpha$, se para todo ω em W , então $M, \omega \models \alpha$ (i.e. α é verdadeiro em todo mundo possível de W). A fórmula α será válida em uma classe C de modelos se para todo modelo M' de C , temos $M' \models \alpha$.

Também é possível considerar múltiplos agentes em uma lógica modal. Neste caso, cada

agente tem a sua própria visão de mundo, expresso por uma relação de acessibilidade individual. Cada agente teria uma modalidade \Box_i , indexada pelo agente, além de uma relação R_i . Em uma lógica epistêmica, podemos utilizar uma modalidade K para representar conhecimento. Assim $K_i\alpha$ significa que o agente i sabe α ; $K_2K_1\neg\alpha$ pode ser lido: o agente 2 sabe que o agente 1 sabe $\neg\alpha$. Por exemplo, o modelo a seguir representando três relações, expressaria um modelo de Kripke para a lógica $KT45^n$, com três agentes (exemplo de [76]). A Figura 3.12 representa o Modelo M a seguir (não mostramos reflexividade, transitividade na figura). $M = \langle W = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6\}; R_1 = \{(\omega_1, \omega_2), (\omega_1, \omega_3), (\omega_3, \omega_2), (\omega_4, \omega_5)\}$ $R_2 = \{(\omega_1, \omega_3), (\omega_4, \omega_5)\}$, $R_3 = \{(\omega_3, \omega_2), (\omega_5, \omega_6)\}$; $v = \{(\omega_1, q), (\omega_2, p), (\omega_2, q), (\omega_3, p), (\omega_4, q), (\omega_6, p)\}$

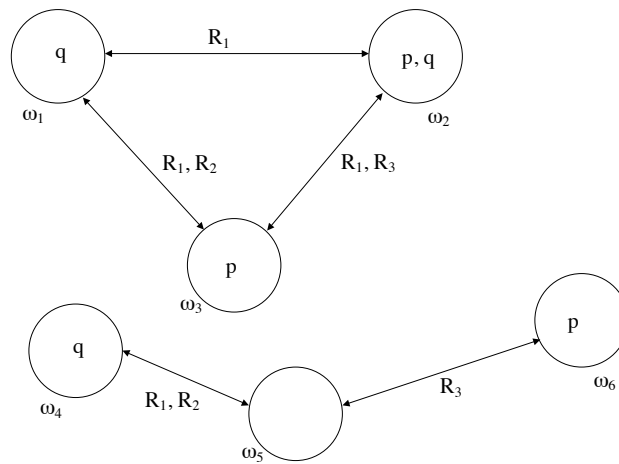


Figura 3.12. Exemplo de Modelo para $KT45^n$ (de [76])

3.2.6.3. Sistemas de Provas Para Lógicas Modais

Vários tipos de sistemas de prova para lógicas modais foram propostos, e.g., [51, 64]. Em alguns destes sistemas, as fórmulas são rotuladas pelos mundos (estados) em que elas são verdadeiras, o que pode facilitar no entendimento e desenvolvimento das provas. Nas regras apresentadas a seguir, no estilo de dedução natural, a notação $\varphi : \omega$ significa que φ é verdadeira no mundo possível ω . A relação de acessibilidade R também é explicitada neste sistema de dedução rotulado, o que também auxilia na derivação de em que mundos (estados) as fórmulas podem ser inferidas a partir da informação sobre as relações R entre mundos. As diferenças entre as diversas lógicas, por exemplo lógicas modais, são ditadas pelas propriedades diferentes das relações de acessibilidade. Isso é possível pela união, nas *unidades declarativas* que são constituídas de fórmulas com rótulos. Também definimos uma álgebra de rotulação que representa as propriedades específicas de cada sistema lógico. Há também regras de inferência estruturais (não formalizadas aqui por simplicidade) que permitem raciocínio sobre diagramas (relações) em configurações.

Na regra $\Box I$ a hipótese $[\mathcal{R}(\omega, g_\varphi(\omega))]$ é interpretada como: dado um mundo possível arbitrário $g_\varphi(\omega)$, se podemos inferir $\varphi : g_\varphi(\omega)$ então é possível provar que $\Box\varphi : \omega$. A

regra $\diamond E$ pode ser vista, informalmente, como uma *skolemização* do quantificador existencial sobre mundos possíveis, o que é semanticamente implicado pela fórmula $\diamond\varphi$ na premissa. Assim, o termo $f_\varphi(\omega)$ define um mundo possível particular unicamente associado à fórmula φ inferido e acessível a partir do mundo possível ω (i.e. $\mathcal{R}(\omega, f_\varphi(\omega))$). A regra de $\diamond I$ representa que se tivermos a relação $\mathcal{R}(\omega_1, \omega_2)$ e se φ é verdade em ω_2 então derivamos $\diamond\varphi$ no mundo ω_1 . Finalmente, a regra $\Box E$ significa que, se $\Box\varphi$ vale no mundo ω_1 e ω_1 está relacionado com (é acessível de) ω_2 então podemos inferir que φ é verdadeiro em ω_2 . Assim, as fórmulas são rotuladas pelos mundos em que elas são verdadeiras para facilitar o processo de raciocínio e inferência. As referências explícitas às relações de acessibilidade também auxiliam na derivação de fórmulas nos mundos relacionados por R [51].

Tabela 3.4. Regras para operadores modais

$\frac{\begin{array}{c} [R(\omega, g_\varphi(\omega))] \\ \vdots \\ \varphi : g_\varphi(\omega) \\ \hline \Box\varphi : \omega \end{array} \Box I}{\diamond\varphi : \omega} \diamond E$	$\frac{\Box\varphi : \omega_1, R(\omega_1, \omega_2)}{\varphi : \omega_2} \Box E$
$\frac{\varphi : f_\varphi(\omega), R(\omega, f_\varphi(\omega))}{\diamond\varphi : \omega} \diamond E$	$\frac{\varphi : \omega_2, R(\omega_1, \omega_2)}{\diamond\varphi : \omega_1} \diamond I$

Exemplo: Vamos provar $\Box p : w_0 \vdash \neg\diamond\neg p : w_0$ usando regras de dedução rotuladas.

1. $C_1\langle\Box p : w_0 \vdash \neg\diamond\neg p : w_0\rangle$ *premissa*
2. $C_2\langle\Box p : w_0, [\diamond\neg p : w_0]\rangle$ *hip.*
3. $C_3\langle\Box p : w_0, R(w_0, f_{\neg p}(w_0)), \neg p : f_{\neg p}(w_0)\rangle$ $\diamond E(C_2)$
4. $C_4\langle p : f_{\neg p}(w_0), \neg p : f_{\neg p}(w_0)\rangle$ $\Box E(C_2)$
5. $C_5\langle\perp : f_{\neg p}(w_0)\rangle$ $\wedge I(C_3)$
6. $C_6\langle\neg\diamond\neg p : w_0\rangle$ $\neg I(C_2, C_5)$

Explicação: Nesta prova, temos como premissa $\Box p : w_0$, isto é no mundo w_0 , $\Box p$ é verdadeiro. Para provar a conclusão $\neg\diamond\neg p : w_0$, assumimos, como hipótese que $\diamond\neg p : w_0$ para obtermos uma contradição \perp usando a regra de $\neg I$ em w_0 . Note que, em w_0 , temos agora $\Box p : w_0$, $\neg\diamond\neg p : w_0$ e $\diamond\neg p : w_0$ (hipótese). Para obtermos uma contradição, temos de usar as regras de eliminação $\Box E$, quanto $\diamond E$. Para usar $\diamond E$, temos mostrar que p é verdade em um novo mundo possível unicamente associado com p , isto é em $f_{\neg p}(w_0)$. Com a contradição entre $p : f_{\neg p}(w_0)$ e $\neg p : f_{\neg p}(w_0)$, concluímos a prova. Note que neste sistema rotulado de provas, fazemos uso das regras para os conectivos clássicos, conforme a Tabela 3.5.

Tabela 3.5. Regras para conetivos clássicos

$\frac{\frac{[\alpha : \omega] \quad [\beta : \omega]}{\gamma : \omega} \quad \gamma : \omega}{\alpha \vee \beta : \omega} \vee E$	$\frac{\alpha : \omega}{\alpha \vee \beta : \omega} \vee I$
$\frac{\alpha \wedge \beta : \omega}{\alpha : \omega, \beta : \omega} \wedge E$	$\frac{\alpha : \omega, \beta : \omega}{\alpha \wedge \beta : \omega} \wedge I$
$\frac{\alpha \supset \beta : \omega, \alpha : \omega}{\beta : \omega} \supset E$	$\frac{[\alpha : \omega] \quad \beta : \omega}{\alpha \supset \beta : \omega} \supset I$
$\frac{\neg \neg \alpha : \omega}{\alpha : \omega} \neg E$	$\frac{[\alpha : \omega] \quad \perp : \omega'}{\neg \alpha : \omega} \neg I$

3.2.6.4. Lógica em Inteligência Artificial: Integrando Raciocínio e Aprendizado

Em 2003, Valiant propôs um grande desafio para a Ciência da Computação [95]. O desafio consiste no desenvolvimento de modelos efetivos de computação cognitiva integrando raciocínio e aprendizado.

“The aim here is to identify a way of looking at and manipulating common-sense knowledge that is consistent with and can support what we consider to be the two most fundamental aspects of intelligent cognitive behaviour: the ability to learn from experience, and the ability to reason from what has been learned. We are therefore seeking a semantics of knowledge that can computationally support the basic phenomena of intelligent behaviour.”[95]

Recentemente, esta área tem sido objeto de pesquisas de um crescente número de pesquisadores, tendo em vista resultados promissores em inteligência artificial [55]. Para responder efetivamente a este desafio, além de conhecimentos sólidos de ciência da computação, é necessário entendimento da ciência cognitiva, área de estudo interdisciplinar da mente e cérebro. Esta área envolve psicologia, filosofia, IA, neurociência, linguística, antropologia. Em cognição, estudamos os processos do raciocínio humano, enquanto que em computação estes processos podem ou não ter inspiração na natureza. Várias áreas nos auxiliam a construir artefatos/algoritmos/sistemas computacionais mais expressivos, que podem contribuir neste desafio. Outro desafio, relacionado ao proposto por Valiant é o clássico questionamento levantado por Turing:

I propose to consider the question, “Can machines think?” This should begin with definitions of the meaning of the terms “machine” and “think”. The definitions might be framed so as to reflect so far as possible the normal use of the words,... but this attitude is dangerous, If the meaning of the words “machine” and “think” are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, “Can machines think?” is to be sought in a statistical survey such as a Gallup poll. But this is absurd.... [94].

Para realizar computação cognitiva, diversas habilidades devem ser consideradas. Uma alternativa inicial seria expressar algumas destas habilidades em sistemas artificiais. Estes modelos cognitivos mais completos requerem, portanto, a integração entre aprendizado, formação de regras (lógicas) e raciocínio (computação) [77]. Para construir modelos computacionais cognitivos que exigem a integração de raciocínio expressivo (através de lógicas) e aprendizado robusto é necessário definir uma semântica adequada para computação cognitiva, i.e. que represente habilidades cognitivas.

Como aprendizado de máquina é usualmente estudado através de abordagens estatísticas, experimentais, enquanto que o raciocínio em IA é estudado através de abordagens baseadas em lógicas, uma alternativa seria desenvolver modelos integrados, híbridos, por exemplo, modelos neuro-simbólicos. A computação neuro-simbólica explora os benefícios de cada paradigma: aprendizado (neural) e raciocínio simbólico (lógico) [59]. O exemplo a seguir mostra como modelos neuro-simbólicos podem ser realizados. Nesta abordagem, o conhecimento é representado por uma linguagem lógica simbólica, enquanto que a computação é realizada por algoritmos conexionistas, através de redes neurais. Para ilustrar a abordagem, utilizaremos a Lógica Modal Conexionista (Connectivist Modal Logic - CML) [58]. O *insight* em CML é considerar as redes neurais como redes de mundos possíveis, como na Figura 3.13.

No exemplo, considere que as 3 redes estão relacionadas, podendo se comunicar. Considere as redes neurais N_1, N_2, N_3 como mundos possíveis. Observe que em N_1 é possível, por exemplo, usar o raciocínio sobre a fórmula $\Box q$ para deduzirmos q em N_2 . Por sua vez, em N_2 temos s ; assim, podemos mostrar $\Diamond s$ em N_1 . Outros raciocínios são possíveis. Regras de inferência podem ser aprendidas pela rede neural, integrando, assim, raciocínio lógico e aprendizado de máquina. Observe que estas redes neuro-simbólicas tem propriedades interessantes, inclusive em relação à computação de pontos fixos de programas lógicos. Por exemplo, para qualquer programa P, existe uma rede neural N tal que N computa o operador de ponto fixo de programas modais, de programas lógicos que representam raciocínio intuicionista e de linguagens restritas de programas temporais [56, 57, 58].

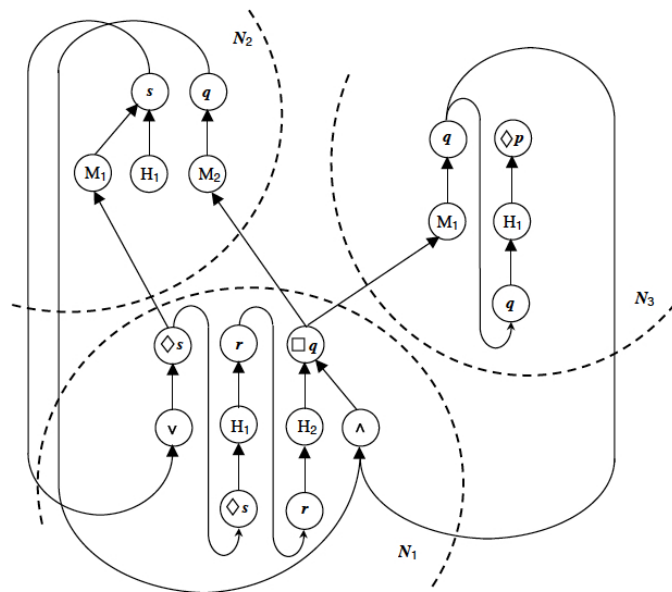


Figura 3.13. Conjunto de redes que representa o programa
 $P = \{r \rightarrow \Box q : \omega_1; \Diamond s \rightarrow r : \omega_1; s : \omega_2; q \rightarrow \Diamond p : \omega_3, R(\omega_1, \omega_2), R(\omega_2, \omega_3)\}$

Outras áreas em que a integração de lógica e aprendizado teve aplicações relevantes incluem raciocínio e aprendizado temporal, inclusive no desenvolvimento de simuladores de treinamento [61], devido à capacidade de aprender novas regras observando-se experts e novos usuários em treinamento. Nos problemas tipicamente utilizados como casos de estudo em sistemas distribuídos (*muddy children puzzle*, *wise men puzzle*, *dining philosophers* [62]), os resultados de aplicação desta metodologia também foram promissores, pois o sistema é capaz de aprender a resolver estes estudos de caso, encontrando a solução dos mesmos (veja detalhes em [59]). Finalmente, em engenharia de software, a abordagem permite adaptação e evolução de modelos e especificações temporais, servindo para verificar sistemas quando descrições de modelos são incompletas [49].

Também é relevante mencionar que a integração entre **lógica e probabilidade** tem sido tema de pesquisas por um longo período. Historicamente, De Morgan e Boole já haviam realizado estudos em que associavam raciocínio probabilístico e lógica clássica. Esta área de pesquisa tem como um de seus objetivos integrar a representação simbólica da lógica - que representa verdades absolutas, com a representação de incertezas - através de probabilidade [71, 75]. Embora este pareça ser um objetivo contraditório, a combinação de raciocínio qualitativo (simbólico) e quantitativo (probabilístico) pode levar à construção de sistemas, particularmente em inteligência artificial, mais expressivos [88].

3.2.7. Sumário

O estudo de sistemas lógicos têm sido fundamental desde as origens da Ciência da Computação, a ponto da lógica ser citada como o cálculo da computação [72]. Recentemente, temos presenciado resultados muito promissores de pesquisas em Ciência da Computação e Inteligência Artificial [90, 97], que demonstram que os sistemas computacionais atuais já são capazes de integrar habilidades cognitivas complexas, como o aprendizado a partir



Figura 3.14. Michael O. Rabin, Leslie G. Valiant

de experiências e o raciocínio sobre o conhecimento adquirido do ambiente. A construção de sistemas computacionais com complexidade crescente, exige, portanto, conhecimento sofisticado de técnicas, modelos e fundamentos da Ciência da Computação.

Os grandes desafios científicos sempre exigiram e, cada vez mais, irão demandar habilidade e capacidade de formalizar e construir raciocínios não triviais em sistemas de computação. Este capítulo ofereceu uma introdução mínima e incompleta à lógica aplicada. Esperamos que o leitor, a partir da bibliografia, possa explorar novos caminhos de forma independente. Para concluir de forma promissora, citamos Michael Rabin, vencedor do ACM Turing Award em 1976:

Our field is still in its embryonic stage. It's great that we haven't been around for 2000 years. We are at a stage where very, very important results occur in front of our eyes. M.O. Rabin em [91]

3.3. Conclusão e Perspectivas

Apresentamos uma introdução à área de Teoria da Computação com ênfase em duas subáreas fundamentais: Complexidade e Lógica Computacional. As duas subáreas escolhidas refletem as respectivas atuações dos autores em pesquisa e também refletem a dicotomia aceita na comunidade que distingue as chamadas teorias A e B, uma dicotomia adotada inclusive pelo periódico internacional *Theoretical Computer Science*. Enquanto que a subárea denominada de “teoria A” tem foco em algoritmos e sua complexidade usando ferramentas analíticas, combinatórias e probabilísticas, a teoria B tem foco nos métodos formais e lógicas para especificação e verificação.

Recentemente, através da visita de dois renomados pesquisadores internacionais ao Brasil, tivemos oportunidades valiosas para discutir perspectivas no desenvolvimento da área de pesquisa. O professor Moshe Y. Vardi da Rice University nos Estados Unidos é o editor-chefe do periódico *Communications of the ACM* e visitou em 2015 a UFRGS. Na ocasião, o professor Vardi apresentou a palestra “P vs NP” onde apresentou o problema do milênio e comentou sobre as tentativas para solucionar o problema. Mais recentemente, na sua coluna como editor do *CACM*, o professor Vardi escreveu sobre *The Moral Hazard of Complexity-Theoretic Assumptions* e comentou sobre a prova de Lászlo Babai de que o problema de Isomorfismo de Grafos admite um algoritmo de tempo quasi-polinomial. Embora o professor Vardi reconheça a importância do resultado, ele alerta para a distância cada vez maior entre a teoria e a prática como avaliação para medir complexidade e algoritmos. O professor John Hopcroft da Cornell University é ganhador do ACM Turing Award em 1986 e foi convidado para participar das comemorações pelos 100 anos da

Academia Brasileira de Ciências em 2016. Na ocasião, o professor Hopcroft apresentou a palestra “Entering the Information Age” onde questionou sobre o que devemos aprender diante do desafio de extrair informação de conjuntos enormes de dados, armazenamento e privacidade são importantes, e ferramentas matemáticas variadas são necessárias. O professor Hopcroft é autor de vários livros amplamente adotados para o ensino dos fundamentos em Teoria de Computação e mais recentemente tem se dedicado a escrever um novo livro sobre os fundamentos da Ciência de Dados onde propõe cobrir a teoria que possivelmente será usada nos próximos 40 anos, assim como a teoria dos autômatos e algoritmos constituiram a base para a teoria dos últimos 40 anos.

Referências

- [1] L. Alcon, L. Faria, C. M. H. de Figueiredo, and M. Gutierrez, The complexity of clique graph recognition, *Theoret. Comput. Sci.* **410** (2009) 2072–2083.
- [2] L. Alcon, L. Faria, C. M. H. de Figueiredo, and M. Gutierrez, Split clique graph complexity, *Theoret. Comput. Sci.* **506** (2013) 29–42.
- [3] L. Alcón, and M. Gutierrez, Cliques and extended triangles. A necessary condition for planar clique graphs, *Discrete Appl. Math.* **141** (2004) 3–17.
- [4] M. M. Barbosa, C. P. de Mello, and J. Meidanis, Local conditions for edge-colouring of cographs, *Congr. Numer.* **133** (1998) 45–55.
- [5] C. Berge, and V. Chvátal, *Topics on Perfect Graphs*, North-Holland Mathematics Studies, 88. *Annals of Discrete Mathematics*, 21. North-Holland Publishing Co., Amsterdam, 1984.
- [6] H. L. Bodlaender, C. M. H. de Figueiredo, M. Gutierrez, T. Kloks, and R. Niedermeier, Simple max-cut for split-indifference graphs and graphs with few P_4 's, Proc. of Third International Workshop on Experimental and Efficient Algorithms (WEA 2004). *Lecture Notes in Comput. Sci.* 3059 (2004), 87–99.
- [7] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: A survey*. SIAM Monographs on Discrete Mathematics and Applications, 1999.
- [8] K. Cameron, *Polyhedral and Algorithmic Ramifications of Antichains*, Ph.D. Thesis, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, 1982.
- [9] B. L. Chen, H.-L. Fu, and M. T. Ko, Total chromatic number and chromatic index of split graphs, *J. Combin. Math. Combin. Comput.* **17** (1995) 137–146.
- [10] A. G. Chetwynd, and A. J. W. Hilton, Star multigraphs with three vertices of maximum degree, *Math. Proc. Cambridge Philos. Soc.* **100** (1986) 303–317.
- [11] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković, Recognizing Berge graphs, *Combinatorica* **25** (2005) 143–186.
- [12] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The strong perfect graph theorem, *Ann. of Math.* **164** (2006) 51–229.

- [13] V. Chvátal, Star-cutsets and perfect graphs, *J. Combin. Theory Ser. B* **39** (1985) 189–199.
- [14] T. Feder, P. Hell, S. Klein, and R. Motwani, List partitions, *SIAM J. Discrete Math.* **16** (2003) 449–478.
- [15] C. M. H. de Figueiredo, S. Klein, Y. Kohayakawa, and B.A. Reed, Finding skew partitions efficiently, *J. Algorithms* **37** (2000) 505–521.
- [16] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello, On edge-colouring indifference graphs, *Theoret. Comput. Sci.* **181** (1997) 91–106.
- [17] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello, Local conditions for edge-coloring, *J. Combin. Math. Combin. Comput.* **32** (2000) 79–91.
- [18] C. M. H. de Figueiredo, Resolver ou Verificar? Uma pergunta que vale um milhão de dólares. *Ciência Hoje*, Rio de Janeiro pp. 42–46, novembro 2011.
- [19] C. M. H. de Figueiredo, The P vs. NP-complete dichotomy of some challenging problems in graph theory, *Discrete Appl. Math.* **160** (2012) 2681–2693.
- [20] C. M. H. de Figueiredo, and J. L. Szwarcfiter, Emparelhamentos em Grafos: Algoritmos e Complexidade, *XIX Congresso da Sociedade Brasileira de Computação. (Org.). XVIII Jornada de Atualização em Informática.* 1999, pp. 127–161.
- [21] C. M. H. de Figueiredo, J. Meidanis, and C. P. Mello, Coloração em Grafos, *XVII Congresso da Sociedade Brasileira de Computação. (Org.). XVI Jornada de Atualização em Informática* 1997, pp. 01–45.
- [22] L. Fortnow, The status of the P versus NP problem, *Communications of the ACM* **52** (2009) 78–86.
- [23] M. R. Garey, and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-completeness.* WH Freeman, New York, 1979.
- [24] M. Grötschel, L. Lovász, and A. Schrijver, Polynomial Algorithms for Perfect Graphs, In *Topics on Perfect Graphs*, C. Berge and V. Chvátal, eds., North-Holland Mathematics Studies 88, Annals of Discrete Mathematics 21, North-Holland Publishing Co., Amsterdam, 1984, pp. 325–356.
- [25] A. Gyárfás, Problems from the world surrounding perfect graphs, *Zastos. Mat.* **19** (1987) 413–441.
- [26] D. G. Hoffman, and C. A. Rodger, The chromatic index of complete multipartite graphs, *J. Graph Theory* **16** (1992) 159–163.
- [27] I. Holyer, The NP-completeness of edge-coloring, *SIAM J. Comput.* **10** (1981) 718–720.
- [28] D. S. Johnson, *NP-Completeness Columns*, available at <http://www2.research.att.com/dsj/columns/>

- [29] D. S. Johnson, Graph restrictions and their effect, *J. Algorithms* **6** (1985) 434–451.
- [30] R. C. S. Machado, and C. M. H. de Figueiredo, Complexity separating classes for edge-colouring and total-colouring, *J. Braz. Comput. Soc.* **17** (2011) 281–285.
- [31] R. C. S. Machado, C. M. H. de Figueiredo, and K. Vušković, Chromatic index of graphs with no cycle with a unique chord, *Theoret. Comput. Sci.* **411** (2010) 1221–1234.
- [32] T. A. McKee, and F. R. McMorris, *Topics in Intersection Graph Theory*, SIAM Monographs on Discrete Mathematics and Applications, 1999.
- [33] T. Niessen, How to find overfull subgraphs in graphs with large maximum degree, *Electron. J. Combin.* **8** (2001) #R7.
- [34] C. Ortiz Z., N. Maculan, and J. L. Szwarcfiter, Characterizing and edge-colouring split-indifference graphs, *Discrete Appl. Math.* **82** (1998) 209–217.
- [35] M. Plantholt, The chromatic index of graphs with a spanning star, *J. Graph Theory* **5** (1981) 45–53.
- [36] E. Prisner, *Graph Dynamics*, Pitman Research Notes in Mathematics 338, Longman, 1995.
- [37] J. L. Ramirez Alfonsin, and B. A. Reed, *Perfect graphs*, Wiley-Interscience Series in Discrete Mathematics and Optimization, 2001.
- [38] B. Randerath, and I. Schiermeyer. Vertex colouring and forbidden subgraphs—a survey, *Graphs Combin.* **20** (2004) 1–40.
- [39] F. Roberts, and J. Spencer, A characterization of clique graphs, *J. Combin. Theory Ser. B* **10** (1971) 102–108.
- [40] C. Simone, and C. P. de Mello, Edge-colouring of join graphs, *Theoret. Comput. Sci.* **355** (2006) 364–370.
- [41] C. Simone, and A. Galluccio, Edge-colouring of regular graphs of large degree, *Theoret. Comput. Sci.* **389** (2007) 91–99.
- [42] J. P. Spinrad, *Efficient graph representations*, Fields Institute Monographs, 19. AMS, 2003.
- [43] E. Steffen, Classifications and characterizations of snarks, *Discrete Math.* **188** (1998) 183–203.
- [44] J. L. Szwarcfiter, Recognizing clique-Helly graphs, *Ars Combin.* **25** (1997) 29–32.
- [45] J. L. Szwarcfiter, A Survey on Clique Graphs, In *Recent Advances in Algorithms and Combinatorics*, C. Linhares-Sales and B. Reed, eds., CMS Books Math./Ouvrages Math. SMC, 11, Springer, New York, 2003, pp. 109–136.

- [46] N. Trotignon, and K. Vušković, A structure theorem for graphs with no cycle with a unique chord and its consequences, *J. Graph Theory* **63** (2010) 31–67.
- [47] D. B. West, *Introduction to Graph Theory*. Second edition. Prentice Hall, 2001.
- [48] Dirk Baltzly. Stoicism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2014 edition, 2014.
- [49] Rafael V. Borges, Artur d’Avila Garcez, and Luís C. Lamb. Learning and representing temporal knowledge in recurrent networks. *IEEE Transactions on Neural Networks*, 22(12):2409–2421, 2011.
- [50] K. Broda, S. Eisenbach, H. Khoshnevisan, and S. Vickers. *Reasoned Programming*. Prentice Hall, 1994.
- [51] K. Broda, D.M. Gabbay, L.C. Lamb, and A. Russo. *Compiled Labelled Deductive Systems: A Uniform Presentation of Non-classical Logics*. Studies in Logic and Computation. Research Studies Press/Institute of Physics Publishing, Baldock, UK, Philadelphia, PA, 2004.
- [52] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, April 1936.
- [53] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158, 1971.
- [54] Flavio Corrêa da Silva, Marcelo Finger, and Ana C. V. de Melo. *Lógica para Computação*. Thomson, São Paulo, 2006.
- [55] Artur d’Avila Garcez, Marco Gori, Pascal Hitzler, and Luís C. Lamb. Neural-Symbolic Learning and Reasoning (Dagstuhl Seminar 14381). *Dagstuhl Reports*, 4(9):50–84, 2015.
- [56] A.S. d’Avila Garcez and L.C. Lamb. A connectionist computational model for epistemic and temporal reasoning. *Neural Computation*, 18(7):1711–1738, 2006.
- [57] A.S. d’Avila Garcez, L.C. Lamb, and D.M. Gabbay. Connectionist computations of intuitionistic reasoning. *Theoretical Computer Science*, 358(1):34–55, 2006.
- [58] A.S. d’Avila Garcez, L.C. Lamb, and D.M. Gabbay. Connectionist modal logic: Representing modalities in neural networks. *Theoretical Computer Science*, 371(1-2):34–53, 2007.
- [59] A.S. d’Avila Garcez, L.C. Lamb, and D.M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer, 2009.
- [60] Leonardo M. de Moura and Nikolaj Bjørner. Satisfiability modulo theories: introduction and applications. *Commun. ACM*, 54(9):69-77, 2011.

- [61] Leo de Penning, Artur S. d’Avila Garcez, Luís C. Lamb, and John-Jules Ch. Meyer. A neural-symbolic cognitive agent for online learning and reasoning. In Toby Walsh, editor, *IJCAI-11*, pages 1653–1658. IJCAI/AAAI, 2011.
- [62] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
- [63] Frederic B. Fitch. *Symbolic Logic*. The Ronald Press Company, New York, 1952.
- [64] M. Fitting. *Proof methods for modal and intuitionistic logics*. D. Reidel Publishing Company, Dordrecht, 1983.
- [65] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyashev. *Many-dimensional modal logics: theory and applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science, 2003.
- [66] D. M. Gabbay. *Elementary Logics: a Procedural Perspective*. Prentice Hall, London, 1998.
- [67] Dov M. Gabbay and F. Guenther, editors. *Handbook of Philosophical Logic*, volume I-XVIII. Springer, 2008-2015.
- [68] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal basis of fairness. In *Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada, USA, January 1980*, pages 163–173, 1980.
- [69] Dov M. Gabbay and John Woods, editors. *Handbook of The History of Logic*, volume I-XI. Elsevier, 2008-2015.
- [70] G. Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, 39:176–210, 1934.
- [71] J.Y. Halpern. *Reasoning about Uncertainty*, MIT Press, 2003.
- [72] J.Y. Halpern, R. Harper, N. Immerman, P.G. Kolaitis, M.Y. Vardi, and V. Vianu. On the unusual effectiveness of logic in computer science. *Bulletin of Symbolic Logic*, 7(2):213–236, 2001.
- [73] J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.
- [74] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
- [75] C. Howson. Probability and logic. *J. Applied Logic* 1(3-4), 151-165, 2003.
- [76] M. Huth and M. Ryan. *Logic in Computer Science: modelling and reasoning about systems*. Cambridge University Press, 2000.
- [77] R. Khardon and D. Roth. Learning to reason. *Journal of the ACM*, 44(5):697–725, 1997.

- [78] Robert A. Kowalski. *Computational Logic and Human Thinking: How to be Artificially Intelligent*. Cambridge University Press, 2011.
- [79] Robert A. Kowalski. *Logic for Problem Solving, Revisited*. Herstellung und Verlag: Books on Demand, 2014.
- [80] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–4, 1959.
- [81] S. Kripke. Semantic analysis of modal logics I, normal propositional calculi. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.
- [82] C. Lewis. *A Survey of Symbolic Logic*. University of California Press, Berkeley, 1918.
- [83] Z. Manna and R. Waldinger. *The logical basis for computer programming. Volume 1: deductive reasoning*. Addison-Wesley, Boston, 1985.
- [84] E. Mendelson. *Introduction to mathematical logic*. Van Nostrand Reinhold, New York, 1964.
- [85] Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.
- [86] Joan Moschovakis. Intuitionistic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2015 edition, 2015.
- [87] A. Pnueli. The temporal logic of programs. In *Proceedings of 18th IEEE Annual Symposium on Foundations of Computer Science*, pages 46–57, 1977.
- [88] S.J. Russell. Unifying Logic and Probability: A New Dawn for AI? In *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014*, pages 10–14, 2014.
- [89] Ralph Schoenman, editor. *Bertrand Russell: Philosopher of the Century*. Allen and Unwin, London, 1967.
- [90] Bernhard Schoelkopf. Artificial intelligence: Learning to see and act. *Nature*, 518:486–487, 2015.
- [91] Denis Shasha and Cathy Lazere. *Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists*. Copernicus, 1995.
- [92] R. M. Smullyan. *First-Order Logic*. Dover Publications, New York, revised edition, 1995.
- [93] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2), 1936.
- [94] Alan M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

- [95] L. G. Valiant. Three problems in computer science. *Journal of ACM*, 50(1):96–99, 2003.
- [96] Dirk Van Dalen. Intuitionistic logic. In D. Gabbay and F. Guentner, editors, *Handbook of Philosophical Logic*, volume 166 of *Synthese Library*, pages 225–339. Springer Netherlands, 1986.
- [97] Moshe Y. Vardi. Is information technology destroying the middle class? *Commun. ACM*, 58(2):5, 2015.
- [98] M.Y. Vardi. Why is modal logic so robustly decidable? In N. Immerman and P. Kolaitis, editors, *Descriptive Complexity and Finite Models*, volume 31 of *Discrete Mathematics and Theoretical Computer Science*, pages 149–184. DIMACS, 1997.
- [99] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica*, volume I,II, III. Cambridge University Press, Cambridge, 1910, 1912, 1913.
- [100] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Kegan Paul, Trench, Turner & Co, London, 1922.

Capítulo

4

Agentes Inteligentes Conversacionais: Conceitos Básicos e Desenvolvimento

Flávia de Almeida Barros e Patrícia Azevedo Tedesco

Centro de Informática, Universidade Federal de Pernambuco
Av. Jornalista Aníbal Fernandes, s/n - Cidade Universitária
CEP 50.740-560 - Recife – PE - Brasil

{fab,pcart}@cin.ufpe.br

Resumo

Este curso tem por objetivo apresentar um panorama sobre Agentes Inteligentes Conversacionais, incluindo um histórico da área, conceitos básicos, desenvolvimento e aplicações. Tais agentes, também denominados de Chatterbots, são projetados para interpretar a pergunta/intervenção do usuário e oferecer respostas diretas, buscando manter a ilusão de que o interlocutor está conversando com outro ser humano. Atualmente, encontramos esses agentes nas mais diversas aplicações, tais como ensino a distância, jogos sérios, em SACs de lojas online, e mesmo em sistemas voltados a pessoas com deficiência física. Daremos maior ênfase aos Agentes Conversacionais Incorporados, que possuem representação gráfica, movimentos faciais e corporais, e possibilitam diálogos via voz, o que torna a interação mais natural e realista.

Abstract

This course aims to present an overview of Intelligent Conversational Agents, including a historical background, basic concepts, development and applications of such agents. These agents, also named as Chatterbots, are designed to interpret the user's question/intervention and to offer direct answers, with the aim of keeping the illusion that the user is talking to another human being. Currently, we find these in a wide variety of applications, such as distance learning, serious games, in online shops, and as personal assistants. Here we focus on Embodied Conversational Agents, which bear a graphical representation, facial and body movements, and are capable of performing voice dialogues, thus enabling a more natural and realistic interaction.

4.1. Introdução

A ideia de construir sistemas capazes de dialogar com o usuário em linguagem natural surgiu na década de 1950, a partir do trabalho de Alan Turing intitulado “*Computing Machinery and Intelligence*” [Turing, 1950], no qual ele discutia a questão: “As máquinas podem pensar?”. Nesse artigo, Turing propôs o “Jogo da Imitação”, atualmente conhecido como o Teste de Turing¹. Um sistema seria considerado inteligente se ele conseguisse se fazer passar por humano em um diálogo com um interlocutor humano.

O trabalho de Turing abriu a possibilidade de se desenvolverem “máquinas de diálogo”, mais tarde conhecidas como *Chatterbots*² (ou *Chatbots*), para as mais diversas aplicações [Laven, 2016]. Tais sistemas, denominados também de *Agentes Conversacionais* (ACs), são projetados para interpretar a pergunta (ou intervenção) do usuário, e oferecer respostas diretas, sempre buscando manter a coerência do diálogo e a ilusão de que o interlocutor está conversando com outro ser humano.

Encontramos atualmente diversas competições mundiais que realizam versões do Teste de Turing. Destacamos aqui o *Loebner Prize*³, que realiza eventos anuais de premiação, e o *Turing Test 2014*, um evento organizado pela Universidade de Reading (Grã-Bretanha)⁴ que premiou o chatterbot Eugene⁵ como sendo o primeiro sistema a passar no Teste de Turing irrestrito. Vale mencionar que o trabalho de Turing também vem inspirando livros e filmes desde a década de 1960, com destaque para o filme autobiográfico “*The Imitation Game*”⁶ (2014), e diversos filmes do gênero ficção científica, como *Her*⁷ (2013), incluindo enredos onde o Teste de Turing é aplicado a “humanóides” – por exemplo, *Ex Machina*⁸ (2015), e o insuperável *Blade Runner* (1982)⁹.

As primeiras implementações desses sistemas eram puramente baseadas em texto. Na última década, contudo, os chatterbots *stand-alone* textuais perderam espaço para os chamados *Agentes Conversacionais Incorporados* (ACIs), agentes conversacionais que possuem uma representação gráfica e síntese/interpretação da fala (por exemplo, REA [Cassell et al., 1999] e GRETA¹⁰). Essa mudança se deve à crescente tendência de se desenvolverem ambientes virtuais imersivos, baseados em realidade virtual. Nesses casos, o usuário espera que o agente conversacional faça parte do ambiente simulado, possibilitando diálogos via voz, a fim de tornar a interação mais natural e realista.

¹ https://en.wikipedia.org/wiki/Turing_test

² O termo “*chatterbot*” foi introduzido por Michael Mauldin em 1994.

Ver <http://en.wikipedia.org/wiki/Chatterbot>

³ <http://www.loebner.net/Prize/loebner-prize.html>

⁴ <http://www.reading.ac.uk/news-and-events/releases/PR583836.aspx>

⁵ <http://www.reading.ac.uk/sse/news/sse-newsarticle-2014-06-08.aspx>

https://en.wikipedia.org/wiki/Eugene_Goostman

⁶ <http://theimitationgamemovie.com/>

⁷ <http://www.herthemovie.com/#/home>

⁸ <http://exmachina-movie.com/synopsis>

⁹ <http://www.warnerbros.com/blade-runner>

¹⁰ <http://perso.telecom-paristech.fr/~pelachau/Greta/>

Esse cenário motivou o crescimento das pesquisas na área, a fim de prover ACIs mais credíveis. Essas pesquisas envolvem não apenas a área de Processamento de Linguagem Natural (em particular, Análise da Conversão), como também Processamento de voz (síntese e análise), Processamento de imagem, e outras áreas correlatas. No que concerne a modelagem e implementação desses agentes, nota-se uma tendência de uso de representações em XML. O objetivo é atender a requisitos da Engenharia de Software, como modularidade e reúso.

Atualmente, encontramos agentes conversacionais (incorporados ou não) nas mais diversas aplicações, tais como em sistemas de ensino a distância, jogos sérios, em SACs de lojas online, em sistemas voltados a pessoas com alguma deficiência física, entre outros [Chatbots.org, 2016].

Este curso tem por objetivo apresentar a área de *Agentes Inteligentes Conversacionais*, incluindo conceitos básicos, desenvolvimento e aplicações. Em particular, daremos maior ênfase aos Agentes Conversacionais Incorporados (ACIs), que são sua forma mais atual de apresentação.

Este documento prossegue com uma visão geral sobre os *Sistemas Conversacionais*, que incluem qualquer tipo de sistema que seja capaz de manter um diálogo em linguagem natural com o usuário (Seção 4.2.1). A Seção 4.2.2 define os *Agentes Conversacionais*, buscando caracterizá-los como *agentes inteligentes*, dentro da visão mais moderna da Inteligência Artificial [Russel e Norvig, 2010].

A Seção 4.3 apresenta brevemente conceitos básicos sobre duas áreas intimamente ligadas ao desenvolvimento de agentes conversacionais: *Teorias Linguística* para Análise da Conversação e Análise do Discurso (Seção 4.3.1), e *Computação Afetiva* - área de estudos que serve de base para se projetar agentes artificiais com traços de personalidade e emoção (Seção 4.3.2).

A Seção 4.4 trata dos *Agentes Conversacionais textuais* (ou *Chatterbots*), precursores dos Agentes Conversacionais Incorporados. A Seção 4.5 apresenta os ACIs, agentes conversacionais possuem uma representação gráfica e síntese/interpretação da fala. Veremos um breve histórico da área, seguido de técnicas utilizadas na construção desses agentes. Discutiremos algumas aplicações existentes na área, concluindo com uma visão das tendências futuras da área.

A Seção 4.6 conclui o documento trazendo algumas considerações finais, apresentando desafios e tendências futuras de pesquisa e desenvolvimento na área. Por fim, temos a lista de referências bibliográficas citadas no documento.

Por fim, gostaríamos de ressaltar que todos os sítios na Web (URLs) indicados aqui estavam ativos durante o período de escrita deste documento (entre abril e maio de 2016).

4.2. Sistemas e Agentes Conversacionais

Como já mencionado, os sistemas conversacionais são projetados para interpretar a pergunta (ou intervenção) do usuário, e oferecer respostas diretas, buscando manter a coerência do diálogo e a ilusão de que o interlocutor está conversando com outro ser humano. Devido a essas características, Chatterbots podem ser considerados como *agentes inteligentes*, dentro da visão mais moderna da Inteligência Artificial (IA) [Russel e Norvig, 2010]. Tais agentes têm como característica marcante a capacidade de dialogar com os usuários, sendo então chamados de *Agentes Conversacionais*.

Considerando sua grande utilização em sistemas Web (de livre acesso ou não), os agentes conversacionais podem ser considerados como uma forma de interface, que complementa ou substitui outras formas de acesso ao computador. Atualmente, encontramos tais agentes em sistemas de ensino/educação a distância (EaD), em jogos digitais sérios ou para simples entretenimento, em sistemas de FAQ (*Frequent Asked Questions*), em SACs (Sistemas de Atendimento ao Cliente) de lojas online, entre outros (ver [ALICE Bot, 2016], [Cassell et al., 1999], [Chatbots.org, 2016], [Laven, 2016]).

Antes de avançar na apresentação sobre os ACs, vamos situar esses agentes dentro da história do desenvolvimento dos Sistemas Conversacionais. A seguir, a Seção 4.2.2 busca caracterizar os *agentes conversacionais* como *agentes inteligentes*, trazendo definições e estabelecendo a terminologia adotada neste documento. Como veremos, os Agentes Conversacionais são um tipo de sistema conversacional que apresenta características de agente virtual autônomo.

4.2.1 Evolução dos Sistemas Conversacionais

Os primeiros sistemas que buscavam interagir com o usuário em linguagem natural, conhecidos como *Sistemas de Pergunta-Resposta*, datam do início da década de 1960. Exemplos de tais sistemas são BASEBALL [Green et al., 1961], LUNAR [Woods, 1973], e SHRDLU (o mais conhecido e bem sucedido dentre os sistemas iniciais) [Winograd, 1972]. Contudo, tais sistemas não eram capazes de travar um diálogo mais flexível com o usuário, limitando-se a responder perguntas previamente catalogadas.

O primeiro sistema com ampla divulgação capaz de travar diálogos (simples) em linguagem natural foi ELIZA [Weizenbaum, 1966] [Eliza Chat Bot, 2016]. Mais tarde, tais sistemas receberam a denominação de *Chatterbots* (robôs de conversação). Esses sistemas marcaram a década de 1970, ainda compartilhando espaço com os sistemas pergunta-resposta.

Em paralelo aos chatterbots, temos ainda os *Sistemas para compreensão e geração de histórias*, que eram baseados em *scripts* [Schank et al., 1975]. Esses não eram sistemas de diálogo, porém ajudaram a avançar as pesquisas no Processamento de Linguagem Natural (PLN) [Allen, 1995], contribuindo assim para a melhoria dos sistemas conversacionais.

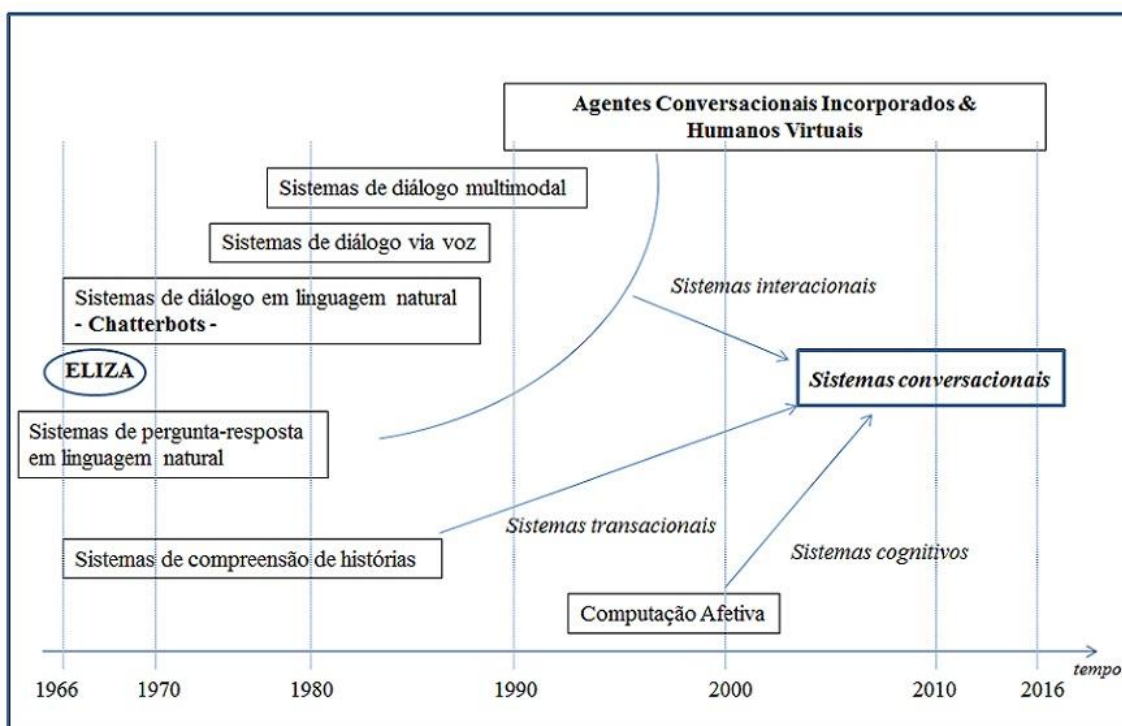


Figura 4.1: História do desenvolvimento dos Sistemas Conversacionais *—*figura adaptada de [Nishida et al., 2014].

Os sistemas conversacionais via voz, uma evolução dos chatterbots textuais, ganharam espaço na década de 1980, acompanhando a evolução na área de processamento de voz (interpretação e síntese). Exemplos de tais sistemas são: *Hearsay III* [Erman et al., 1980] e “*Put That There*” [Bolt, 1980].

No final da década de 1980, surgiu o sistema “*The Knowledge Navigator*”, o primeiro sistema conversacional via voz com uma imagem humana associada (conhecido como *Phil*)¹¹. Phil já pode ser considerado como um agente inteligente virtual (ver Seção 4.2.2). A partir daí, a evolução na área foi bem mais rápida, com os sistemas de diálogo multimodal, chegando até os Agentes Conversacionais Incorporados (e.g.¹², *REA* [Cassell et al., 1999]).

Vale ressaltar, contudo, que os chatterbots puramente textuais ainda são tema de pesquisa e desenvolvimento. Nesse caso, o foco das pesquisas é mais voltado à capacidade de manter diálogos credíveis com os interlocutores (Seção 4.4).

Por fim, destacamos os *Humanos Virtuais* (ou Atores Sintéticos), que também podem ser considerados como sistemas conversacionais, porém com foco não apenas na capacidade dialógica, mas principalmente na ilusão de vida (e.g., Projeto Oz [Mateas, 1999]). A fim de passarem mais credibilidade, os humanos virtuais também possuem personalidade e capacidade de demonstrar emoção, seguindo as pesquisas na área de

¹¹ http://en.wikipedia.org/wiki/Knowledge_Navigator; https://www.youtube.com/watch?v=_a0t2Eb7YJk

¹² e.g.: Abreviatura do termo em Latim “*exempli gratia*”, que, em Português, significa “por exemplo”.

Computação Afetiva [Picard, 2003] (Seção 4.3.2). Tais capacidades podem também ser embutidas nos ACIs, na tentativa de prover diálogos mais naturais e verossímeis.

4.2.2 Agentes Conversacionais como Agentes Inteligentes

Agentes conversacionais têm sido tradicionalmente desenvolvidos usando-se técnicas de Inteligência Artificial [Russel e Norvig, 2010], principalmente as técnicas oriundas da subárea de Processamento de Linguagem Natural [Allen, 1995]. Na década passada, esses agentes/sistemas também passaram a observar requisitos da Engenharia de Software, como modularidade e reuso [Wallace, 2009].

Esta seção busca caracterizar os agentes conversacionais como agentes inteligentes dentro do paradigma mais atual da IA. É importante fazer uma distinção entre agentes conversacionais puramente textuais (aqui denominados de *chatterbots* – Seção 4.4), e os ACs incorporados – agentes animados que possuem uma representação gráfica, sendo capazes de realizar o diálogo por meio de fala e de outras modalidades não verbais, como olhares e gestos (Seção 4.5). Vale ainda ressaltar que alguns *chatterbots* textuais apresentam uma imagem associada, porém não são considerados ACs incorporados, por não apresentarem as outras características dos ACIs mencionadas aqui.

De início, é importante definir quais as características indispensáveis a um software para qualificá-lo como um agente. Segundo [Russel e Norvig, 2010], um *agente* é qualquer entidade humana ou artificial (de software ou de hardware) que, estando imersa (ou situada) em um ambiente físico, virtual ou simulado, é capaz de:

- *perceber* seu ambiente através de sensores (*e.g.*, teclado, câmeras, microfone, protocolos de comunicação em sistemas multiagentes), e
- *agir* sobre esse ambiente através de atuadores (*e.g.*, tela do computador, autofalante, impressora, funções internas que enviam mensagens a outros agentes, braços, pernas...).

Agentes inteligentes (ou *racionais*, como denominados por [Russel e Norvig, 2010]) são aqueles capazes de *raciocinar* com base em suas percepções e seu conhecimento do mundo, a fim de escolher as ações mais adequadas a realizar no ambiente para alcançar seus *objetivos próprios*. Essa definição parece se adequar tanto a agentes humanos quanto aos artificiais.

A partir de agora, discutiremos apenas os *agentes artificiais*, que são o foco deste curso. Assim, quando não houver menção ao tipo de agente (se humano ou artificial), deve-se assumir que estamos nos referindo a agentes artificiais. Vejamos então mais algumas definições sobre os agentes inteligentes.

Agentes inteligentes são ditos *autônomos*, pois não são diretamente controlados por humanos. Podemos citar ainda outras características (opcionais) dos agentes inteligentes, como *adaptabilidade*, e ainda *personalidade* e *emoção*, características estas que buscam aumentar o grau de realismo desses agentes.

Cabe agora examinar os tipos de agentes artificiais que podemos encontrar em ambientes virtuais ou simulados. Apesar de essa ser uma área de pesquisa já bem explorada, ainda não existe uma taxonomia clara e consensual sobre o assunto. Assim

sendo, definiremos os tipos de agentes de acordo com o nosso conhecimento da área. Indicamos algumas referências importantes nas notas de rodapé.

- **Agentes inteligentes**¹³ – Já definidos anteriormente, podendo ser humanos ou artificiais.
- **Agentes virtuais** – Todos os agentes inteligentes que não são humanos.
- **Agentes conversacionais**¹⁴ – Agentes virtuais com capacidade de dialogar com o usuário através de texto ou usando processamento de voz.
- **Agentes credíveis**¹⁵ – Agentes virtuais com personalidade e emoção.
- **Agentes incorporados**¹⁶ – Agentes virtuais que têm uma representação gráfica em 2 ou 3 dimensões dentro de um ambiente virtual/simulado.
- **Agentes avatares**¹⁷ – Inicialmente, esse termo era usado apenas para se referir a agentes que representavam humanos em ambientes virtuais/simulados. Atualmente, o termo também é usado para se referir a agentes artificiais em mundos virtuais persistentes. O site [Chatbots.org, 2016] traz uma discussão muito pertinente sobre o termo e seus variados usos.
- **Agentes animados** (mais conhecidos como *Atores Sintéticos*¹⁸) – são agentes credíveis incorporados, que possuem características especiais como atitudes, emoções e personalidade. A representação gráfica desses agentes é, em geral, gerada pelo computador e renderizada em tempo real, apresentando movimentos corporais, gestos e olhares.

Por fim, vejamos alguns tipos de ACs que combinam várias características dos tipos agentes citados acima.

- **Agentes conversacionais credíveis** – ACs com personalidade e emoção.
- **Agentes conversacionais incorporados** – ACs com representação física, e geralmente com comunicação via voz.
- **ACIs credíveis** (ou **Humanos Virtuais**) – ACIs animados/atores sintéticos.

Vamos adotar essa nomenclatura ao longo deste documento, reconhecendo sempre que nem sempre há consenso com relação a essas definições.

¹³ *Intelligent Agents*. http://en.wikipedia.org/wiki/Intelligent_agent

¹⁴ *Conversational agents*. Ver https://www.chatbots.org/conversational_agent/; <http://digitalmarketing-glossary.com/What-is-Conversational-agent-definition>

¹⁵ *Believable agents*. Ver [Bates, 1994], referência clássica.

¹⁶ *Embodied agents*. Ver http://en.wikipedia.org/wiki/Embodied_agent

¹⁷ <https://www.chatbots.org/avatar/>

¹⁸ *Synthetic actors*. Ver <http://www.rodny.net/kim/SyntheticActors.htm>

4.3. Assuntos Correlatos ±*Background*

Esta seção introduz conceitos básicos indispensáveis ao bom entendimento do conteúdo deste curso. Inicialmente, a Seção 4.3.1 discute alguns conceitos linguísticos relacionados ao desenvolvimento de sistemas e agentes conversacionais. A Seção 4.3.2 trata da Computação Afetiva, uma área de estudos especialmente importante na construção de agentes artificiais (em particular, de ACIs) com traços de personalidade e emoção, buscando dar mais credibilidade a esses agentes.

4.3.1 Um pouco de Linguística ±*Análise da Conversação e Análise do Discurso*

Veremos aqui conceitos básicos de duas áreas de estudos linguísticos intimamente ligadas aos Sistemas/Agentes conversacionais: Análise da Conversação e a Análise do Discurso.

Análise da Conversação

A *Análise da Conversação* estuda a *interação social*, cobrindo comunicações verbais e não verbais em situações do dia a dia. Essa área de estudo tem grande interesse para o presente curso, pois seu arcabouço teórico é frequentemente usado como base para projetar e analisar agentes conversacionais. Ao longo do texto, veremos como generalizar os conceitos aqui apresentados, definidos para conversações entre interlocutores humanos, para cobrir também as interações com agentes conversacionais.

Os estudos iniciais na área da Análise da Conversação cobriam apenas conversações sociais, que podem ser definidas como a “troca informal de ideias” entre duas ou mais pessoas (em contraste com discursos, aulas, debates, etc.)¹⁹. Contudo, seus métodos foram mais tarde adotados para estudar também interações institucionais/formais (e.g., julgamentos em tribunais, consultas médicas), e conversações orientadas a tarefas (como ocorre nos ambientes educacionais).

Apresentamos aqui um resumo breve sobre a Análise da Conversação, a fim de facilitar o entendimento do presente curso. Segundo [Marcuschi, 1986], uma conversação se distingue de outros processos comunicativos (como discursos e aulas) por apresentar algumas características básicas. Dentre elas, destacam-se:

- 1) Interação entre pelos menos dois interlocutores;
- 2) Ocorrência de pelo menos uma contribuição de cada interlocutor;
- 3) Sincronismo – em contraste com processos comunicativos assíncronos, como diálogos via WhatsApp²⁰, por exemplo; e
- 4) Existência de um tema comum aos falantes.

Assim definida, a conversação engloba interações por telefone, via Skype²¹, e em aplicações online variadas, através das quais os usuários podem conversar com outros usuários ou com agentes conversacionais virtuais. Exemplos dessas aplicações são salas de *chat*, sistemas de EaD, SACs, jogos digitais sérios ou para simples

¹⁹ <http://www.oldandsold.com/articles10/voice-17.shtml>

²⁰ <https://www.whatsapp.com/>

²¹ <http://www.skype.com/>

entretenimento, e ainda os mundos virtuais²² na Web, geralmente desenvolvidos para jogos (games).

Apesar de parecer livre, a conversação é uma atividade estruturada, que apresenta dois níveis distintos de organização, o nível global e o nível local.

No que concerne a *organização global*, é possível identificar três fases distintas de uma conversação: abertura, desenvolvimento e fechamento. Na *abertura*, os interlocutores procuram identificar os demais participantes da conversa (ou se apresentar ao grupo, se necessário), cumprimentando-se. No *desenvolvimento*, os interlocutores conversam sobre os tópicos escolhidos, podendo ocorrer várias mudanças de tópico. No *fechamento*, em geral os usuários se despedem. Os agentes conversacionais devem ser projetados para obedecer a essa estrutura, a fim de manter conversações naturais e verossímeis com os usuários.

A *organização local* da conversação, por sua vez, tem por base elementos organizadores, dentre os quais destacam-se o *turno*, as *sequências*, a *preferência* e os *tópicos* da conversação, brevemente descritos a seguir.

Turno é definido como a produção de um falante enquanto está com a palavra, incluindo aí a possibilidade do silêncio, quando ele for significativo [Marcuschi, 1986]. Durante uma conversação, cada interlocutor deve ficar restrito a falar nos turnos a ele alocados. A *tomada de turno* é definida como o processo pelo qual os interlocutores alocam turnos de fala para si. Podemos identificar diferentes estratégias para tomada de turno, por exemplo, começar a falar assim que o outro falante fizer uma breve pausa ou completar uma oração. Claramente, a tomada de turno em conversações com mais de dois interlocutores é bem mais complexa, pois requer também determinar quem será o próximo a falar.

No caso dos agentes conversacionais textuais, em geral não há dificuldades para se sinalizar a troca de turno, pois isso fica claro quando o usuário conclui a digitação do seu texto. Porém, quando se trata de ACIs com processamento de voz, a situação é semelhante a uma conversação entre humanos. Como esperado, o problema se agrava quando lidamos com ACIs em ambientes virtuais multiusuário.

Sequências, mais conhecidas como *pares adjacentes*, estruturam as conversações de forma padronizada [Schegloff, 1972]. Cada par adjacente define uma sequência de dois turnos que coocorrem, organizando assim a conversação no nível local. Por exemplo, quando um falante cumprimenta o outro (e.g., “Bom dia!”), o segundo falante quase obrigatoriamente responde ao cumprimento. Como exemplos de pares adjacentes, citamos: {pergunta, resposta}, {cumprimento, cumprimento/silêncio}, {convite, aceitação/recusa}, {ordem, execução}.

A noção de *preferência* se reflete na escolha do segundo turno de cada par adjacente, sendo “preferidas” as escolhas mais comumente observadas nas interações sociais. Tomando como exemplo o par {convite, aceitação/recusa}, quando um interlocutor convida o outro a realizar alguma atividade, ele espera que seu convite seja aceito, sendo esta a opção socialmente *preferida* na formação desse par adjacente. Contudo, o outro interlocutor poderá também recusar o convite.

²² http://en.wikipedia.org/wiki/Virtual_world, http://pt.wikipedia.org/wiki/Mundo_virtual

Os agentes conversacionais projetados para trabalhar com pares adjacentes podem ter suas preferências determinadas a priori. Caso o agente tenha personalidade e seja capaz de demonstrar emoção, as preferências na formação dos pares adjacentes devem estar em consonâncias com seus estados emocionais e de humor (Seção 4.3.2).

É possível extrapolar a ideia dos pares adjacentes para *comunicação gestual* (não apenas verbal). Um sorriso ou um aceno podem indicar a intenção do interlocutor em cumprimentar o outro, mesmo sem pronunciar/escrever nenhuma palavra. Assim, podemos pensar em pares adjacentes “multimodais”²³, não apenas falados ou escritos. Isso é mais um recurso importante para os ACIs.

Os *tópicos* (temas) da conversação devem ser escolhidos de comum acordo entre os interlocutores, havendo uma mudança natural de tópico durante o desenrolar da interação. Note que, em conversações sociais, a *mudança de tópico* é, em geral, negociada entre os falantes a cada turno; porém deve ocorrer de forma natural, para não quebrar a fluência da conversa. Assim, podemos fazer uma distinção entre *mudança* e *quebra de tópico*, sendo esta última identificada quando ocorre uma interrupção brusca ou inesperada na conversação.

No que concerne os agentes conversacionais, identificamos ACs projetados para conversar sobre tópicos predeterminados (e.g., agentes tutores), e outros que são projetados para conversações livres (Seções 4.4 e 4.5). Em qualquer dos casos, os ACs devem ser capazes de identificar automaticamente mudanças de tópico iniciadas pelo usuário, e responder adequadamente a isso sem comprometer a fluência do diálogo.

Além desses elementos organizadores, vale ainda citar os *reparos*, definidos como mecanismos através dos quais os falantes lidam com problemas/mal entendidos ocorridos durante a conversação. Nota-se que o falante que introduziu a fonte do problema é quem busca reparar/corrigir o erro (essa é a preferência). Assim, em geral, esse falante inicia seu próximo turno com um reparo (e.g., uma correção ou uma observação), ou tenta utilizar algum silêncio do outro falante para antecipar o reparo.

Esse é um recurso muito importante nos ACs, uma vez que podem ocorrer muitos mal entendidos em conversações com esses agentes. As razões são diversas: erros do sistema ao mapear som (voz) em texto; dificuldades do sistema em interpretar o texto (devido a imperfeições nas frases do usuário, ou à falta de conhecimento do sistema), entre outras.

Análise do Discurso

A *Análise do Discurso* (AD) estuda “os princípios que governam a produção de sequências estruturadas de frases (*discurso* - escrito ou falado)” [Barros e Robin 2001]. Incluímos aqui interações faladas e/ou escritas, síncronas ou assíncronas, com um falante/escritor (e.g., monólogos, cartas, avisos) ou vários interlocutores.

Como visto anteriormente, a Análise da Conversação se ocupa de estudar a interação social síncrona entre pelo menos dois falantes. Assim, vemos que a AD se distingue da Análise da Conversação no seu foco, bem como nos métodos utilizados

²³ Multimodais – que integram mais de um modo de comunicação (escrita, fala, gestos, expressões faciais, movimentos corporais).

para estudar/processar a língua natural. A AD busca identificar como cada frase influencia na interpretação das frases subsequentes, sendo indispensável para a resolução de referências pronominais e de dêiticos (e.g., amanhã, aqui, depois, etc.). Esses processos não são cobertos pela Análise da Conversação. Veremos agora alguns conceitos básicos dessa área necessários para o bom entendimento deste curso: tópico e focos do discurso, segmentos/segmentação do discurso, e marcadores discursivos.

Em geral, cada discurso possui um tema geral/central, que motiva sua existência – e.g., ensinar a fazer um bolo, informar sobre algum acontecimento, ou apenas conversar sobre o cotidiano (caso das conversações sociais livres). Esse tema será aqui chamado de *tópico do discurso*. Além disso, os discursos apresentam um ou mais *focos*, temas secundários, porém ligados ao tópico principal, que são abordados durante o discurso (e.g., onde comprar os ingredientes, como misturá-los, como cozer o bolo, etc.). No caso das conversações sociais, a mudança de foco ocorre com maior frequência, dada a maior liberdade desse tipo de interação. Já em discurso muito curtos (como notícias curtas), pode haver apenas um foco, que coincide com o tópico geral.

Assim como as frases são estruturas com base nos sintagmas (grupos nominais, grupos verbais, etc.), o discurso é organizado em *segmentos*, que são os seus *elementos estruturantes*. Cada segmento agrupa as frases consecutivas que tratam de um mesmo foco. Um segmento longo pode ser subdividido em *subsegmentos*, formando assim uma estrutura hierárquica de segmentos “aninhados”.

Do ponto de vista prático, a segmentação do discurso auxilia na resolução de referências pronominais (e.g., ela, eles) e dêiticos (e.g., aqui, hoje). Ao tentar resolver uma referência pronominal (e.g., ela), damos preferência a candidatos (substantivos femininos no singular) que ocorrem dentro do mesmo segmento de discurso no qual o pronome ocorreu. Em seguida, investigam-se os referentes que ocorrem nos segmentos de mais alto nível, hierarquicamente superiores ao segmento onde o pronome ocorreu. O mesmo procedimento pode ser adotado para os dêiticos.

Contudo, no discurso (escrito ou falado), nem sempre encontramos um delimitador claro para indicar as fronteiras de cada segmento. Delimitamos “mentalmente” essas fronteiras com base na identificação de mudanças no foco do discurso – como dito, cada segmento se refere a um único foco de discussão. Assim, a resolução de referências pronominais e dêiticos geralmente ocorre de forma simples. Porém, a *segmentação automática* do discurso é bem mais complexa, sendo um dos maiores problemas dos sistemas para processamento do discurso.

Vale a pena observar que existem alguns fenômenos linguísticos que podem auxiliar nessa segmentação. Dentre eles, destaca-se a utilização de *marcadores discursivos* (e.g., a propósito, em seguida, ao contrário, mudando de assunto) como elementos que sinalizam manutenção ou mudança de foco. Outra dica a observar são as *repetições* no discurso. Observa-se que a manutenção do foco do discurso, em geral, vem marcada por repetições dos mesmos termos (geralmente, substantivos), ou pelo uso de pronomes (que substituem esses substantivos). Assim, o surgimento de novos termos no discurso sinaliza uma mudança de foco.

Apesar de ser uma tarefa difícil, a segmentação automática do discurso é indispensável aos sistemas conversacionais. Tal tarefa tem sido tratada de maneiras

distintas, como pode ser visto na literatura relacionada (e.g., [Grosz e Sidner, 1986], [Walker e Whittaker, 1990], [Galley et al., 2003]).

No caso dos ACs textuais (chatterbots), a resolução de referências pronominais e dêiticos pode ser mais simples, dado que os turnos são claramente marcados, e o chatterbot tem mais “tempo” para processar as informações de que dispõe a fim de encontrar os referentes corretos. Nesses casos, o agente pode até confirmar com o usuário se fez uma substituição correta. Já no caso dos ACIs com processamento de voz, essa tarefa se torna mais difícil do que no caso das conversações entre humanos. Note que tais ACIs trabalham em tempo real, e não dispõem de todo o conhecimento que um humano tem para interpretar corretamente essas referências.

Para simplificar o processamento da conversação, alguns ACs simplesmente não aceitam o uso de pronomes e dêiticos. Porém, isso não parece uma boa prática, dado que um dos objetivos desses agentes é prover uma conversação verossímil. Assim, alguns ACIs trazem um módulo específico para tratamento do diálogo, a fim de prover resolução de referências em geral (ver Seção 4.5).

4.3.2 Computação Afetiva para ACs

Com o objetivo de aumentar o grau de realismo dos agentes virtuais, pesquisadores vêm construindo modelos que buscam simular características humanas, como a personalidade e a emoção. Esses trabalhos utilizam conceitos da *Computação Afetiva*, que estuda a modelagem computacional de fenômenos afetivos e influências emocionais [Picard, 2003] [Calvo et al., 2015]. De forma geral, podemos dizer que o objetivo desses modelos é detectar, interpretar, processar e simular emoções humanas.

Segundo [Bates, 1994], que trabalha com atores sintéticos, esses agentes devem agir e reagir de modo semelhante ao que entendemos por “comportamento humano”. Essas características também são desejáveis no contexto dos agentes conversacionais, uma vez que ACIs também buscam se assemelhar aos humanos, a fim de criar empatia com os usuários, tornando a comunicação mais natural e efetiva.

Esta seção discute brevemente modelos para modelagem de personalidade e de emoção em agentes virtuais. Vale ressaltar que os modelos de personalidade e de emoção não são excludentes, podendo ser usados de maneira complementar na modelagem de agentes artificiais.

Modelagem Computacional de Personalidade

Dentro da Psicologia, entende-se que a *personalidade* é um componente mais estável no ser humano, sendo a característica mais duradoura de cada indivíduo. A personalidade em geral determina o comportamento, as atitudes e ações mais frequentes do indivíduo.

Estudos na área da Psicologia já propuseram diversas teorias para modelar personalidade, sendo a abordagem taxonômica a mais utilizada para esse fim²⁴. Essa abordagem classifica os indivíduos por meio de suas características: um indivíduo é classificado como introvertido se ele apresenta características como “quieto”, “tímido” e “reservado”.

²⁴ <http://personality-project.org/revelle.html>

Dentro dessa abordagem, a teoria de maior destaque para a Computação Afetiva é a *Teoria dos Traços* [Allport, 1938], que busca determinar quais traços de personalidade (características) podem caracterizar o comportamento de um indivíduo. A partir da *Teoria dos Traços*, foram criados modelos reduzidos que podem ser utilizados na modelagem computacional da personalidade.

Dentre os modelos criados, os que mais se destacam são o *Big Five* [Saucier e Goldberg, 1998] e o *Five Factor* [McCrae e Costa Jr., 1998]. Embora apresentem pequenas diferenças, na literatura, os dois modelos são conhecidos pelo termo *Big Five*, sem distinção. Ambos trabalham com a distribuição e redução dos traços catalogados por Allport em cinco fatores/dimensões.

Essas cinco dimensões são conhecidas pelo acrônimo OCEAN. O acrônimo vem das iniciais dos termos em inglês usados para denominar os aspectos escolhidos pelos autores como sendo determinantes na personalidade:

- *Openness* – ser curioso, ser criativo, estar aberto a viver novas experiências;
- *Consciousness* – ter responsabilidade, ser confiável, ser consciente, pensar nas consequências de seus atos antes de agir;
- *Extraversion* – ser extrovertido, ser sociável;
- *Agreeableness* – ser delicado/gentil, ter espírito de cooperação; e
- *Neuroticism* – ser ansioso, nervoso, demonstrar instabilidade emocional.

Como dito, esse modelo assume que a personalidade pode ser caracterizada a partir desses cinco aspectos, indicando-se o grau/peso de cada um deles na personalidade daquele indivíduo. Quando desejamos modelar a personalidade de agentes virtuais, é necessário inicialmente definir o tipo de personalidade desejada para o agente (amigável, extrovertido, etc.). A partir daí, devemos definir o peso de cada um dos cinco aspectos na personalidade do agente (por exemplo, se queremos modelar um agente muito introvertido, devemos associar um valor muito baixo ao aspecto *Extroversão*).

O intervalo de variação desses valores (e.g., de 0 a 5), bem como o incremento de variação (de 1 em 1; de 0,5 em 0,5; etc.) serão escolhidos por quem está construindo o agente. Se desejarmos uma modelagem mais refinada, devemos trabalhar com intervalos de variação menores. É necessário também definir a interação entre esses vários aspectos, para que o agente apresente uma personalidade credível.

Modelagem Computacional de Emoção

Voltando a atenção agora à modelagem de *emoções*, podemos ver que essa característica varia de acordo com a situação atual em que o indivíduo se encontra, podendo um indivíduo vivenciar várias emoções ao longo de um mesmo dia (ou até ao mesmo tempo – e.g., alegria e vergonha; medo e surpresa).

Estudos na área de emoção podem ser classificados em quatro perspectivas teóricas (abordagens) diferentes: Darwiniana, Jamesiana, Cognitiva e Sócio Construtivista. De acordo com [Kasap e Magnenat-Thalmann, 2007], atualmente modelos de emoção baseados na abordagem Cognitiva têm sido preferidos (em

detrimento dos modelos baseados na abordagem Darwiniana), uma vez que eles explicam melhor o processo geral de como emoções ocorrem e afetam nossa tomada de decisão. Tais modelos levam em conta a nossa “apreciação” (percepção) do ambiente, incluindo não apenas a situação atual, mas também eventos passados e nossas expectativas sobre o futuro.

No que concerne a modelagem/simulação de emoção em agentes virtuais, destaca-se o *Modelo OCC* (sigla derivada dos sobrenomes dos seus criadores - Ortony, Clore e Collins) [Ortony et al., 1988]. Esse modelo pode ser considerado como padrão para representação computacional de emoções, tendo sido usado na construção de diversos agentes virtuais [Bartneck, 2002].

Vale lembrar que as ações realizadas no ambiente (repostas do agente) são diretamente influenciadas pelos seus estados emocionais. Isso inclui a escolha das expressões faciais, dos olhares, gestos e do texto falado/escrito, bem como a coordenação entre todas essas facetas. Daí a grande importância de se modelar adequadamente essa característica nos agentes virtuais credíveis.

O modelo OCC propõe 22 categorias emocionais, tais como: raiva, satisfação, amor, vergonha, admiração, esperança, etc. Contudo, esse modelo é bastante flexível, não impondo o uso de todas essas categorias na modelagem da emoção dos agentes. Essa escolha fica a critério do desenvolvedor do agente. Além disso, é possível também definir novas categorias de emoções a partir da combinação de emoções mais básicas (e.g., desapontamento = tristeza + surpresa), uma vez que, como já dito, é bastante comum vivenciarmos mais de uma emoção ao mesmo tempo.

Como no caso do OCEAN, aqui também devemos estipular intervalos e incrementos de variação para descrever os valores que as categorias de emoção podem assumir na modelagem do agente. O ideal seria trabalhar com espectros contínuos, mas isso não é possível.

O modelo OCC não deixa explícito como a emoção é influenciada pela personalidade, dando liberdade ao projetista para criar modelos diferentes. Contudo, independente do modelo de personalidade adotado, é necessário considerar a influência dessa característica na escolha das emoções que o agente vai demonstrar a cada turno de percepção-ação. Relembramos que essa escolha deve considerar também a percepção atual do ambiente, eventos passados e os objetivos e preferências do agente.

Por fim, destacamos que alguns autores propõem ainda a modelagem do *humor*, como um complemento ao estado emocional do agente. Segundo [Wilson, 2000], humor é um estado emocional prolongado, causado pelo efeito cumulativo de emoções momentâneas, sendo uma propriedade dinâmica, que varia no tempo. Contudo, a depender da personalidade, podemos sorrir, mesmo estando de mau humor.

4.4. Agentes Conversacionais Textuais ±Chatterbots

Esta seção inicia com a apresentação de algumas competições internacionais que realizam o Teste de Turing para avaliar o desempenho de chatterbots. A seguir, temos um histórico sobre chatterbots, com foco nas técnicas utilizadas para manutenção do diálogo com o usuário. Por fim, discutiremos os problemas ainda apresentados por esses bots, a algumas direções futuras na área. Como já dito, alguns chatterbots textuais apresentam uma imagem associada, porém não são considerados agentes incorporados, por não apresentarem as outras funcionalidades que caracterizam os ACIs.

4.4.1 Prêmios e competições

Uma das competições mais conhecidas na área é o Prêmio Loebner (*Loebner Prize*)²⁵, sendo vista como “a primeira instanciação formal de um Teste de Turing”. Trata-se de uma competição anual que premia o chatterbot que conseguir “enganar” os juízes por mais tempo. A competição implementa um “Teste de Turing restrito”, que se tornou mais viável do que a proposta inicial de passar no teste de modo irrestrito. No decorrer do texto, faremos menções a chatterbots vencedores do prêmio Loebner.

Recentemente, em 2014, um Teste de Turing foi realizado como parte de um evento especial, organizado pela *Royal Society* (Grã-Bretanha) para celebrar os 60 anos da morte de Turing. Nesse evento, o chatterbot *Eugene Goostman* foi premiado e reconhecido como sendo o primeiro sistema a passar no Teste de Turing. Eugene se fez passar por um adolescente com pouco conhecimento de inglês, e assim conseguiu enganar 33% dos juízes participantes do evento (mais do que os 30% considerados suficientes para que o software seja considerado inteligente)²⁶. Destacamos ainda o *Chatterbox Challenge*²⁷, outro evento de grande visibilidade na área.

4.4.2 Histórico ±Gerações de Chatterbots

A partir de uma análise sobre a literatura relacionada, é possível dividir a história dos chatterbots em três gerações, que se diferenciam principalmente pelas técnicas utilizadas na manutenção do diálogo com o usuário [Neves, 2005].

A *primeira geração* utilizava técnicas simples de casamento de padrão [Weizenbaum, 1966]; a *segunda geração* passou a utilizar técnicas mais complexas do ponto de vista de Inteligência Artificial, como, por exemplo, redes de ativação [Mauldin, 1994]; e a *terceira geração* se apoia em técnicas mais complexas de casamento de padrão, baseadas em XML [ALICE Bot, 2016].

Primeira Geração

O primeiro chatterbot com ampla divulgação foi ELIZA²⁸ [Weizenbaum, 1966], marcando o início da primeira geração de agentes conversacionais textuais. Esse sistema simula um psicanalista da linha Rogeriana [Eliza Chat Bot, 2016].

²⁵ http://en.wikipedia.org/wiki/Loebner_Prize

²⁶ Ver http://pt.wikipedia.org/wiki/Eugene_Goostman e <http://www.zdnet.com/article/computer-chatbot-eugene-goostman-passes-the-turing-test/>

²⁷ <http://www.chatterboxchallenge.com>

²⁸ <http://en.wikipedia.org/wiki/ELIZA>

Os *bots* da primeira geração buscam manter diálogos com o usuário utilizando técnicas muito simples de casamento de padrões com base em palavras-chave, podendo também tratar frases inteiras. De modo simples, esses sistemas utilizam pares <Pergunta-Resposta> (ou mais precisamente, pares <Estímulo, Resposta>) para manter o diálogo. A cada nova intervenção do usuário, o sistema busca em sua base a entrada mais semelhante à frase do usuário, devolvendo a resposta associada à entrada selecionada. Esses sistemas apresentam apenas um módulo de processamento.

O Exemplo 4.1 ilustra a técnica de casamento de padrões utilizada por ELIZA. Apresentamos trechos de diálogos com ELIZA, incluindo o padrão de pergunta-resposta selecionado para cada caso. Os termos sublinhados indicam as palavras-chave usadas no processo de casamento de padrão. Empregamos aqui os seguintes símbolos: <I> texto de entrada do usuário; <E> padrão de entrada de um par; <R> padrão de resposta associado a essa entrada na base de pares <E,R>; <C> resposta do chatterbot.

Exemplo 4.1: Trechos de diálogo com ELIZA.

<I1> Hi!

Padrão selecionado:

{<E> Hi/Hello; <R> Hello, how are you?}

<C1> Hello, how are you?

<I2> Today I am feeling a bit sad.

{<E> (slot1) I am feeling (slot2);_ <R> Why are you feeling (slot2)?}

<C2> Why are you feeling a bit sad?

O sucesso de ELIZA abriu caminho para essa nova aplicação, e as técnicas utilizadas na sua implementação influenciam a construção desses sistemas até os dias de hoje [Eliza Chat Bot, 2016]. Outro chatterbot dessa geração que alcançou fama foi PARRY [Colby et al., 1972], que simulava um doente mental com esquizofrenia e paranoia.

Apesar do sucesso desses bots, as limitações das técnicas utilizadas não favorecem a construção de um diálogo verossímil, coeso e coerente por muito tempo. Para se obter um grau aceitável de realismo nos diálogos, é necessário construir uma base de padrões muito grande, a fim de tentar cobrir as possíveis variações de frases dos usuários. Além disso, os sistemas iniciais não armazenavam conhecimento sobre um domínio particular ou informações sobre o usuário (seu perfil), o que resultava em diálogos muito superficiais e iguais para qualquer usuário. Por fim, os sistemas iniciais não tratavam adequadamente fenômenos discursivos, como referências pronominais e repetições, uma vez que eles mantêm apenas o registro de sua última réplica (com o intuito de evitar a repetição das suas próprias respostas). Algumas dessas questões foram tratadas pelos chatterbot da geração seguinte.

Segunda Geração

A segunda geração de chatterbots (início da década de 1990) passou a incorporar técnicas mais avançadas, oriunda da IA e do PLN, a fim de aumentar o realismo dos diálogos. Os sistemas passaram a apresentar uma arquitetura mais modular, seguindo assim os avanços da Engenharia de Software.

Tais sistemas contam com um módulo de processamento morfossintático, a fim de cobrir variações morfológicas e sintáticas das frases dos usuários. Esse módulo implementa um mecanismo de normalização de palavras e frases, com base em uma tabela de substituições. Apesar de simples, essa solução conseguiu melhorar bastante a fluência dos diálogos com os chatterbots [Neves, 2005]. Além disso, alguns sistemas da dessa geração passaram a incorporar bases com conhecimento sobre tópicos específicos, seguindo também uma tendência da IA de construir Sistemas Especialistas. Por fim, tais sistemas também passaram a considerar informações referentes aos interlocutores, através da manutenção de um modelo simplificado do usuário baseado em frames.

O bot mais popular dessa geração é JULIA [Mauldin, 1994], desenvolvido para um ambiente de entretenimento virtual multiusuário de 1989 (TinyMUD²⁹). JULIA participava do ambiente como se fosse um usuário humano interagindo com outros humanos através de *salas de chat*. Para dar mais credibilidade ao diálogo, JULIA guardava um perfil do usuário com informações básicas, como nome, país e cidade de origem. Essas informações eram adquiridas pelo *bot* durante o diálogo com o usuário.

A base de diálogos de JULIA consistia em um grafo (Rede de Ativação), no qual cada nó possui um título, palavras-chave, um valor de ativação e um molde de réplica (semelhante aos padrões de ELIZA, com *slots* a serem preenchidos) – ver Exemplo 4.2. Os nós estão ligados por arestas com pesos positivos ou negativos. O valor de ativação de um nó pode ser aumentado ou diminuído durante o processamento da entrada. Ao final, o nó com maior valor de ativação é selecionado.

A fim de aumentar a credibilidade, JULIA utilizava as informações do perfil do usuário para compor suas réplicas. Assim, no exemplo abaixo, o nome do usuário faria parte da réplica do bot, e.g.: "Do you have any pets Ryan?".

Exemplo 4.2: Nó <havepet> da Rede de Ativação de JULIA.

Título	<havepet>
Valor de ativação	1.0
Palavras-chave	*pet*
Réplica associada	Do you have any pets <name>?
Nós com ligações positivas	<havepet-1> <havepet-2>
Nós com ligações negativas	<havepet-6> <havepet-7>

Assim como ELIZA, JULIA influenciou toda uma geração de chatterbots, tendo servido como ponto de partida para muitos outros bots de sucesso, tais como JFRED³⁰ (um *framework* para o desenvolvimento de chatterbots). JFRED foi desenvolvido por Robby Garner³¹, vencedor do prêmio Loebner de 1998 e 1999 com o bot Albert One³².

Apesar do avanço em relação à primeira geração, os chatterbots dessa geração ainda apresentavam problemas variados. Assim, alguns sistemas da primeira geração continuaram superando os sistemas dessa geração nos concursos do Prêmio Loebner. O problema principal desses bots é a necessidade de se refazer a Rede de Ativação a cada

²⁹ TinyMUD (Tiny Multi-User Dungeons) - Um ambiente virtual de jogo multiusuário desenvolvido por James Aspnes em 1989 - <http://www.cs.yale.edu/homes/aspnes>

³⁰ <http://www.turinghub.org/JFRED.php>

³¹ http://en.wikipedia.org/wiki/Robby_Garner

³² http://en.wikipedia.org/wiki/Albert_One

novo domínio de aplicação (uma vez que essas redes são dependentes do domínio). Outras questões são a necessidade de controlar o andamento global da conversação (início, desenvolvimento e fechamento), e de oferecer tratamento adequado a intervenções (perguntas) desconhecidas [Neves et al., 2006].

Terceira Geração

A terceira geração de chatterbots (iniciada no final da década de 1990) apresenta uma arquitetura modular e mais completa do ponto de vista do PLN, tendo como marco principal o uso de linguagens de marcação do tipo XML (*eXtensible Markup Languages*)³³.

A primeira e mais importante linguagem dessa geração é a AIML (*Artificial Intelligence Markup Language*) [Wallace, 2009]. Tal linguagem possui uma estrutura simples do ponto de vista computacional (uma vez que se inspirou nas técnicas usadas por ELIZA), porém oferece uma capacidade de processamento de diálogos bem superior às técnicas usadas nos bots das duas primeiras gerações.

AIML pode ser vista como um novo “*framework*” para o desenvolvimento de chatterbots, oferecendo tratamento de informação em quatro níveis linguísticos: morfossintático, semântico, discursivo e pragmático. Para isto, o framework utiliza diversas variáveis de controle, bem como bases de dados/conhecimento em AIML.

Nível morfossintático:

- A *Base de substituição* é usada tanto para corrigir erros de digitação (e.g., `<substitute find= "becuase" replace= "because"/>`), quanto para padronizar termos a partir de variações de digitação que não são necessariamente um erro (e.g., `<substitute find= "ohh" replace= "oh"/>`).
- A *Base de categorias* realiza o tratamento sintático da entrada do usuário. Tal base consiste em unidades identificadas pelo elemento `<category>`, compostas por um padrão de entrada associado a um ou mais moldes de réplica, semelhante aos sistemas de primeira geração (Exemplo 4.3). Os padrões de entrada podem usar caracteres especiais (e.g., * _) para casar com frases variadas digitadas pelos usuários, substituindo esses caracteres por qualquer expressão (Exemplo 4.4). Assim, se um usuário digita “*What is an iPad?*”, e o sistema não tiver informação sobre isso, o bot pode responder “*I don't know what is an iPad*”.

Exemplo 4.3: Modelo geral de categoria AIML.

```
<category>
<pattern>padrão de entrada</pattern>
<template>
  <random>
    <li>molde de réplica 01
    <li> molde de réplica n
  </random>
</template>
</category>
```

³³ <https://pt.wikipedia.org/wiki/XML>

Exemplo 4.4: & DWHJRULD \$,0/ FRP 3 7

```
<category>
<pattern>what is *</pattern>
<template>I don't know what is <star>.</template>
</category>
```

Nível semântico:

- A *Base de descrição do Chatterbot* é formada por uma série de elementos `<property>`, que descrevem as características de cada bot em particular (e.g., `<property name="name" value="ALICE"/>`; `<property name="gender" value="female"/>`; `<property name="birthday" value="1995"/>`, etc.).
- A *Base de Tópicos* é formada por categorias classificadas por tópico. Essas são categorias mais específicas, que carregam informação semântica. Essa base pode ter, por exemplo, um conjunto de categorias específicas associadas ao tópico "LOVE", que só são utilizadas quando o tópico vigente do diálogo for "LOVE".

Nível discursivo:

- A *Base de substituição de pronomes* é semelhante à Base de substituição morfológica, porém os elementos `<substitute>` estão vinculados a um elemento `<person>`, indicando que essas substituições são pronominais (e.g., `<person>` `<substitute find=" he was " replace=" I was "/>`; ... `<substitute find=" to you " replace=" to me "/>`; ...; `</person>`).
- As *Variáveis de controle de dêiticos* (e.g., aqui, hoje, mais tarde, etc.) são definidas em tempo de execução. Essas variáveis são definidas pelo elemento `<set>` e recuperadas pelo elemento `<get>`, inseridos nos moldes de réplica das categorias AIML.

Nível pragmático:

- As *Variáveis com informação sobre o usuário* (seu perfil) são definidas da mesma forma que as variáveis de controle de dêiticos, e incluem informações sobre, por exemplo, o nome e gênero do interlocutor.

AIML foi usada na implementação de ALICE [ALICE Bot, 2016], que conseguiu destaque internacional ao vencer três edições do *Prêmio Loebner*, bem como uma edição do *Chatterbox Challenge*. Devido ao seu grande sucesso, ALICE foi a responsável pela difusão de AIML, que se tornou o estado da arte em tecnologia para criação de chatterbots. AIML já deu origem a centenas de chatterbots em línguas e domínio diversos³⁴.

Por fim, vale citar aqui um trabalho que estendeu AIML padrão, incorporando informações de intencionalidade (no nível pragmático), a fim de evitar a quebra de fluência do diálogo e a perda de credibilidade do chatterbot [Neves, 2005]. O autor propôs a incorporação de novas *tags* de intenção ao AIML padrão, tendo por base a Análise da Conversação [Marcuschi, 1986].

³⁴ AIML bots - <http://www.alicebot.org/aimlbots.html>

4.4.2 Aplicações Atuais, Limitações e Tendências Futuras

Esta seção conclui a discussão sobre agentes conversacionais chatterbots apresentando brevemente suas limitações e tendências futuras.

Aplicações Atuais

Apesar das suas limitações do ponto de vista do PLN, os chatterbots são utilizados por milhares de usuários na Web, não só para manter diálogos genéricos, mas também para intermediar o acesso a outros sistemas [Laven, 2016], podendo ainda desempenhar o papel de “acompanhantes digitais” do usuário em sistemas imersivos.

As aplicações de chatterbots são bastante variadas, podendo-se encontrar tais agentes como aplicações isoladas (*stand-alone*) na Internet para simples conversação (e.g., Eliza e ALICE), já bastante discutidos neste documento, ou embutidos em sistemas interativos variados (algumas vezes servindo como interface de acesso ou como acompanhante digital do usuário). Neste segundo caso, citamos: Chatterbots assistentes virtuais³⁵ (incluindo aqui bots do tipo FAQ - *Frequent Asked Questions* e SAC - Sistemas de Atendimento ao Cliente), bots em jogos digitais (incluindo jogos sérios), Chatterbots em sistemas de ensino/educação³⁶, chatterbots interface de acesso a outros sistemas computacionais, entre outras.

As referências a seguir, algumas já indicadas anteriormente, oferecem uma grande quantidade e variedade de exemplos de chatterbots para os mais variados fins: [Laven, 2016], [ALICE Bot, 2016], [Chatbots.org, 2016].

Por fim, ressaltamos que existem várias empresas dedicadas à criação de chatterbots³⁷. Algumas oferecem plataformas gratuitas para que o próprio usuário desenvolva seu bot (e.g., Pandora Bots³⁸).

Limitações e Estratégias simples para manutenção dos diálogos

Apesar do seu grande sucesso, chatterbots apresentam limitações na manutenção de diálogos fluentes e verossímeis, como já dito. Essas limitações se devem principalmente à grande dificuldade em se construir um sistema completo de processamento de linguagem natural (que realize tanto interpretação como geração de LN), principalmente nas etapas de análise semântica e pragmática, que envolvem interpretação de sentido/significado e intenção/intensionalidade. Assim, esses bots incorrem em erros, tendo dificuldade para manter diálogos coerentes.

Algumas estratégias simples e bem-humoradas para driblar essas situações são apresentadas no texto online “10 Truques que Chatbots usam para fazer você acreditar que eles são humanos”³⁹. Exemplos desses truques são: escreva uma frase completa a cada turno, mantenha o turno de fala (usando respostas como “Umm...”, que dão a impressão de que o bot está pensando), limpe a tela a cada resposta (para o interlocutor

³⁵ https://www.chatbots.org/virtual_assistant/

³⁶ https://www.chatbots.org/industry/learn_lookup/

³⁷ Empresas desenvolvedoras de chatterbots: <https://www.chatbots.org/companies/>

³⁸ <http://www.pandorabots.com/>

³⁹ 10 Tricks That Chatbots Use to Make You Believe They're Human. <http://io9.com/5901579/10-tricks-that-chatbots-use-to-make-you-believe-theyre-human>.

não ter a memória do diálogo, e então não perceber as quebras de coesão), mude de assunto com frequência (como acontece em diálogos reais), simule uma briga (isso não exige memória!), entre outros.

Além das dificuldades inerentes ao PLN, restam ainda as dificuldades relacionadas à criação de Bases de Conhecimento (BC) com informação pertinente ao domínio da aplicação, um problema geral da IA baseada em conhecimento. Por exemplo, se queremos construir um chatterbot para um sistema de EaD sobre História do Brasil, esse bot precisa de uma base de conhecimento que seja o mais completa possível, e que seja sempre mantida atualizada.

Quando o bot não encontra na BC a informação necessária para responder ao usuário, ele terá que contornar a situação, a fim de não interromper o diálogo. Na maioria das aplicações vistas, em caso de falta de informação na BC, os bots simplesmente oferecem links na Web relacionados à informação solicitada.

Tendências Futuras

Chatterbots certamente continuarão a existir tanto como aplicações *stand-alone* como fazendo parte dos sistemas interativos do futuro (funcionando como interfaces), provendo acesso em linguagem natural aos sistemas de computação em geral. Contudo, em sua maioria, tais sistemas irão requerer comunicação via voz (em lugar de texto digitado). Para isto, é necessário utilizar sintetizadores e interpretadores de voz, capazes de intermediar a comunicação entre chatterbot e usuário. Um exemplo desses sistemas é o Siri do iPhone⁴⁰.

Por fim, destacamos uma evidência de outra ordem: chatterbots *stand-alone* puramente textuais estão perdendo espaço nos dias atuais, com a crescente tendência de se desenvolverem ambientes virtuais imersivos, baseados em realidade virtual. Nesses casos, o usuário espera que o agente conversacional faça parte do ambiente simulado, possibilitando diálogos via voz, a fim de tornar a interação mais natural e realista. Claramente, quando o diálogo é realizado somente pelo meio textual, podem-se perder aspectos importantes da interação, fazendo com que o significado da mensagem seja mal interpretado.

Assim, vemos como tendência futura o crescimento dos agentes conversacionais incorporados com comunicação via voz, que oferecem, além das capacidades dialógicas, também uma imagem associada, ampliando assim sua esfera de uso. Esse será o tema da próxima seção deste documento.

⁴⁰ <https://www.apple.com/ios/siri/> & [http://pt.wikipedia.org/wiki/Siri_\(software\)](http://pt.wikipedia.org/wiki/Siri_(software))

4.5. Agentes Conversacionais Incorporados

Como já discutido, apesar do seu grande sucesso, chatterbots têm uso limitado em sistemas imersivos. Assim, voltamos agora atenção para os *Agentes Conversacionais Incorporados*. ACIs são agentes que possuem uma representação gráfica em 2D ou 3D (em geral, com aparência humana), sendo capazes de realizar diálogos por meio de texto, de fala e de outras modalidades não verbais, como olhares e gestos. Assim como os ACs textuais, encontramos ACIs na Web ou como aplicações *stand-alone*. Em geral, ACIs são usados em aplicações online, para prover serviços personalizados, como assistentes pessoais, ou ainda para auxiliar usuários a navegar em sites.

Alguns autores definem os ACIs como “Agentes Interface” (*Interface Agents*). Por exemplo, [Cassell e Vilhjálmsson, 1999] descrevem os ACIs como “interfaces multimodais” que implementam, tanto quanto possível, todos os elementos que os humanos usam para se comunicar. Incluem-se aqui fala, gestos e movimentos corporais, expressões faciais, linguagem corporal, e respostas com entonação baseada em emoção. Segundo [Cassell e Vilhjálmsson, 1999], ACIs podem ser definidos como interfaces de computador que conseguem lidar bem com as dificuldades de uma conversação, elaborando suas “falas” em função das exigências do diálogo, considerando emoção, personalidade e algumas características das conversações sociais.

Há também autores que nomeiam alguns ACIs como “Humanos Virtuais” (do inglês, *Virtual Humans*), principalmente quando eles incluem capacidades mais avançadas de raciocínio e demonstram emoção (e.g, [Egges et al., 2004]). Assim, optamos por não excluir nenhum trabalho apenas com base na terminologia utilizada para nomear os agentes, e sim com base na descrição do agente.

A seguir, a Seção 4.5.1 traz um breve histórico da área. Mudando o foco para questões mais técnicas, a Seção 4.5.2 apresenta uma arquitetura genérica para construção de ACIs, discutindo detalhes sobre seus componentes. A Seção 4.5.3 discute algumas abordagens adotadas na implementação de desses agentes. A Seção 4.5.4 traz aplicações atuais, e a Seção 4.5.5 apresenta limitações e tendências futuras na área.

4.5.1 Breve Histórico

No final da década de 1990, em paralelo ao ainda intenso trabalho no campo da construção de diálogos coerentes para chatterbots, observamos os primeiros esforços na construção de ACs incorporados. Os primeiros pesquisadores a se destacarem nessa área foram Justine Cassell⁴¹ (um dos seus maiores expoentes), que iniciou seus trabalhos com *agentes conversacionais autônomos animados* [Cassel et al., 1994], e Joseph Bates, que trabalhava na modelagem de emoção para agentes credíveis [Bates, 1994].

O termo “Agente Conversacional Incorporado” (do inglês, *Embodied Conversational Agent*⁴²) foi usado pela primeira vez em 1998, no *First Workshop on Embodied Conversational Characters*⁴³. O foco inicial das pesquisas em ACIs foi na *modelagem da comunicação entre humanos*, uma vez que essa é a atividade central dos

⁴¹ <http://www.justinecassell.com/>

⁴² https://www.chatbots.org/embodied_conversational_agent/

⁴³ <http://www.sigchi.org/chi98/events.html>

ACIs. Os pesquisadores buscavam definir quais aspectos da comunicação entre humanos poderiam ser simulados através dos ACIs, a fim de torná-los mais credíveis, e quais seriam as áreas de pesquisa relacionadas a essa simulação.

Segundo [Cassell et al., 2000a], tais agentes devem ter capacidade de: reconhecer entradas verbais e não verbais; responder com saídas verbais e não verbais; lidar com elementos organizadores da conversação, tais como tomada de turno e reparo (Seção 4.3.1); reconhecer a fase da conversação; e contribuir com o discurso gerando novas proposições (não apenas respostas), sendo assim um agente proativo.

Contudo, apenas estudos na área de comunicação (interpretação/geração de texto e processamento de voz) não eram suficientes para prover todas as características “humanas” aos ACIs. Assim, observamos o surgimento de grupos multidisciplinares, capazes de integrar diferentes tecnologias a fim de criar ACIs mais complexos e multifacetados. Destacam-se aqui o *MIT Media Lab* (em particular, o *Gesture and Narrative Language Group*⁴⁴); e o *Information Science Institute (ISI)*⁴⁵ da *University of Southern California (USC)*.

A partir daí, a pesquisa passou a priorizar a construção de agentes animados, com representação visual em 3D, capazes de observar seu contexto, e ainda de exibir emoções que mudariam de acordo com o contexto do diálogo. Um dos primeiros ACIs desenvolvidos no ISI foi STEVE (*Soar Training Expert for Virtual Environments*)⁴⁶, um agente tutor que ensinava, interativamente, como operar um painel de controle de navios [Rickel e Johnson, 1999]. STEVE se comunicava com o usuário através de voz sintetizada e possuía imagem em 3D. O ACI inicial foi expandido e usado em um projeto de construção de ambientes mais imersivos para treinamento militar.

Por volta de 2000, a USC criou o *Institute for Creative Technologies (ICT)*⁴⁷, que passou a abrigar muitos dos pesquisadores do ISI. No ICT, esses pesquisadores desenvolveram muitos outros agentes pedagógicos (tutores) que ajudavam usuários a desenvolver habilidades em análise de tarefas, negociação, tomada de decisão e lideranças complexas. Tais agentes eram capazes de compreender entradas escritas ou faladas, apresentando uma representação visual que incluía linguagem corporal, expressões faciais, olhares variados, entre outras habilidades [Gratch et al., 2002].

Esses agentes já apresentavam uma modelagem de emoções sofisticada, sendo capazes de demonstrar raiva, e até de se recusar a continuar interagindo com o usuário. Para tanto, eles necessitavam manter uma memória com conhecimento sobre seu ambiente e sobre os outros participantes do ambiente (virtuais ou humanos). Atualmente, o ICT conta com vários protótipos de Humanos Virtuais⁴⁸, a maioria deles voltada a aplicações militares de ajuda, havendo também alguns agentes tutores.

Em paralelo aos trabalhos do ISI e do ICT, o *Gesture and Narrative Language Group (GNL)* do *MIT Media Lab* também contribuiu muito para o desenvolvimento da área. Esse grupo, extinto em 2003, era liderado por Justine Cassell - atualmente

⁴⁴ GNL - <http://www.media.mit.edu/gnl/>

⁴⁵ ISI - <http://www.isi.edu/>

⁴⁶ STEVE - <http://www.isi.edu/isd/VET/steve-demo.html>

⁴⁷ ICT - <http://ict.usc.edu/>

⁴⁸ Protótipos do ICT - <http://ict.usc.edu/prototypes/>

trabalhando na *Carnegie Mellon University*⁴⁹. Apesar de extinto, o site do GNL ainda guarda uma lista dos projetos desenvolvidos na época⁵⁰.

Os pesquisadores do GNL dividiram o trabalho realizado pelo grupo em três gerações de humanoides (ou humanos virtuais)⁵¹. A primeira geração tem como representante o sistema *Animated Conversation*⁵², no qual dois caracteres autônomos animados conversam entre si. Apesar de não haver nenhum humano participando do diálogo, esse foi o primeiro sistema capaz de gerar automaticamente comunicação na forma verbal (entonação e fala) e não verbal (gestos com as mãos e expressões faciais).

A segunda geração é representada por *Gandalf*⁵³, um *cartoon* animado (representado apenas pelo seu rosto) que era capaz de responder perguntas sobre o sistema solar usando processamento de voz. Gandalf também era capaz de perceber movimentos a partir de um sistema eletromagnético de rastreamento usado pelo usuário. Assim, ele era capaz de responder também a comportamentos não verbais. Apesar de Gandalf operar em tempo real, suas saídas (*outputs*) eram selecionadas a partir de uma biblioteca de respostas prontas (não sendo, portanto, muito flexível).

A terceira geração de humanoides passou a apresentar um corpo totalmente articulado (com maior flexibilidade de movimentos), sendo também capaz de perceber o usuário através de câmeras. Dentre eles, destaca-se REA, certamente o ACI mais conhecido do grupo [Cassel et al., 1999], e que serviu de base para o desenvolvimento de muitos outros ACIs.

REA (*Real Estate Agent*)⁵⁴ desempenha o papel de uma corretora de imóveis que mostra propriedades virtuais aos usuários e tenta vender uma casa. Esse domínio foi escolhido por possibilitar conversações tanto orientadas a tarefas (e.g., vender imóveis), como conversações sociais. Ao contrário de Gandalf, REA sintetiza suas respostas em tempo real (incluindo aqui fala e gestos com a mão), levando em conta o contexto comunicativo. Para tanto, o sistema utiliza uma gramática e um léxico, contando ainda com um modelo simples de discurso para determinar atos de fala [Searle, 1976], bem como para resolver e gerar referências pronominais.

Atualmente, encontramos no *MIT Media Lab* dois grupos de pesquisa que mantêm alguns projetos relacionados ao desenvolvimento de humanos virtuais/ACIs: o grupo *Personal Robots*⁵⁵ e o grupo *Affective Computing*⁵⁶.

Observando os trabalhos desenvolvidos na última década, nota-se a preocupação em prover uma sincronização mais realista entre som, gestos e olhares, bem como com a coerência do comportamento do agente em relação ao contexto do diálogo com o usuário. Além disso, tais agentes devem incorporar características como personalidade e emoção. Alguns desses trabalhos mais recentes serão mencionados nas Seções 4.5.3 e 4.5.4.

⁴⁹ <http://www.justinecassell.com/>

⁵⁰ <http://www.media.mit.edu/gnl/projects.html>

⁵¹ <http://www.media.mit.edu/gnl/projects/humanoid/>

⁵² *Animated Conversation* - <http://www.media.mit.edu/gnl/projects/animated/>

⁵³ *Gandalf* - <http://alumni.media.mit.edu/~kris/gandalf.html>

⁵⁴ *REA* - <http://www.media.mit.edu/gnl/projects/humanoid/>

⁵⁵ <http://www.media.mit.edu/research/groups/personal-robots>

⁵⁶ <http://www.media.mit.edu/research/groups/affective-computing>

Por fim, destacamos um trabalho recente, que surgiu da demanda por mais realismo nas aplicações em tempo real com imersão: os *ACIs Avatares* (ACIAs) [Morie et al., 2013]. Tais agentes, especialmente projetados para ambientes (mundos) virtuais persistentes, são incorporados em avatares 3D - o mesmo tipo de representação que os participantes humanos usam-, estando portanto sujeitos às mesmas regras e restrições (de interação, de design, etc.) utilizadas na construção dos avatares de usuários humanos desses ambientes virtuais. Note que esses ACIs avatares são diferentes dos apresentados por [Cassell e Vilhjálmsson, 1999], que são avatares representando usuários em ambientes virtuais multiusuários.

Os ACIAs devem ser capazes de se adaptar a ambientes em mudança constante, bem como de reconhecer os usuários do mundo virtual, incorporados em uma grande variedade de avatares. Além disso, eles devem ser persistentes em um mundo que, a princípio, nunca é “desligado”. Além disso, caso o mundo virtual sofra uma interrupção e seja reiniciado (um *reboot*), o ACIA deve ser capaz de se reiniciar automaticamente.

4.5.2 Modelando/Projetando ACIs

Esta seção apresenta uma arquitetura genérica para ACIs, e discute alguns aspectos centrais no *design* desses agentes. Segundo [Kasap e Magnenat-Thalmann, 2007], esses aspectos são: a forma de *interação* com o agente, a sua *autonomia* para controlar dinamicamente o andamento do diálogo, e a *personificação* do agente, através da inclusão de características humanas, como personalidade e emoção. O objetivo central de cobrir todos esses aspectos é simular a experiência da conversação face a face entre humanos, buscando estabelecer um melhor engajamento do usuário no diálogo.⁵⁷

4.5.2.1 Arquitetura genérica

Com o crescimento das aplicações em tempo real, como *games* e interfaces, *comportamento autônomo* tornou-se uma característica vital para agentes virtuais - em particular, para os ACs (incorporados ou não), que devem ser capazes de dialogar de forma natural e coerente com seus interlocutores (usuários do sistema).

Contudo, prover comportamento autônomo a ACIs é uma tarefa complexa, por envolver não apenas a escolha do conteúdo (textual ou falado) a ser apresentado ao usuário, como também o tom de voz, e por fim a imagem (movimentos, olhares, expressões faciais), que devem estar sincronizados e devem ser coerentes com o contexto do diálogo em curso. Deve-se ainda considerar o estado emocional do agente em cada turno do diálogo.

Voltando às definições vistas na Seção 4.2.2, relembramos que agentes inteligentes funcionam em ciclos “percepção-raciocínio-ação”, definindo seu comportamento com base nas percepções oriundas do ambiente em que estão imersos. Mais detalhadamente, os agentes autônomos devem ser capazes de:

- *Perceber* seu ambiente através de sensores (e.g., teclado do computador, câmeras, microfone, protocolos de comunicação em sistemas multiagentes);

⁵⁷ Note que esses autores trabalham com “humanos virtuais inteligentes”, que são em geral mais complexos do que os ACIs.

- *Raciocinar*, utilizando as percepções do ambiente, conhecimento do domínio e regras, para escolher a melhor ação possível a realizar no ambiente naquele momento; e
- *Agir* sobre esse ambiente através de atuadores (*e.g.*, tela do computador, autofalante, funções internas que enviam mensagens a outros agentes, etc.).

Assim, uma arquitetura genérica para modelar agentes inteligentes deve contar com três módulos básicos, um para cada “tarefa macro” (perceber, raciocinar e agir). Contudo, observamos uma modularização mais refinada nas arquiteturas propostas para ACIs, que geralmente dividem esses três módulos básicos em componentes menores e mais específicos. Isso se deve à complexidade de modelar e construir ACIs.

De acordo com [Kasap e Magnenat-Thalmann, 2007], a maioria dos trabalhos separa a modelagem da “mente” do agente da modelagem das suas ações, uma vez que a mente humana trabalha com objetivos de mais alto nível (como “ir ao supermercado” para então comprar os produtos desejados), em lugar de planejar detalhes de movimento como “vou dar um passo com meu pé esquerdo”. Isso justifica as arquiteturas modulares usadas para a construção desses agentes.

A arquitetura de alto nível apresentada a seguir foi inspirada em [Kenny et al., 2007], que trabalha com humanos virtuais, contando com 7 componentes:

1. *Reconhecimento de fala* - produz texto a partir da entrada de voz do usuário;
2. *Reconhecimento de imagens* - produz uma representação interna da percepção visual do agente (gestos, expressões do usuário);
3. *Interpretação de linguagem natural* - constrói uma representação semântica interna a partir do texto da entrada;
4. *Raciocínio*⁵⁸ - raciocina com base nas percepções atuais usando regras/planos (base de conhecimento), e gera ações (na forma de representação interna do texto e da imagem de saída).
 - Em geral, os agentes têm um *modelo do mundo* (memória), para guardar os estados do ambiente, bem como as percepções e ações anteriores;
 - Agentes mais sofisticados contêm um *modelo do discurso* ou *modelo de diálogos orientados a tarefas* (para ACIs orientados a tarefa), *modelo do usuário*, e *modelo de emoção/personalidade*.
5. *Geração de comportamento não verbal* - seleciona gestos, posturas e olhares para o agente com base no texto de resposta, no histórico do diálogo e na emoção do agente;
6. *Síntese de voz* – mapeia o texto de saída do agente para som (voz sintetizada);
7. *Animação* – gera a ação/reação de saída do ACI em resposta às percepções atuais, sincronizando a voz com os gestos e os olhares selecionados previamente.

Observe que, como essa arquitetura geral foi proposta para Humanos Virtuais, nem todos os ACIs implementam todos os componentes citados aqui. O tipo de

⁵⁸ [Kenny et al., 2007] nomeiam esse módulo como “Agente Inteligente”.

interação e o tipo de comportamento escolhidos para o ACI vão influenciar diretamente na escolha da arquitetura ideal para cada ACI.

Antes de avançar na apresentação dos componentes dessa arquitetura, consideramos importante discutir sobre formalismos usados para representação interna das percepções e ações do agente (fala/texto, gestos, movimentos, e emoção do usuário). Note que essas representações internas são utilizadas na passagem de informação entre os componentes da arquitetura geral.

4.5.2.2 XML como padrão para comunicação multimodal

Seguindo a tendência iniciada com os chatterbots (e.g., ALICE com AIML), os ACIs atuais também adotam linguagens do tipo XML como representação interna do conhecimento relacionado ao diálogo multimodal - percepções e ações do agente, bem como para representar a base de padrões de entrada e de saída desses agentes.

Como visto anteriormente, o texto de entrada, produzido pelo componente (1), é mapeado em uma representação interna pelo componente (3). Os padrões de entrada da base são então utilizados pelos componentes (2) e (3), a fim de gerar uma representação interna da entrada completa do usuário (texto e dados multimodais). Já os padrões de saída são usados na composição da resposta do agente para o turno atual do diálogo.

Podemos citar aqui diversas linguagens tipo XML usadas na construção de ACIs, com destaque para o XML usado pela ferramenta BEAT [Cassel et al., 2001], e para as linguagens *Function Markup Language* (FML) e *Behavior Markup Language* (BML), utilizadas pelo framework SAIBA (ver detalhes na Seção 4.5.3.3). Existem também linguagens específicas para representar comunicação não verbal através de *gestos*, por exemplo, a SiGML (*Signing Gesture Markup Language*)⁵⁹, uma linguagem de marcação para gestos do agente.

A seguir, temos um exemplo de saída em XML gerada pela ferramenta BEAT [Cassel ET AL., 2001].

Exemplo 4. ([HPSOR GH VDtGD B B SdneRkrm bf pWiRuál actor´

```
<AnimEvent: GAZE w=1 t=0.0spec=AWAY_FROM_HEARER>
<AnimEvent: GAZE w=3 t=0.517 spec=TOWARDS_HEARER>
<AnimEvent: R_GESTURE_START w=3 t=0.517 spec=BEAT>
<AnimEvent: EYEBROWS_START w=3 t=0.517 spec=null>
<AnimEvent: L_GESTURE_START w=7 t=1.338 spec=ICONIC VIRTUAL >
<AnimEvent: R_GESTURE_START w=7 t=1.338 spec=ICONIC VIRTUAL >
<AnimEvent: EYEBROWS_START w=7 t=1.338 spec=null>
<AnimEvent: L_GESTURE_END w=9 t=2.249 spec=null>
<AnimEvent: R_GESTURE_END w=9 t=2.249 spec=null>
<AnimEvent: EYEBROWS_END w=9 t=2.249 spec=null>
```

Esse exemplo traz a definição do comportamento do agente em relação a cada modo de interação: *gaze* = olhar, *gesture* = gesto, *eyebrows* = posição das sobrancelhas. Os códigos que aparecem nas definições são ponteiros para o banco de dados que

⁵⁹ SiGML - http://www.visicast.cmp.uea.ac.uk/eSIGN/Images/Flyer_SiGMLSigning.pdf
<http://jeroenarendsen.nl/2007/01/esign-editor-turns-hamnosys-into-sigml-for-signing-avatars/>

armazena a representação de cada elemento visual. A síntese da fala é realizada em outra etapa de processamento.

Gostaríamos ainda de apresentar brevemente uma iniciativa do consórcio W3C na área de interação multimodal. Trata-se do trabalho do *W3C Multimodal Interaction Working Group*⁶⁰, cujo objetivo central é propor e desenvolver padrões que possibilitem estender a Web, para permitir *interação multimodal* (e.g., voz, texto digitado, caneta, visão, gestos, interfaces gráficas, interfaces hápticas, etc.) *de modo amplo* (todos, em qualquer lugar, a qualquer hora, usando qualquer dispositivo), com serviços que se adaptem ao dispositivo, ao usuário e às condições do ambiente.

Destacamos aqui a *Arquitetura Multimodal* proposta pelo grupo, uma arquitetura genérica para interfaces multimodais que pode ser estendida à medida que novos modos de acesso sejam criados/disponibilizados. Essa arquitetura trabalha com dados representados usando linguagens do tipo XML:

- *EMMA - Extended Multimodal Annotation Language*, utilizada para representar informação multimodal automaticamente extraída da entrada do usuário;
- *InkML - Ink Markup Language*, projetada para lidar com escrita cursiva, gestos, desenhos, notação musical, entre outras linguagens notacionais; e
- *EmotionML - Emotion Markup Language*, utilizada para representar emoção e outros estados afetivos.

Apesar de essa iniciativa não ser voltada ao desenvolvimento de ACIs, e sim a interfaces multimodais em geral, ele é relevante para nós, uma vez que trata do estabelecimento de padrões para comunicação multimodal. Além do seu valor técnico, esse trabalho aponta para uma clara tendência de se trabalhar cada vez mais com linguagens do tipo XML.

Passaremos agora a discutir os componentes da arquitetura genérica para ACIs.

4.5.2.3 Componentes de interação com o ambiente

Observando a arquitetura genérica descrita na Seção 4.5.2.1, vemos que os componentes (1) e (2) são responsáveis pela percepção do agente, enquanto os componentes (6) e (7) são responsáveis pela realização de ações no ambiente. De acordo com [Pauchet e Sabouret, 2012], a *interação* entre usuário e ACI pode ocorrer de três maneiras:

- *Percepção com múltiplos inputs*: textual (via caixas de texto), fala (via microfone), sensores de movimento (via câmera), etc.
- *Ação com múltiplos outputs*: textual, áudio, expressões faciais, movimentos corporais, iluminação da cena, etc.
- *Percepção e ação multimodal*: incluindo conversação via fala, e podendo incluir percepção do estado do ambiente e do usuário, tratamento de emoção, etc.

A escolha de quais componentes serão implementados no ACI vai depender diretamente do tipo de interação desejada.

⁶⁰ W3C MMI Group - <http://www.w3.org/2011/03/mmi-charter>

Não será possível discutirmos aqui as técnicas utilizadas pelos componentes de interação com o ambiente, nem os principais problemas enfrentados pelas áreas de pesquisa relacionadas com processamento (interpretação e síntese) de voz e imagem, uma vez que as autoras do curso não trabalham nessas áreas de pesquisa. Para os interessados nesses temas, indicamos como ponto de partida o trabalho de [Kasap e Magnenat-Thalmann, 2007], que traz uma ampla revisão do estado da arte sobre Humanos Virtuais. Apesar de não ser uma publicação muito recente, essa foi a única revisão realmente abrangente que encontramos sobre o assunto. As outras publicações tendem a relatar pesquisas particulares, com uma seção sobre trabalhos relacionados muito reduzida e focada no tema central da pesquisa sendo relatada.

Kasap e Magnenat-Thalmann (2007) discutem, na seção 2.1 do seu artigo, a modelagem da *percepção visual* do agente como um todo, indicando referências na área. A seção 3 desse mesmo artigo foca especificamente em ACIs, apresentando técnicas e trabalhos sobre modelagem de *expressões faciais* do ACI, e trabalhos sobre comunicação não verbal do ACI através de *gestos*. Por fim, a seção trata ainda da sincronização de apresentações multimodais (movimentos, expressões faciais, olhares, fala), destacando a SiGML, já mencionada aqui (Seção 4.5.2.2). Veremos mais tarde que alguns ACIs guardam apenas o padrão textual de saída, gerando os dados multimodais correspondentes (voz e imagem) em tempo real, enquanto outros ACIs mantêm registros completos com texto e os dados multimodais correspondentes.

4.5.2.4 Componentes de Raciocínio

Os componentes (4) e (5) da arquitetura genérica apresentada aqui são responsáveis por determinar as ações que o ACI deve realizar a cada turno de percepção-raciocínio-ação, levando em conta as percepções atuais do agente, os seus objetivos (indispensável para diálogos orientados a tarefa), e o conhecimento disponível através dos modelos mantidos pelo agente (modelo do mundo, modelo do discurso, modelo do usuário e modelo de emoção/personalidade).

Veremos mais tarde que algumas implementações de ACIs aglutinam esses componentes em um só, enquanto que outras implementações desmembram esses componentes em vários módulos específicos, sendo alguns dedicados a gerenciar os modelos mantidos pelo agente durante a interação com o usuário (Seção 5.3). Por serem de natureza distinta, cada um desses módulos pode utilizar abordagens diferentes de representação do conhecimento (e.g. lógica, objetos) e raciocínio.

A partir da literatura relacionada, podemos identificar diferentes abordagens para a construção dos componentes de tomada de decisão de ACIs/humanos virtuais (e.g., raciocínio simbólico baseado em regras, aprendizagem de máquina, abordagem estatística) [Russel e Norvig, 2010].

Ao examinar as implementações de ACIs, nota-se uma predominância da abordagem baseada em IA simbólica, na qual o agente “raciocina” aplicando regras predefinidas ao conhecimento disponível [Kasap e Magnenat-Thalmann, 2007]. Contudo, já existem algumas implementações de ACIs baseadas em Aprendizagem de Máquina (AM), tema que será brevemente discutido no final desta seção.

Raciocinando com base em Regras e Conhecimento

Observando os modelos e as implementações de ACIs existentes, podemos distinguir a utilização predominante de três abordagens de raciocínio simbólico: a abordagem deliberativa, a reativa, e a abordagem baseada em objetivos. Vale ressaltar que essas abordagens não são excludentes, como veremos na Seção 4.5.3, que traz um modelo de ACI que combina as duas primeiras abordagens aqui citadas.

Inicialmente, ressaltamos que as arquiteturas mencionadas nesta seção se referem apenas ao componente de raciocínio do ACI⁶¹, diferentemente da arquitetura geral descrita na Seção 4.5.2.1. Contudo, os pesquisadores da área denominam esses modelos como “arquiteturas”. Assim, optamos por manter a nomenclatura utilizada na literatura relacionada.

Como representante principal da *abordagem deliberativa*, temos a arquitetura SOAR⁶² (um acrônimo para *State, Operator And Result*), baseada em regras de produção. Essa arquitetura foi lançada em 1983, e alcançou grande sucesso, encontrando-se atualmente na sua versão 9. Um dos primeiros agentes construídos com base na Soar foi STEVE, já citado na Seção 4.5.1. O site oficial do projeto Soar⁶³ apresenta a evolução dessa arquitetura, que passou a incorporar também aprendizagem baseada em reforço e modelagem de emoção.

Apesar do sucesso das versões iniciais da Soar, a abordagem deliberativa não se mostrou adequada para sistemas em tempo real. Surgiram então arquiteturas com foco nos *sistemas reativos dinâmicos*, tendo como representante principal a *arquitetura de subordinação* de Brooks [Brooks, 1991]. Essa arquitetura, voltada aos sistemas multiagentes, mantém uma hierarquia de subordinação que define a precedência de cada agente em relação aos outros. Os comportamentos dos agentes são representados dentro dessa hierarquia com estruturas simples (semelhantes a estruturas *IF-THEN*). Além disso, existem soluções baseadas em Máquinas de Estado Finito, bem como abordagens baseadas em Probabilidade e em Lógica Difusa. Uma leitura importante sobre esse assunto é [Wooldridge, 2002].

Atualmente, as arquiteturas mais usadas em ambientes virtuais são as *baseadas em objetivo* (do inglês, *goal-driven architectures*), tendo como principal representante a arquitetura BDI⁶⁴ (acrônimo para *Belief-Desire-Intention* – Crenças, Desejos e Intenções). Esse modelo, que surgiu na década de 1970, ainda é amplamente usado na implementação de tipos variados de agentes, incluindo ACIs e atores sintéticos. Devido à sua importância, esse modelo será brevemente apresentado a seguir.

Modelo/Arquitetura BDI

No final da década de 1980, surgiu o modelo BDI (*Belief-Desire-Intention* – Crenças, Desejos e Intenções) [Rao e Georgeff, 1995], que foi e ainda é largamente usado para modelar agentes virtuais creíveis, devido à sua acurácia. Outra referência importante

⁶¹ Excetuando-se aqui a arquitetura SOAR, que é completa.

⁶² [http://en.wikipedia.org/wiki/Soar_\(cognitive_architecture\)](http://en.wikipedia.org/wiki/Soar_(cognitive_architecture))

⁶³ <http://soar.eecs.umich.edu/>

⁶⁴ http://en.wikipedia.org/wiki/Belief-desire-intention_software_model & http://www.inf.ufrgs.br/prosoft/bdi4jade/?page_id=46

sobre o assunto é [Wooldridge, 2002]. O modelo/arquitetura BDI é baseado na Teoria do Raciocínio Prático de Bratman [Bratman, 1987]. Essa teoria busca explicar o raciocínio prático humano, que é o processo de decidir o que fazer a cada momento no tempo (turno percepção-ação), de modo a alcançar os objetivos desejados.

Os agentes BDI trabalham com crenças, desejos e intenções, sendo modelados para demonstrar “atitudes mentais”. As *crenças* do agente consistem no conhecimento que ele tem sobre o mundo, incluindo o conhecimento sobre si mesmo (o que ele é capaz de fazer) e sobre os outros agentes do ambiente. As crenças podem incluir regras de inferência, que permitem ao agente deduzir novas crenças a partir do conhecimento atual. O termo “crença” é usado para deixar claro que as crenças do agente podem não corresponder à verdade, e que podem mudar ao longo do tempo.

Os *desejos* representam o estado motivacional do agente, consistindo em situações ou objetivos a serem alcançados. O agente deve buscar ativamente alcançar seus objetivos. Note que o agente pode ter desejos conflitantes (por exemplo, economizar dinheiro e comprar o melhor carro do mercado), porém o conjunto de *desejos ativos* em cada instante não pode ser conflitante. Os desejos que o agente se compromete a atingir, i.e, seus desejos ativos, também são conhecidos como Objetivos.

As *intenções* representam o estado deliberativo do agente (o que ele escolheu fazer). As intenções podem ser vistas como os desejos que o agente se comprometeu a realizar (em geral, executando um *plano* para chegar aos objetivos). Planos podem ser definidos como sequências de ações que o agente realiza para alcançar suas intenções.

A atividade (ou reatividade) dos agentes é, em geral, disparada por *eventos*, que podem ser externos (percebidos via sensores), ou internos (gerados com o intuito de disparar atualizações ou planos de atividades). Eventos podem atualizar as crenças do agente, disparar planos ou modificar alguns de seus objetivos.

Apesar de ser amplamente utilizada em aplicações diversas, a arquitetura BDI tem algumas limitações já conhecidas, destacando-se aqui a falta de uma representação explícita dos objetivos, e de mecanismos para planejamento “para frente”. Além disso, o modelo BDI não oferece mecanismos explícitos para aprendizagem, o que dificulta a implementação de agentes com capacidade de aprender e de se adaptar. Existem, contudo, diversos trabalhos que estendem o modelo BDI, buscando sanar suas limitações (e.g., [Ancona e Mascardi, 2004], [Pokahr et al., 2005], entre outros).

Modelo do Discurso/Diálogo

Como visto, ACIs são projetados para suportar diálogos multimodais, combinando texto, voz, gestos, e por vezes também movimentos corporais. Assim, o gerenciamento de diálogos com ACIs é naturalmente mais complexo do que no caso dos ACs textuais.

A fim de prover um controle mais sofisticado do fluxo do diálogo, alguns ACIs implementam um módulo exclusivo para Gerenciamento de Diálogo⁶⁵, com regras que definem estratégias para guiar a escolha dos padrões de saída que o agente deve

⁶⁵ Contudo, notem que existem ACIs cujo diálogo é gerenciado por módulos de mais alto nível, que coordenam o comportamento do ACI como um todo.

apresentar ao usuário. Esse módulo mantém uma *Base de Padrões* (e.g., pares adjacentes) usados durante o diálogo, e um *Modelo do Discurso*, que armazena o histórico do diálogo atual. Esse histórico é essencial na resolução de referências pronominais e de dêiticos. No caso de ACIs com personalidade e/ou humor, o histórico também registra os estados emocionais do agente.

Essas informações são usadas como contexto na escolha dos pares adjacentes que formam o diálogo, incluindo também a escolha dos modos de comunicação a serem utilizados pelo agente (e.g., os cumprimentos verbais podem ser acompanhados ou substituídos por acenos, uma vez que estamos tratando com agentes multimodais). Esse módulo também deve gerenciar a tomada de turno, decidindo quando o ACI mantém o turno e quando passa o turno para o usuário. O ideal é que exista um equilíbrio entre os turnos do usuário e os turnos do ACI, de modo a simular um diálogo natural.

É possível identificarmos dois tipos de diálogo com agentes conversacionais (e até mesmo com humanos): (1) os diálogos livres/sociais (e.g., [Bickmore e Cassell, 2004]); e (2) os diálogos orientados a tarefa (do inglês, *task-oriented dialogues*), nos quais o agente faz o papel de um especialista que guia um humano novato na realização de uma tarefa, ou na assimilação de algum conteúdo a ser aprendido (e.g., STEVE, já mencionado anteriormente). As estratégias de controle do andamento diálogo devem levar em conta o tipo de diálogo que o ACI realiza.

Relembramos aqui que os diálogos são organizados em três fases: abertura, desenvolvimento e fechamento. Para ACIs voltados a diálogos livres, a fase de desenvolvimento não apresenta uma estrutura (linear ou hierárquica) rígida entre os pares adjacentes que podem surgir durante a conversação. Além disso, o tópico e os focos da conversação devem ser selecionados de comum acordo com o usuário, durante o desenrolar do diálogo.

Já os diálogos com ACIs orientados a tarefa apresentam uma estrutura mais rígida, uma vez que têm um objetivo a cumprir. Esses ACIs deverão manter também um *Modelo da Tarefa*, com seus objetivos (implícitos ou explícitos). Nesse caso, o desenrolar do diálogo deve ser estruturado de acordo com as subfases da tarefa a ser realizar (ou nas subfases de apresentação do conteúdo a ser estudado pelo aluno, no caso de ACIs para Educação). Na fase de fechamento, o agente deve verificar se o seu objetivo foi alcançado (com a realização da tarefa por parte do usuário). Tais agentes são mais fortemente baseados em conceitos da Análise do Discurso (Seção 4.3.1).

Modelo do Usuário

Modelos do Usuário (MU) são uma representação do conhecimento sobre o usuário que o sistema armazena [Kobsa e Pohl 1995]. Esse modelo consiste em uma coleção de dados sobre um usuário específico, sendo o ponto de partida para quaisquer adaptações do comportamento do sistema a esse usuário. Assim, o MU é mais uma base de conhecimento que pode ser mantida pelos ACIs.

Podemos classificar os MU, em três dimensões, considerando: (1) a frequência de atualização das suas informações; (2) a forma de coleta de informações sobre o usuário; e (3) o tipo de raciocínio utilizado para inferir conhecimento sobre o usuário.

No que concerne a dimensão (1), os MUs podem ser *estáticos* ou *dinâmicos*. No primeiro caso, as informações são geralmente coletadas no momento do cadastramento do usuário, quando ele responde a um questionário provendo dados pessoais, como nome, idade, interesses/preferências e habilidades. Essas informações não serão modificadas ao longo da interação do usuário com o sistema. Já os modelos dinâmicos são atualizados à medida que o usuário interage com o sistema. A informação estática do MU é comumente chamada de *perfil do usuário*.

A coleta de informações (dimensão 2) pode ser realizada de forma *explícita* ou *implícita*. No primeiro caso, o sistema faz perguntas periódicas ao usuário, a fim de diagnosticar suas preferências, seus objetivos ou seu conhecimento sobre algum tema (no caso de sistemas de aprendizagem). Já no caso da coleta implícita, o sistema observa a interação do usuário, e coleta as informações relevantes, como, por exemplo, tarefas realizadas, navegação, entre outras. As informações a serem coletadas são especificadas pelo projetista do sistema.

Por fim, vejamos a dimensão (3). O tipo de raciocínio utilizado para inferir conhecimento sobre o usuário é, na maioria dos casos, baseado em regras. Contudo, já encontramos atualmente modelos mais sofisticados, baseados em Aprendizagem de Máquina. Em ambos os casos, o objetivo dessa inferência/aprendizagem é prover adaptação à medida que o usuário interage com o sistema.

No entanto, pode ser difícil coletar informações sobre o usuário. Nesses casos, quando há informações disponíveis sobre um grupo de usuários, podemos utilizar modelos baseados em estereótipos [Rich, 1989], que classificam o usuário de acordo com as características do grupo a que ele pertence. Essas informações são então usadas na adaptação do sistema. No caso dos modelos dinâmicos que coletam informação implicitamente, é possível iniciar a interação com base em um modelo baseado em estereótipo, e atualizar essas informações à medida que a interação avança, a fim de prover adaptações mais personalizadas.

Modelando Emoção e Personalidade - ACIs com Afetividade

Modelos de Computação Afetiva para ACIs têm por objetivo detectar, interpretar, processar e simular emoções humanas. Aplicações de ACIs com afetividade seguem o chamado “*loop-afetivo*” [Hook, 2008]. Esse *loop* conta com três passos, que estão alinhados com o ciclo geral de percepção-raciocínio-ação dos agentes inteligentes, já discutido aqui:

- 1) Reconhecer a emoção do usuário;
- 2) Selecionar a ação mais adequada, considerando também o estado do usuário; e
- 3) Sintetizar a resposta afetiva mais apropriada.

Os modelos OCEAN e OCC já foram largamente usados na construção de ACIs [Bartneck, 2002], bem como na elaboração de modelos e arquiteturas genéricos para ACIs com afetividade (personalidade, humor e/ou emoção) – ver Seção 4.3.2. Podemos citar aqui o MCAChat, um Modelo de Computação Afetiva para a construção de Chatterbots com emoção usando o modelo OCC construído em coautoria com uma das autoras deste curso [Jacob Jr. et al., 2013].

Algumas aplicações utilizam modelos de personificação em três camadas, considerando o humor como uma característica “meio-termo” entre as emoções temporárias e a personalidade permanente do agente [Kasap e Magnenat-Thalmann, 2007]. Como exemplo, temos ALMA (A Layered Model of Affect) [Gebhard, 2005], que busca prover um *profile* de personalidade, bem como humor e emoção em tempo real usando os modelos OCEAN e OCC.

ACIs com afetividade podem ser encontrados em várias aplicações, tais como ACIs para “contação” de histórias (do inglês, *storytelling*), que interagem com o usuário, buscando envolvê-lo na história [Gelin et al., 2010]; ACIs tutores – e.g., INES [Heylen et al., 2009], ACIs assistente pessoais [Mancini et al., 2008], entre outras. Vale a pena também observar o trabalho desenvolvido pelo *Affective Computing group* do MIT Media Lab⁶⁶.

Adaptação e Aprendizagem

Como já mencionado aqui, a maioria dos modelos e frameworks para a construção de agentes virtuais trabalha com regras predefinidas de percepção-raciocínio-ação. Contudo, ambientes dinâmicos requerem dos agentes a capacidade de se *adaptar* a situações novas (não previstas). Adaptação pode ser obtida através de mecanismos de *aprendizagem em tempo real* (online) [Hapfelmeier et al., 2012], que diferem da aprendizagem em modo *batch*, realizada com base em exemplos previamente coletados e, em geral, etiquetados.

A aprendizagem online possibilita a indução de padrões de comportamento durante a interação entre agente e usuário. Por exemplo, o agente pode construir “regras sociais” durante uma dada interação, aprendendo a distinguir entre comportamentos “aceitáveis” e “inaceitáveis” para aquela aplicação (e.g., xingamentos, mau humor, etc.). Essas regras também podem ser usadas pelo agente para classificar as percepções oriundas do ambiente (e.g., o agente pode se negar a continuar o diálogo com um usuário que demonstrar um comportamento inaceitável, desrespeitoso).

Contudo, aprendizagem online baseada na interação com humanos é uma área de pesquisa ainda relativamente recente, e bastante árdua. [Kasap e Magnenat-Thalmann, 2007] citam algumas iniciativas nessa área:

- O trabalho do extinto *MIT Synthetic Characters Group*, que tinha por objetivo treinar um cachorro virtual usando aprendizagem por reforço [Blumberg et al., 2002].
- A arquitetura ALEC para aprendizagem emocional, que provê aprendizagem cognitiva e de emoções, além de tomada de decisão, usando principalmente redes neurais [Gadano, 2003].
- O trabalho de [Wan e Tang, 2002], que apresenta um agente virtual (ator sintético) capaz de aprender a realizar tarefas específicas (como pular obstáculos) usando algoritmos evolucionários.

⁶⁶ <http://www.media.mit.edu/research/groups/affective-computing>

- [Gatzoulis et al. 2004] apresentam um agente virtual robô servente baseado em aprendizagem difusa por reforço.
- O trabalho de [Dinerstein e Egbert, 2005] combina aprendizagem de emoções e imitação, apresentando um agente virtual (ator sintético) que se adapta rapidamente através da interação com humanos.

Claramente, ainda há muito trabalho a ser feito nessa área, a fim de prover os ACIs com capacidade de aprendizagem online e adaptação. Esse caminho fica então indicado como uma tendência futura nas pesquisas.

4.5.3 Implementando ACIs

Discutiremos aqui algumas abordagens adotadas por pesquisadores da área na implementação de ACIs. Inicialmente, salientamos que, apesar de a arquitetura apresentada na Seção 4.5.2.1 ser bastante completa e geral, por simplificação, a maioria dos ACIs não implementa todos os componentes ali detalhados. [Kopp et al., 2014] argumentam que um grande número de ACIs tem sido construído com base em modelos que não apresentam processamento incremental da entrada/saída. Tais modelos integram o processamento da entrada, raciocínio (planejamento e deliberação) e geração da saída, a fim de simplificar a implementação dos ACIs.

Contudo, como veremos abaixo, já existem trabalhos que buscam sanar essa deficiência, oferecendo frameworks com processamento incremental, que possibilitam a criação de ACIs mais flexíveis e verossímeis. A seguir, veremos alguns dos trabalhos mais proeminentes na área: o trabalho de Cassell e seu grupo, seguido da arquitetura SAIBA e de algumas de suas extensões.

4.5.3.1 O trabalho de Cassell e seu grupo

A arquitetura inicial proposta por [Cassell et al. 2000b] caracteriza-se por apresentar processamento sequencial e não incremental. Essa arquitetura é mais restrita do que a vista na Seção 4.5.2.1, consistindo em um módulo *gerenciador de entradas* (que fusiona áudio, voz e entradas gestuais), que aciona o *módulo de raciocínio*, responsável por gerar o comportamento do agente com base na interpretação da entrada e no *planejamento da saída* (mais detalhes abaixo). Além disso, a arquitetura conta com um *módulo reativo*, que mapeia diretamente algumas entradas em saídas, a fim de implementar reações rápidas.

A cada turno, o agente deve optar por seguir o caminho *deliberativo* (através do módulo de raciocínio) ou o *reativo* (disparando o módulo reativo). Contudo, em ambos os casos, o processamento utiliza planos autocontidos ou especificações de conjuntos completos de comportamentos de entrada e saída. Assim, o sistema produz uma lista completa de comportamentos, que podem ser executados com agendamento prévio (baseado em tempo absoluto), ou então podem ser disparados por eventos específicos. Essa lista especifica antecipadamente (de forma não incremental) como alguns comportamentos do agente devem se desenvolver ao longo do tempo.

Uma importante extensão desse trabalho foi a ferramenta BEAT (*The Behavior Expression Animation Toolkit*) [Cassell et al., 2001], para criação de agentes animados com aspecto humano a partir de texto. Segundo os autores, da mesma forma como

sistemas de texto-voz sintetizam voz a partir de texto, BEAT sintetiza/gera comportamento conversacional não verbal (i.e., gestos, olhares) em agentes incorporados a partir de texto.

A ferramenta mapeia o texto digitado pelo construtor do agente em comportamentos através de um conjunto de regras que consideram a informação linguística e contextual contida no texto para controlar os movimentos (mãos, braços, face) e a entonação do agente animado. Os comportamentos gerados são armazenados para uso futuro, quando o agente estiver de fato dialogando com algum usuário dentro de alguma aplicação.

BEAT utiliza uma linguagem tipo XML desenvolvida pelos autores para representar “funções comunicativas”⁶⁷. O sistema associa ao texto de entrada (digitado pelo construtor do ACI) *tags* que informam sobre os sinais não verbais pertinentes ao texto/contexto da entrada. A saída do agente também contém *tags* especificando o tipo de animação a ser realizada, e por quanto tempo.

Vale ressaltar que nem todos os ACIs trabalham com bancos de dados de imagens pré-prontas. Existem sistemas mais sofisticados que são capazes de gerar esses elementos gráficos em tempo real, de acordo como andamento do diálogo (a parte textual) e das percepções do agente sobre os movimentos do usuário [Kasap e Magnenat-Thalmann, 2007].

O sistema foi construído para operar em tempo real, sendo modular e extensível. BEAT permite que o usuário insira regras para a criação de novos comportamentos (movimentos e olhares), possibilitando assim a criação de agentes com diferentes personalidades e humor. Apesar de ainda não utilizar processamento incremental na geração da saída (uma vez que os comportamentos são previamente gerados), essa ferramenta oferece um grande avanço na construção de ACIs.

O trabalho de Cassell e seu grupo alcançou grande sucesso, por ter qualidade e também por ter sido um dos pioneiros na área. De fato, esses trabalhos serviram como base/inspiração para a maioria dos ACIs existentes, como pode ser facilmente visto na literatura relacionada (e.g., Max [Kopp et al., 2005], [Kenny et al., 2007], dentre dezenas de outros trabalhos).

4.5.3.2 Buscando Processamento Incremental

Apesar de se observar o uso predominante da arquitetura de Cassell et al. por tantos anos, já existem atualmente modelos que oferecem processamento incremental, buscando superar esse estilo rígido de processamento.

De acordo com [Kopp et al., 2014], a primeira proposta de processamento incremental para gerar comportamento multimodal (fala e gestos sincronizados) foi o sistema ACE (*Articulated Communicator Engine*) [Kopp e Wachsmuth, 2004]. Nesse sistema, o processamento se dá conectando automaticamente *unidades de comportamento* que mantêm sua sincronização interna. Essa abordagem foi estendida pela arquitetura ASAP [van Welbergen et al., 2012] [Kopp et al., 2014], que provê

⁶⁷ Ver http://en.wikipedia.org/wiki/Jakobson's_functions_of_language

unidades de comportamento mais granulares, que podem ser ligadas para antecipar eventos de entrada. Voltaremos a falar nesse trabalho mais abaixo.

O trabalho de [Nijholt et al., 2008], por sua vez, propõe uma abordagem que vincula diretamente a ocorrência (*timing*) do comportamento do agente às previsões (expectativas) de ocorrência de eventos realizados pelo usuário, alcançando assim um grau mais refinado de coordenação temporal com os movimentos do usuário.

Processamento incremental também passou a ser considerado como indispensável para a modelagem de diálogos mais naturais. Por exemplo, o trabalho de [Schlangen e Skantze, 2011] descreve os mecanismos internos em ACs em termos de módulos abstratos que se comunicam através de unidades incrementais (*incremental units - IUs*), que são passadas entre os módulos. Muitas implementações desse modelo foram desenvolvidas e muitos aspectos de ACs têm sido modelados com sucesso dentro desse framework de processamento incremental. Contudo, as implementações da arquitetura UI têm sido uni modais até o momento (tipicamente, fala e texto).

4.5.3.3 O Framework SAIBA

Atualmente, podemos observar uma preocupação em padronizar a geração da saída dos ACIs, de modo a permitir arquiteturas mais modularidade e maior flexibilidade do comportamento gerado. A arquitetura/framework SAIBA [Kopp et al., 2006] é o exemplo mais proeminente dessa tendência. Como podemos ver na página web do projeto⁶⁸, o principal objetivo do SAIBA (*Situation, Agent, Intention, Behavior, Animation*) é oferecer um framework unificado para geração de comportamento multimodal para ACIs, de modo a facilitar a interação e o compartilhamento de recursos entre pesquisadores da área.

Esse framework especifica a geração multimodal em três etapas de processamento de mais alto nível, cada etapa podendo ser desmembrada em módulos menores:

- 1) Planejamento do conteúdo (intenção comunicativa do agente);
- 2) Planejamento do comportamento do agente (i.e., planejamento da realização multimodal do conteúdo); e
- 3) Realização dos comportamentos planejados.

Essas etapas de processamento se comunicam via representações em duas linguagens tipo XML: *Function Markup Language* (FML) e *Behavior Markup Language* (BML). Buscando neutralidade, essas linguagens são: livres de aplicação e de domínio; independentes do processador de som e do gerador de imagens; e distinguem com clareza informações de natureza distinta (informações relacionadas a funções *versus* informações relacionadas a processos de especificação do comportamento).

A informação passada da etapa (1) para a etapa (2) descreve *intenções comunicativas*⁶⁹ e de expressão do agente, sem ainda fazer referências ao seu comportamento físico (i.e., como ele irá realizar essas intenções). Essa informação é

⁶⁸ SAIBA - <http://www.mindmakers.org/projects/saiba/wiki>

⁶⁹ Pragmática – intenções comunicativas e atos de fala – http://pt.wikipedia.org/wiki/Atos_de_fala
<http://plato.stanford.edu/entries/pragmatics/> & <http://plato.stanford.edu/entries/speech-acts/>

especificada usando a linguagem FML, que oferece uma descrição semântica dos aspectos relevantes no planejamento do comportamento verbal e não verbal do agente.

A informação oriunda da etapa (2) descreve o modo como os comportamentos multimodais devem ser realizados pela etapa final do processo de geração (etapa 3). Essa informação é representada usando a linguagem BML, que provê uma descrição geral e independente dos realizadores de som e imagem que podem ser usados pelo ACIs. Contudo, essa representação deve prover um bom nível de detalhe na descrição do comportamento do agente, desde a mera ocorrência e o *timing* relativo das ações envolvidas, até a definição detalhada de uma forma particular de comportamento.

Esse modelo, bem como as linguagens FML e BML, têm se tornado um padrão para geração de comportamento em ACIs. Contudo, segundo [Kopp et al., 2014], SAIBA foi planejado para ser um modelo de alto nível. Assim, o framework não detalha como deve ser realizada a coordenação entre entradas e saídas do agente, nem como deve ser implementado o processamento incremental.

4.5.3.4 Extensões de SAIBA

Podemos encontrar na literatura relacionada muitos trabalhos que se inspiram no framework SAIBA. Porém, não haveria tempo para discutirmos todas essas iniciativas aqui. Assim, apresentaremos brevemente duas de suas extensões mais importantes: os trabalhos de [Scherer et al., 2012] e de [Kopp et al., 2014].

Trabalhos recentes têm estendido o framework SAIBA para incluir também a representação do comportamento não verbal do usuário, com o objetivo de considerar suas intenções comunicativas na interação com o ACI. Essa tarefa requer a percepção e a integração de uma grande variedade de pistas (dicas) de comportamento, como por exemplo, risadas, palavras faladas, mudanças de entonação e gestos simultâneos.

O trabalho de [Scherer et al., 2012], na área de reconhecimento automático de comportamento, propõe a linguagem PML (*Perception Markup Language*) como um padrão para representar percepções de comportamento não verbal do usuário. Segundo os autores, PML possibilita que os agentes virtuais reajam ao comportamento do usuário de modo adequado, melhorando a fluidez das interações entre usuário e agente. PML também trabalha com probabilidades para representar incerteza. Os diferentes níveis de processamento de PML foram projetados para controlar/coordenar atualizações do estado do diálogo, planejamento de intenções e geração adequada de comportamento.

De forma geral, os pesquisadores na área de ACI estão buscando desenvolver modelos que favorecem processamento incremental do diálogo, bem como uma integração mais flexível entre percepção e ação. Dentre essas iniciativas, destaca-se ASAP (*Artificial Social Agent Platform*), um framework para construção de agentes capazes de prover respostas rápidas e adaptação a novas situações, a fim de oferecer diálogos multimodais fluentes em tempo real.

Segundo [Kopp et al., 2014], ASAP estende SAIBA para uma arquitetura completa de entrada-saída com processamento incremental. ASAP inclui uma extensão dos modelos de representação (em especial, BML), para cobrir especificação e geração incrementais; e a utilização de PML para representar a percepção do agente.

ASAP provê uma coleção de módulos de software para construção de agentes conversacionais virtuais e robôs. Estão disponíveis também ferramentas para viabilizar a construção desses agentes, como IPAACA, um *middleware* que implementa uma arquitetura para processamento incremental. Mais informações podem ser obtidas na página web do projeto⁷⁰.

4.5.4 Aplicações Atuais

Assim como no caso dos ACs textuais (chatterbots), podemos encontrar ACIs como aplicações na Internet⁷¹, ou como agentes embutidos em sistemas variados. Contudo, as possibilidades de uso dos ACIs são bem mais amplas, devido a eles apresentarem um estilo de interação mais rico do que os chatterbots textuais – ver [Nijholt, 2004].

As aplicações citadas nesta seção são tanto de ACIs mais simples quanto de humanos virtuais, uma vez que alguns autores não diferenciam esses dois tipos de agentes. Dentre as diversas aplicações de ACIs, destacam-se ACIs tutores, ACIs assistentes pessoais, ACIs para SAC, ACIs para entretenimento, terapia online, entre outros. Abaixo discutiremos suas características principais e daremos alguns exemplos de aplicações encontradas na literatura.

- *Os ACIs tutores* são companheiros virtuais em ambientes de aprendizagem [Kerry et al., 2009] [Veletsianos e Russell, 2014] ou em ambientes de treinamento [Kenny et al., 2007a]. Esses agentes são bastante utilizados em ambientes de EaD, com o objetivo de diminuir a sensação de isolamento dos alunos. Moyo [2014] mostra que ACIs conseguem promover o aprendizado dos estudantes, por causar empatia e motivá-los a permanecer no ambiente. Além disto, Tais agentes são capazes de demonstrar tarefas complexas, buscando trazer o foco de atenção do aluno para detalhes importantes da apresentação através de movimentação e gestos (como movimentar-se dentro do ambiente virtual, e apontar para informações importantes). Alguns desses agentes são capazes de demonstrar emoção. Segundo [Johnson et al., 2000], esses agentes melhoram o aprendizado dos alunos/aprendizes.
- *ACIs acompanhantes/assistentes pessoais*⁷² têm usos muito diversos, como por exemplo, servir como acompanhantes de idosos que vivem sozinhos [Ring et al., 2013], [Garrido et al., 2013]. Nesses casos, tais agentes atuam como interfaces para os ambientes inteligentes [de Carolis et al., 2012], auxiliando os habitantes a utilizarem os recursos disponíveis. Ou, no caso de idosos, ajudando-os com suas necessidades de medicação.
- *ACIs para sistemas de terapia online* (do inglês, *e-therapy*), que, como acontece com sistemas de EaD, também possuem grandes taxas de evasão [Grolleman et al., 2006]. Um exemplo desse tipo de ACI é o do SimCoach, projeto do Institute for Creative Technologies, que provê três ACIs diferente para auxílio⁷³.

⁷⁰ <http://asap-project.ewi.utwente.nl/>

⁷¹ Contudo, não é comum encontrar ACIs para simples conversação na Web, como ELIZA e ALICE.

⁷² Alguns autores denominam os ACIs tutores como assistentes pessoais, uma vez que essas duas aplicações apresentam muitas semelhanças.

⁷³ <http://ict.usc.edu/prototypes/simcoach/>

- *ACIs para e-fitness*, que atuam como *personal trainers* [Bickmore, 2003].
- *ACIs comentaristas e apresentadores* são uma aplicação que vem emergindo nos últimos anos, e podem ser utilizados para comentar eventos em tempo real, jogos, passeios turísticos etc. Como exemplo dessa categoria de ACI temos Eric [Strauss e Kipp, 2008].
- *ACIs para Serviço de Atendimento ao Consumidor (SAC)* são um tipo de aplicação já consolidada, e que vem crescendo ao longo dos anos [Zdenek, 2007] [Kuligowska, 2015]. Como exemplos temos Anna, assistente da IKEA⁷⁴ e a Lu, assistente do Magazine Luiza⁷⁵.
- *ACIs em Jogos Digitais* para simples entretenimento. Esses assistentes buscam aumentar o grau de realismo do jogo [Lankes e Bernhaupt, 2011] [Downs e Sundar 2011].
- *ACIs em Jogos Sérios* têm o objetivo de aumentar o interesse e o grau de imersão do jogo [Laufer e Tatai, 2004] [Rosmalen et al., 2012]. Esse tipo de ACI pode se confundir com ACIs tutores, uma vez que em ambos os casos o objetivo do ACI é auxiliar no aprendizado.
- *ACIs para “Contação” de Histórias* (do inglês, *storytelling*) não são ainda muito difundidos, porém são muito interessantes [Ryokai et al., 2003], [Gelin et al., 2010].

Assim como no caso dos chatterbots, com ELIZA e ALICE, também existem ACIs que ganharam destaque, como STEVE e REA. Um ACI mais atual que também se destacou foi GRETA, que é capaz de se comunicar em 6 idiomas distintos⁷⁶. GRETA foi desenvolvida dentro do projeto europeu SEMAINE⁷⁷, que deu origem a alguns ACIs muito interessantes, como POPPY e SPIKE, que variam seu comportamento e seu estado emocional durante o desenrolar da conversação. Mais exemplos de ACIs podem ser encontrados no site chatbots.org, em constante atualização.

Voltando a atenção novamente para Computação Afetiva, destacamos os ACIs tutores com afetividade, que buscam reconhecer as emoções dos estudantes a fim de promover um aprendizado mais efetivo. Um exemplo desse tipo de aplicação é o agente PrimeClimb [Davoodi et al., 2013], que usa o modelo OCC para tentar reconhecer o motivo da emoção do usuário (se ela é devida a algo que o estudante realizou –e.g., orgulho ou vergonha, – ou devida ao comportamento do estudante –e.g., admiração ou reprovação).

Quando o agente é capaz de detectar corretamente a emoção do usuário, ele pode adequar sua resposta, auxiliando aquele estudante de forma mais efetiva. Mais um exemplo de ACI que utiliza o modelo OCC pode ser visto no trabalho de [Zong et al., 2000]. Nesse trabalho, o ACI expressa emoção por meio de diferentes ações (gestos faciais e corporais), e modificando parâmetros do discurso, como volume, velocidade e ênfase em certas palavras.

⁷⁴ http://www.ikea.com/ms/en_US/customer_service/contact_us/contact.html

⁷⁵ <http://www.magazineluiza.com.br/portaldalu/>

⁷⁶ <http://perso.telecom-paristech.fr/~pelachau/Greta/>

⁷⁷ <http://www.semaine-project.eu/>

4.5.5 Limitações e Tendências Futuras

Devido à grande quantidade e à variedade de implementações de ACIs, fica difícil apontar as limitações desses agentes de forma geral (como foi feito no caso dos chatterbots textuais). Aqui, cada aspecto do ACI tem suas limitações próprias (e.g., limitações na percepção visual do agente, na interpretação/síntese de voz, na construção dos diálogos, etc.).

Assim, cada aspecto interacional do ACI deve ser inicialmente investigado separadamente: sensores e softwares (que podem limitar a percepção do agente); atuadores e processadores de voz e imagem (que podem degradar a qualidade da saída gerada pelo agente - suas ações), e módulos de gerenciamento do diálogo (que nem sempre conseguem manter uma conversação plausível e natural com o interlocutor). Contudo, vale notar que as limitações existentes podem afetar uma aplicação e não afetar outra, tudo depende do foco da aplicação.

Nesse contexto, citamos a dissertação de mestrado de [Jonkman, 2012], que discute as limitações do comportamento não verbal de um agente para contação de histórias. Já [Bartneck e Lyons, 2009] discutem as limitações na modelagem de expressões faciais em geral. Encontramos ainda muitos artigos que apontam as limitações dos reconhecedores de voz como responsáveis pelo nem sempre satisfatório desempenho dos ACIs (e.g., [López-Cózar et al., 2011]).

Existem também as dificuldades em integrar todas as modalidades da comunicação, coordenando satisfatoriamente gestos, olhares e fala a fim de prover uma conversação natural [Kopp et al., 2014]. Em relação às técnicas utilizadas, nota-se que ainda há muito trabalho a ser feito em médio e longo prazo, especialmente no que se refere à aprendizagem online e à adaptação dos ACIs.

Apesar das limitações atuais, podemos identificar um uso crescente de ACIs em aplicações diversas, como visto na Seção 4.5.4. Destaca-se aqui seu uso em interfaces do usuário, principalmente nas aplicações que requerem comunicação multimodal em tempo real [André e Pelachaud, 2010]. Essa tendência é justificada por esses agentes oferecerem aos usuários um estilo de interação natural (via fala e gestos). Além disso, ACIs podem também demonstrar emoção, tornando a interação mais verossímil e menos “anônima” para o usuário. Agentes bem projetados tornam a interação com os computadores mais agradável, fomentando bons sentimentos nos seus usuários, dando um *feedback* positivo de que o usuário é capaz de interagir com um computador. E nada pode ser mais simples do que interagir com uma máquina via fala e gestos, como é a proposta dos ACIs.

Outro aspecto muito relevante é o *comportamento social* dos ACIs. Acreditamos que os ACIs do futuro deverão possuir características colaborativas, sendo capazes de interagir com outros atores, levando em consideração sinais como empatia, emoções, papéis dos usuários e relações sociais. Também é uma tendência futura a participação dos ACIs em grupos de usuários, a fim de melhorar os relacionamentos de longo prazo. Por fim, destacamos a personificação de tais agentes como característica central para o seu sucesso.

4.6. Conclusão

Este documento apresentou um panorama abrangente sobre os Agentes Inteligentes Conversacionais, com destaque para os ACs incorporados. Discutimos *background*, histórico, modelos, implementações e aplicações de tais agentes. Apresentamos ainda as limitações desses agentes, bem como as tendências futuras na área.

Encontramos atualmente diversos grupos de pesquisa com projetos relacionados ao desenvolvimento de agentes conversacionais e/ou de humanos virtuais, contribuindo de forma consistente para o crescimento da área. Muito provavelmente, os principais trabalhos futuros nessa área terão sua semente nesses grupos.

O site [Chatbots.org, 2016] afirma que foram identificadas 14.052 universidades e instituições que desenvolvem algum trabalho relacionado à área de conversação com Inteligência Artificial. Uma lista de instituições acadêmicas pode ser vista nesse site⁷⁸. Abaixo, destacamos algumas instituições indicadas no site, e mais outras que julgamos também relevantes para essa área:

- ArticulaLab - Carnegie Mellon University⁷⁹ (Estados Unidos)
- Media Lab - Massachusetts Institute of Technology⁸⁰ (Estados Unidos)
- Institute for Creative Technology - University of Southern California⁸¹ (Estados Unidos)
- TELECOM ParisTech⁸² (França)
- Human-Centered Multimedia - Augsburg University⁸³ (Alemanha)
- Human Media Interaction - University of Twente⁸⁴ (Holanda)
- DFKI (Deutsche Forschungszentrum für Künstliche Intelligenz) – Centro de pesquisa alemão para inteligência artificial⁸⁵ (Alemanha)

Além das instituições acadêmicas, encontramos também empresas de softwares desenvolvendo produtos com ACs. O site [Chatbots.org, 2016] apresenta uma lista de empresas que atuam nessa área⁸⁶. Dentre elas, destacamos *Artificial Solutions*⁸⁷, que alcançou sucesso produzindo ACs assistentes virtuais para empresas variadas. Mais exemplos de ACs comerciais, com indicação das empresas desenvolvedoras, podem ser vistos no artigo de [Kuligowska, 2015].

Assim, acreditamos que o futuro aponta para um uso sempre crescente de agentes conversacionais em sistemas com diversas finalidades (design, educação, etc) e sobre plataformas variadas.

⁷⁸ <https://www.chatbots.org/universities/>

⁷⁹ <http://www.articulab.justinecassell.com/index.html>

⁸⁰ <http://media.mit.edu/research/groups-projects>

⁸¹ <http://ict.usc.edu/research/>

⁸² <http://perso.telecom-paristech.fr/~pelachau/site/projects.html#>

⁸³ <http://www.informatik.uni-augsburg.de/en/chairs/hcm/>

⁸⁴ <http://hmi.ewi.utwente.nl/>

⁸⁵ <http://www.dfki.de/web/research>

⁸⁶ <https://www.chatbots.org/companies/>

⁸⁷ <http://www.artificial-solutions.com/natural-language-interaction-solutions/digital-employee/>

4.7. Referências Bibliográficas⁸⁸

- ALICE Bot - Artificial Linguistic Internet Computer Entity. *ALICE A.I. Foundation* (2016). Online em <http://www.alicebot.org/>.
- Allen, J.F. (1995). *Natural Language Understanding*. The Benjamin Cummings Pub. Company, New York.
- Allport, G. W. (1938). *Personality: a psychological interpretation*. Henry Holt and Company, New York.
- Ancona, D., Mascardi, V. Hübner, J.F. e Bordini, R. (2004). "Coo-AgentSpeak: Cooperation in AgentSpeak through Plan Exchange". In: Proc. of AAMAS 2004 (Int. Conf. on Autonomous Agents and Multiagent Systems), ACM press, pp.698-705.
- André, E. e Pelachaud, C. (2010). "Interacting with embodied conversational agents". In: *Speech technology*. Springer US, pp. 123-149.
- Barros, F.A. e Robin, J. (2001). "Processamento de Linguagem Natural". *Revista Eletrônica de Iniciação Científica*, v.1, pp. 1 – 61.
- Bartneck, C. (2002). "Integrating the OCC model of emotions in embodied characters". In: *Workshop on Virtual Conversational Characters: Applications, Methods and Research Challenges*, Melbourne.
- Bartneck, C. e Lyons, M. J. (2009). "Facial Expression Analysis, Modelling and Synthesis: Overcoming the Limitations of Artificial Intelligence with the Art of the Soluble". In: *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*. IGI Global., pp. 33-53.
- Bates, J. (1994). "The role of emotion in believable agents". *Communications of the ACM*, v. 37, n.7, pp. 122-125.
- Bickmore, T. (2003). "Relational Agents: Effecting Change through Human-Computer Relationships". Tese de Doutorado. Media Arts & Sciences, Massachusetts Institute of Technology. <http://affect.media.mit.edu/pdfs/03.bickmore-phd.pdf>
- Bickmore, T. e Cassell, J. (2004) "Social dialogue with embodied conversational agents." *Advances in Natural Multimodal Dialogue Systems*. Springer Netherlands, pp. 23-54.
- Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M. e Tomlinson, B. (2002). "Integrated learning for interactive synthetic characters". In: *Proceedings of the ACM SIGGRAPH*.
- Bolt, R. A. (1980) "Put-that-there": Voice and gesture at the graphics interface, v.14, n.3, ACM.
- Bratman, M. (1987). *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

⁸⁸ Todos os links da Web indicados neste documento estavam ativos na data de finalização deste documento.

- Brooks, R.A. (2001). "How to build complete creatures rather than isolated cognitive simulators". In: *Architectures for Intelligence*, Lawrence Erlbaum Associates, K. VanLehn, ed., pp. 225–239.
- Calvo, R. A., D'Mello, S., Gratch, J. e Kappas, A. (Eds.) (2015). *The Oxford Handbook of Affective Computing*. Oxford University Press.
- Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S. e Stone, M. (1994). "Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents". In: *Computer Graphics, ACM SIGGRAPH*, pp. 413–420.
- Cassell, J., Bickmore, T., Billinghamurst, M., Campbell, L., Chang, K., Vilhjálmsón, H. e Yan, H. (1999). "Embodiment in Conversational Interfaces: Rea". In: *Proceedings of the CHI'99 Conference*, pp. 520-527.
- Cassell, J. e Vilhjálmsón, H. (1999). "Fully Embodied Conversational Avatars: Making Communicative Behaviors Autonomous". *Autonomous Agents and Multi-Agent Systems*, v.2, n.1 pp. 45-64.
- Cassell, J., Sullivan, J., Prevost, S. e Churchill, E. (eds) (2000). *Embodied Conversational Agents*, MIT Press.
- Cassell, J., Bickmore, T., Vilhjálmsón, H., e Yan, H. (2000a). "More than Just a Pretty Face: Affordances of Embodiment". In: *Proceedings of the Conference on Intelligent User Interfaces*, pp. 52-59.
- Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsón, H., e Yan, H. (2000b). "Human Conversation as a System Framework: Designing Embodied Conversational Agents". In: [Cassel et al. 2000].
- Cassel, J., Vilhjálmsón, H., e Bickmore, T. (2001). "BEAT: the Behavior Expression Animation Toolkit". In: *Proceedings of SIGGRAPH '01*. pp. 477–486.
- Chatbots.org (2016). *Virtual Agents / Chatbots Directory*. <https://www.chatbots.org/>.
- Chatterbot. (2016). "Wikipedia, The Free Encyclopedia". <http://en.wikipedia.org/wiki/Chatterbot>.
- Colby, K.M., Hilf, F.D., Weber, S., e Kraemer, H.C. (1972). "Turing-like Indistinguishability Tests for the Validation of a Computer Simulation of Paranoid Processes". *Artificial Intelligence*, v.3, pp. 199-222.
- Davoodi, A., Kardan, S. e Conati, C. (2013). "Understanding Users' Interaction Behavior with an Intelligent Educational Game: Prime Climb." In: *AIED 2013 Workshops Proceedings Volume 2 Scaffolding in Open-Ended Learning Environments (OELEs)*. pp. 9-16.
- de Carolis, B. Ferilli, S. e Novielli, N. (2012) "Recognizing the User Social Attitude in Smart Environments", *Ambient Intelligence*, pp. 240-255.
- Dinerstein, J. e Egbert, P. K. (2005) "Fast multi-level adaptation for interactive autonomous characters". *ACM Transactions on Graphics*, v. 24 n.2, pp.262– 288.

- Downs, E. P., e Sundar, S. S. (2011). "We won" vs. "they lost": Exploring egoenhancement and self-preservation tendencies in the context of video game play. *Entertainment Computing*, v. 2, pp. 23-28.
- Egges, A, Kshirsagar, S, Magnenat-Thalmann, N (2004). "Generic personality and emotion simulation for conversational agents". *Computer Animation and Virtual Worlds*, v.15, n.1 pp. 1-13.
- Eliza Chat bot (2016). "Is she a Rogerian psychotherapist, a semi intelligent chat bot, or just a toy?" <http://nlp-addiction.com/eliza/>
- Erman, Lee D., et al. (1980) "The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty." *ACM Computing Surveys (CSUR)*, v.12, n.2, pp. 213-253.
- Gadanhó, S. (2003). "Learning behavior-selection by emotions and cognition in a multi-goal robot task". *Journal of Machine Learning*, v. 4, pp. 385–412.
- Galley, Michel, et al. (2003). "Discourse segmentation of multi-party conversation". In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics,
- Garrido, P., Sanchez, A., Martinez, F. J., Baldassarri, S., Cerezo, E. e Seron, F. J. (2013). "Using 3d virtual agents to improve the autonomy and quality of life of elderly people". In: *Ambient Intelligence-Software and Applications*. Springer International Publishing. pp. 129-136.
- Gatzoulis, C., Tang, W., Wan, T.R. (2004). "Fuzzy Reinforcement Learning for an Evolving Virtual Servant Robot". In: *IEEE International Workshop on Robot and Human Interactive Communication held at Kurashiki, Japan*. IEEE, pp. 685- 690.
- Gebhard, P. (2005). "Alma –A Layered Model of Affect". In: *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems*. pp. 29–36.
- Gelin, R., d'Alessandro, C., Le, Q., Deroo, O., Doukhan, D., Martin, J. C., Pelachaud, C., Rilliard, A. e Rosset, S. (2010). "Towards a Storytelling Humanoid Robot". In: *AAAI Fall Symposium: Dialog with Robots*.
- Gratch, J., Rickel, J., André, E., Cassell, J., Petajan, E., e Badler, N. (2002). "Creating Interactive Virtual Humans: Some Assembly Required". *IEEE Intelligent Systems*, July/August, pp. 54-61.
- Green Jr, B., et al. (1961) "Baseball: an automatic question-answerer". In: *Western Joint IRE-AIEE-ACM computer conference*. ACM Press.
- Grolleman, J., van Dijk, B., Nijholt, A. e van Emst, A. (2006). "Break the habit! designing an e-therapy intervention using a virtual coach in aid of smoking cessation". In: *Persuasive Technology*. Springer Berlin Heidelberg. pp. 133-141.
- Grosz, Barbara J., e Candace L. Sidner (1986). "Attention, intentions, and the structure of discourse." *Computational linguistics*, v.12, n.3, pp. 175-204.
- Hapfelmeier, A., T. Hothorn, K. Ulm, e C. Strobl (2012). "A new variable importance measure for random forests with missing data". *Statistics and Computing*, pp. 1–14.

- Heylen, D., Theune, M., Op den Akker, R. e Nijholt, A. (2009). “Social agents: The first generations”. In: 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII 2009). IEEE. pp. 1-7.
- Höök, K. (2008). “Affective loop experiences – what are they?” Lecture Notes in Computer Science, v.5033, pp. 1–12.
- Jacob Jr., A., da Mata, E., Santana, A., Francês, C., Costa, J. e Barros, F. (2013). “Adapting Chatterbots Interaction for Use in Children’s Education”. In: Emerging Research and Trends in Interactivity and the Human-Computer Interface. pp. 413-428.
- Johnson, W.L., Rickel, J.W., e Lester, J.C. (2000) “Animated pedagogical agents: Face-to face interaction in interactive learning environments”. International Journal of Artificial Intelligence in Education, v. 11, pp. 47–78.
- Jonkman, F. (2012). “Nonverbal behaviour of an embodied storyteller”. Dissertação de Mestrado. University of Twente, NL. http://essay.utwente.nl/61766/1/Master_thesis_Ferry_Jonkman.pdf
- Kasap, Z. e Magnenat-Thalmann, N. (2007). “Intelligent virtual humans with autonomy and personality: State-of-the-art”. Intelligent Decision Technologies. vol 11, n.2. pp 3-15.
- Kenny, P., Hartholt, A., Gratch, J., Traum, D., Marsella, S. e Swartout, B. (2007). “The More the Merrier: Multi-Party Negotiations with Virtual Humans”. In: Proc. of the National Conference on Artificial Intelligence (AAAI 2007), pp. 1970-1971.
- Kenny, P., Hartholt, A., Gratch, J., Swartout, W., Traum, D., Marsella, S. e Piepol, D. (2007). “Building interactive virtual humans for training environments”. In: Proceedings of I/ITSEC. Vol. 174.
- Kerry, A., Ellis, R. & Bull, S. (2009). “Conversational agents in E-Learning”. In: Applications and Innovations in Intelligent Systems XVI. Springer, pp. 169-182
- Kobsa, A. e Pohl, W. (1995) “The user modeling shell system BGP-MS” User Modeling and User-Adapted Interaction. v.4, n.2. pp 59–106
- Kopp, S., Gesellensetter, L., Krämer, N., e Wachsmuth, I. (2005), “A conversational agent as museum guide – design and evaluation of a real-world application”. In: Panayiotopoulos et al. (eds.): Intelligent Virtual Agents Berlin: Springer-Verlag. pp. 329-343.
- Kopp, S., e Wachsmuth, I. (2004). "Synthesizing multimodal utterances for conversational agents". Journal of Computer Animation and Virtual Worlds, v.15, pp. 39-52.
- Kopp, S., van Welbergen, H., Yaghoubzadeh, R. e Buschmeier, H. (2014). “An architecture for fluid real-time conversational agents: integrating incremental output generation and input processing”. Journal on Multimodal User Interfaces, v.8, n.1, pp. 97-108.

- Kuligowska, C. (2015). "Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents". In: Professionals Center for Business Research, Vol 2.
- Lankes, M. e Bernhaupt, R. (2011). "Using embodied conversational agents in video games to investigate emotional facial expressions". In: Entertainment Computing. v.2, n.1, pp. 29-37.
- Laufer, L. e Tatai, G. (2004). "Learn, Chat and Play—An ECA supported stock markets elearning curricula". In: Proceedings of the IASTED International Conference on Web-Based Education (WBE 2004). ACTA Press, Innsbruck. pp. 16-18.
- Laven, S. (2016). Chatterbot Central - *The Simon Laven Page*. <http://www.simonlaven.com/>
- López-Cózar, R., Callejas, Z., Espejo, G. e Griol, D. (2011). "Enhancement of Conversational Agents by Means of Multimodal Interaction". In: Conversational Agents and Natural Language Interaction: Techniques and Effective Practices. IGI Global. pp. 223-242.
- Mancini, M., Niewiadomski, R., Bevacqua, E. e Pelachaud, C. (2008). "Greta: a SAIBA compliant ECA system". In: Workshop on Animated Conversational Agents (WACA'08). Paris, França.
- Marcuschi, L.A. (1986) *Análise da Conversação*. Editora Ática, São Paulo.
- Mateas, M. (1999) *An Oz-centric review of interactive drama and believable agents*. Springer Berlin Heidelberg.
- Mauldin, M. L. (1994). "Chatterbots, TinyMUDs, and the Turing Test: Entering the Loebner Prize Competition". In: Proceedings of Twelfth National Conference on Artificial Intelligence. pp. 16-21.
- McCrae, R. R. e Costa Jr., P. T. (1998). "A five-factor theory of personality". In: *Handbook of personality: Theory and Research*. Vol. 2, pp. 139-153.
- Morie, J.F., Chance, E., Haynes, K. e Rajpurohit, D. (2013). "Embodied conversational agent avatars in virtual worlds: Making today's immersive environments more responsive to participants". In: *Believable bots*. Springer, pp.99-118.
- Moyo, S. G. (2014). "Effective Tutoring with Empathic Embodied Conversational Agents". Tese de Doutorado. The Open University.
- Neves, A. M. M.(2005). "IAIML - Um Mecanismo para o Tratamento de Intenção em Chatterbots". Tese de Doutorado em Ciência da Computação. Universidade Federal de Pernambuco.
- Neves, A. M. M., Barros, F. A. e Hodges, C. (2006). "iAIML: a Mechanism to Treat Intentionality in AIML Chatterbots". In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006). pp. 225-231.
- Nijholt, A. (2004). "Where computers disappear, virtual humans appear". *Computers & Graphics*. v.28, n.4, pp. 467-476.
- Nijholt, A., Reidsma, D., van Welbergen, H., op den Akker, H. , e Ruttkay, Z.M. (2008). "Mutually coordinated anticipatory multimodal interaction". In *Verbal and*

- nonverbal features of human–human and human–machine interaction. Springer, Berlin. pp 70–89.
- Nishida, T., Nakazawa, A., Ohmoto, Y. e Mohammad, Y. (2014). "History of Conversational System Development". In: *Conversational Informatics: A Data-Intensive Approach with Emphasis on Nonverbal Communication*. Springer. p. 43-62.
- Ortony, A., Clore, G. L. & Collins, A (1988). *The Cognitive Structure of Emotions*. Cambridge University Press, 1st. ed.
- Pauchet A. e Sabouret N. (2012). "Embodied Conversational Agents and Affective Computing". In: *European Agent Systems Summer School -EASSS, Valencia, Espanha*.
- Picard, R. (2003). "Affective computing: challenges". *International Journal of Human-Computer Studies*. v.59, n.1–2, pp. 55-64.
- Pokahr, A., Braubach, L. e Lamersdorf, W. (2005) "A Flexible BDI Architecture Supporting Extensibility". In: *Proc. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, pp. 379–385,
- Rao, A. e Georgeff M. (1995). "BDI Agents, from Theory to Practice" In: *Proceedings of the First International Conference on Multiagent Systems*. pp 312-319
- Rich, E. (1989) "Stereotypes and User Modelling", In: Kobsa A. And Wahlster W. (eds.) *User Models in Dialog System*, Springer, Berlin.
- Rickel, J. e Johnson W.L. (1999). "Animated agents for procedural training in virtual reality: Perception, cognition, and motor control." *Applied Artificial Intelligence*. v.13, pp. 343-382.
- Ring, L., Barry, B., Totzke, K. e Bickmore, T. (2013). "Addressing Loneliness and Isolation in Older Adults: Proactive Affective Agents Provide Better Support". In: *Conference on Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association* pp. 61-66.
- Russell, S. e Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall. 3rd Edition.
- Ryokai, K., Vaucelle, C. e Cassell, J. (2003). "Virtual Peers as Partners in Storytelling and Literacy Learning." *Journal of Computer Assisted Learning*, v.19, n.2. pp. 195-208.
- Saucier, G. e Goldberg, L. (1998). "What is beyond the big five". *Journal of Personality*. v.66, pp. 495–524.
- Schank, R. C., e Abelson, R. (1975). *Scripts, plans, and knowledge*. New Haven, CT: Yale University.
- Schegloff, E. A. (1972) "Sequencing in conversational openings." *Directions in sociolinguistics*, pp. 346-380.
- Scherer, S., Glodek, M., Schwenker, F., Campbell, N. e Palm, G. (2012), "Spotting Laughter in naturalistic multiparty conversations: a comparison of automatic online and offline approaches using audiovisual data", *ACM Transactions on Interactive*

- Intelligent Systems: Special Issue on Affective Interaction in Natural Environments, v.2, n.1, pp. 4:1-4:31
- Schlangen D. e Skantze. . (2011) "A General, Abstract Model of Incremental Dialogue Processing". *Dialogue & Discourse*, v. , n.1. pp:83–111.
- Searle, J. R. (1976). *Expression and Meaning*. Cambridge University Press.
- Strauss, M. e Kipp, M. (2008) "ERIC: a generic rule-based framework for an affective embodied commentary agent", In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pp. 47-54
- Kenny, P., et al. (2007). "Building Interactive Virtual Humans for Training Environments" In: *Proc. of the Simulation and Education Conference*.
- Tang, W. e Wan, T. R. (2002) "Integrated Learning for Interactive Synthetic Characters". In: *WSCG– Short Papers*, pp. 137–144
- Turing, A. (1950). "Computing machinery and intelligence". *Mind*. v. 49, pp. 433-460. <http://www.csee.umbc.edu/courses/471/papers/turing.pdf>
- van Welbergen H, Reidsma D, Kopp S (2012) , "An Incremental Multimodal Realizer for Behavior Co-Articulation and Coordination" In: *LNCS*. Nakano Y, Neff M, Paiva A, Walker M (Eds) v.7502, Springer. pp. 175-188.
- Veletsianos, G., e Russell, G. S. (2014) "Pedagogical agentes". In: *Handbook of research on educational communications and technology*. Springer, pp. 759-769.
- Walker, M. e Steve W. (1990) "Mixed initiative in dialogue: An investigation into discourse segmentation." In: *Proceedings of the 28th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics.
- Wallace, R. (2009). "The Anatomy of A.L.I.C.E.". *Parsing the Turing Test*. Springer. pp. 181-210.
- Weizenbaum, J. (1966). "ELIZA: A computer program for the study of Natural Language Communication between man and machine". *Communications of ACM*. v.9, n.1, pp. 35-36. <http://dl.acm.org/citation.cfm?doid=365153.365168>
- Wilson, I. (2000). "The artificial emotion engine, driving emotional behaviour". In *AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*. pp. 20-22.
- Winograd, Terry (1972). "Understanding natural language." *Cognitive psychology*, v.3, n.1, pp.1-191.
- Woods, W. A. (1973) "Progress in natural language understanding: an application to lunar geology." In: *Proceedings of the June 4-8, 1973, national computer conference and exposition*. ACM..
- Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, England.
- Zdenek, S. (2007). "Just Roll Your Mouse over Me: Designing Virtual Women for Customer Service on the Web". *Technical Communication Quarterly*. v.16, n.4, pp. 397-430.

Zong, Y., Dohi, H. e Ishizuka, M. (2000). "Multimodal Presentation Markup Language MPML with emotion expression functions attached". In: Proceedings of Third International Symposium on Multimedia Software Engineering. pp. 359-365

Capítulo

5

Interfaces Cérebro-Computador

João Luís Garcia Rosa

Laboratório de Computação Bio-inspirada (BioCom)

Departamento de Ciências da Computação

Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo

Av. Trabalhador São-carlense, 400 - Centro, São Carlos, SP, CEP: 13566-590 - Tel. 16-3373-6640

joaoluís@icmc.usp.br

Abstract

Brain-Computer Interface (BCI) is a form of communication that enables individuals unable to perform movements to connect to external assistive devices using brain signals. Noninvasive BCIs capture changes in blood flow or fluctuations in electric and magnetic fields caused by the activity of large populations of neurons. The electroencephalography (EEG), a non-invasive technique, measures the electrical activity of the brain in different locations of the head, typically using electrodes placed on the scalp. With the proper removal of artifacts, signal processing and machine learning, human EEG carries enough information about the intention of planning and execution. The objective of this mini-course is to show how the understanding of the activity of the brain can provide a way to allow communication without muscle movements.

Resumo

Interface cérebro-computador (BCI) é uma forma de comunicação que permite que indivíduos incapazes de realizar movimentos se conectem a dispositivos externos usando sinais cerebrais. BCIs não-invasivas capturam mudanças no fluxo sanguíneo ou flutuações de campos elétricos e magnéticos causados pela atividade de grandes populações de neurônios. A eletroencefalografia (EEG) mede a atividade elétrica do cérebro em diferentes locais da cabeça, geralmente usando eletrodos colocados no couro cabeludo. Com a remoção adequada de artefatos, processamento de sinal e aprendizagem de máquina, a EEG humana carrega informações suficientes sobre a intenção de planejamento e execução. O objetivo do minicurso é mostrar como o entendimento da atividade do cérebro pode fornecer uma maneira para permitir a comunicação sem movimentos musculares.

5.1. Introdução

5.1.1. Por que Interface Cérebro-Computador?

Interfaces Cérebro-Computador (Brain-Computer Interfaces) (BCI) são uma forma de comunicação que *habilita* indivíduos *incapazes* de realizar movimentos a se conectarem com dispositivos externos de assistência usando sinais cerebrais. Uma BCI é um mecanismo que permite a um usuário *interagir com o mundo exterior* por meio de medidas de atividade neural associadas com processos mentais [46].

A expressão *interface cérebro-computador* apareceu pela primeira vez na década de 1970 [66, 67]. Vidal usou a expressão para representar o conceito de colocar os *sinais elétricos observáveis do cérebro* para trabalhar como carregadores de informação em comunicações homem-computador ou para o propósito de controlar aparato externo [26]. Veja os componentes de um sistema BCI na Figura 5.1.

Figura 5.1. Esquema de uma BCI. Adaptado de [54].

Em geral, o registro da atividade neural é realizado por técnicas não invasivas. BCIs não-invasivas capturam mudanças no fluxo sanguíneo ou flutuações em campos elétricos ou magnéticos causados pela atividade de grandes *populações de neurônios* [57]. A *eletroencefalografia (EEG)*¹, uma técnica *não-invasiva*, mede a atividade elétrica do cérebro em diferentes posições da cabeça, tipicamente usando eletrodos no couro cabeludo [44, 22].

Com a remoção adequada de artefatos², processamento de sinal e aprendizado de máquina, o registro da EEG do ser humano apresenta informações suficientes sobre a *intenção* de planejamento e execução. Veja um sinal registrado pela EEG na Figura 5.2.

Modelos cerebrais com base em *neurodinâmica* procuram entender e representar as razões pelas quais os neurônios são células excitáveis. A corrente elétrica microscópica de cada neurônio soma-se com as correntes de outros neurônios, o que provoca uma

¹*Eletroencefalografia (EEG)* é o método ou técnica de gravação dos potenciais elétricos que são gerados espontaneamente durante a atividade fisiológica de células no córtex cerebral. *Eletroencefalograma (EEG)* refere-se ao registro ou traços obtidos pela eletroencefalografia. Observe que a mesma sigla EEG é empregada nos dois casos, restando ao contexto sua distinção. Máquinas de EEG são projetadas para adquirir, processar, e exibir os sinais de EEG como formas de onda ou outras formas de dados sobre uma tela ou no papel.

²*Artefatos* são potenciais elétricos indesejados obtidos na EEG, normalmente de origem fisiológica (contração, músculos, olhos) ou externa [14].

Figura 5.2. Um segundo de sinal registrado pela EEG [25].

diferença de potencial elétrico macroscópica, medida pela EEG, que registra a atividade dos padrões de *populações de neurônios*. Isto é, um bom modelo neural deve reproduzir a dinâmica de neurônios, tendo em conta as *propriedades dinâmicas de populações de neurônios*, além das propriedades eletrofisiológicas de neurônios individuais.

O objetivo é mostrar como o entendimento da *atividade elétrica do cérebro* pode fornecer uma maneira para permitir a comunicação *sem* movimentos musculares. A intenção é, a partir do estudo do *comportamento neurodinâmico do cérebro*, investigar formas e propor modelos que permitam criar interfaces cérebro-computador, preferencialmente não-invasivas.

Nas últimas décadas, as BCIs baseadas em EEG têm atraído a atenção de pesquisadores no campo da neurociência, engenharia neural e reabilitação clínica. O plano é usar os dados obtidos por meio da BCI para analisar os *movimentos pré-motores*, mudanças no cérebro que ocorrem *antes* que haja realmente um movimento, e aplicá-los a um tratamento adequado de dispositivos protéticos.

5.1.2. Conhecendo o cérebro

Dois conceitos muito diferentes em relação à forma como o cérebro funciona como um todo coexistem em neurociência: o *modelo clássico*, no qual o cérebro é descrito como sendo composto de uma série de cadeias causais formadas por redes nervosas que funcionam em paralelo (as redes neurais artificiais convencionais [30]); e o *modelo neurodinâmico*, no qual o cérebro é operado por um caos dinâmico não-linear, que se parece com ruído, mas que apresenta uma espécie de ordem oculta [18].

Walter Freeman argumenta que existem duas unidades básicas de organização do cérebro: o *neurônio* e a *população de neurônios* [18]. Embora o neurônio tenha sido a base para a neurobiologia, massas de neurônios que interagem formando *populações de neurônios* são consideradas para uma *visão macroscópica do cérebro*. Como os neurônios, as populações de neurônios também têm estados e padrões de atividade, mas elas realizam diferentes “tarefas” macroscópicas. Entre o neurônio microscópico e essas “tarefas” macroscópicas, há populações *mesoscópicas* [18].

5.1.3. Registro

A eletroencefalografia (EEG) registra a atividade elétrica de populações mesoscópicas de neurônios. Uma BCI não-invasiva compreende a captação e análise de tal atividade, com o objetivo de oferecer uma saída alternativa às saídas normais musculares. Modelos de cérebro baseados em populações de neurônios com comportamento neurodinâmico e caótico, como os conjuntos-K [16], podem ser empregados para construir tal BCI.

As interfaces cérebro-computador (BCI) compreendem o uso da análise *on-line* de ondas cerebrais para obter informação sobre o *estado mental* de um sujeito como base para realizar alguma ação externa, como selecionar uma letra de um teclado virtual ou mover um dispositivo robótico, fornecendo um *canal de comunicação alternativa e controle* que não depende do caminho de saída normal do cérebro para nervos e músculos periféricos, que podem estar paralisados em alguns pacientes [49].

5.1.4. Tipos de BCI

Uma BCI pode ser classificada como [57]: (1) *invasiva*, baseada em eletrodos implantados, permite registrar atividade elétrica de neurônios individuais. Necessita de cirurgia; (2) *não-invasiva*, baseada em medições de atividade elétrica de populações de neurônios, por intermédio, por exemplo, da EEG. Não há necessidade de cirurgia, apenas o uso de uma “touca” de eletrodos na cabeça (Figura 5.3); e *semi-invasiva*, técnica intermediária, em que há uma certa “invasão”. Eletrodos são colocados na superfície do cérebro. Há necessidade de incisão cirúrgica.

Figura 5.3. As ondas cerebrais são registradas pela “touca” de eletrodos e analisadas por um computador [15].

A detecção e interpretação automáticas de atividades do cérebro relacionadas à atenção ou à *intenção* apresentam interesse significativo para muitas aplicações militares e civis. Essa interpretação da atividade cerebral pode prover informação sobre os movimentos intencionados de uma pessoa, movimentos imaginados ou foco de atenção e, portanto, pode ser informação valiosa para otimizar ou substituir a comunicação tradicional baseada em motor, ou seja, comunicação baseada em movimento muscular, movimentos de mãos, braços, olhos, língua, etc., entre uma pessoa e um computador ou outro dispositivo de saída [8].

5.2. Neurociência Básica

5.2.1. Neurônios

O neurônio típico tem muitos dendritos (entradas) e um axônio (saída). O axônio transmite informação por meio de trens de pulso *microscópicos* (*spikes*). Os dendritos integram

informação por meio de ondas contínuas de corrente iônica. Os neurônios estão ligados por sinapses. Cada sinapse conduz corrente elétrica. A corrente microscópica de cada neurônio é somada com as correntes de outros neurônios, o que provoca uma diferença de potencial *macroscópica*, medida com um par de eletrodos extracelulares (E) como a EEG [18, 21]. A EEG registra os padrões de atividade de *populações de neurônios mesoscópicas*. A soma das correntes que um neurônio gera em resposta a estímulos elétricos produz o potencial pós-sináptico. Veja a Figura 5.4.

Figura 5.4. Neurônio típico mostrando os dendritos (entradas), onde geralmente ocorrem as sinapses (Sy), o corpo celular, o axônio (saída), a zona de gatilho (ZG) e a direção do potencial de ação (*spike* elétrico). Note que as letras 'E' representam o par de eletrodos extracelulares. Adaptado de [58] e [18].

O tipo predominante de sinapse no cérebro do mamífero é a *sinapse química*, que opera por meio de liberação de uma substância química, o *neurotransmissor*, do terminal pré-sináptico para o terminal pós-sináptico, em resposta a uma despolarização (Figura 5.5). *Despolarização* é uma redução do potencial da membrana celular para zero mV, sendo que o interior do neurônio torna-se mais positivo. A despolarização para um nível de potencial crítico, o *limiar*, causa o início de um impulso. No seu pico, o interior da célula torna-se positivo em relação ao seu exterior.

Figura 5.5. Nas sinapses químicas, o terminal pré-sináptico libera uma substância química, o neurotransmissor, em resposta a uma despolarização [43, 58].

Nas sinapses químicas, há a *fenda sináptica*, que é uma pequena separação celular entre as células. Estão presentes vesículas contendo moléculas de neurotransmissores no terminal pré-sináptico, e quando o potencial de ação alcança estas vesículas, os neurotransmissores são liberados para a fenda sináptica.

As células nervosas influenciam outras por: (1) *excitação*, ou seja, elas *produzem* impulsos em outras células e (2) *inibição*, ou seja, elas *previnem* a liberação de impulsos em outras células.

Em uma sinapse excitatória, o transmissor liberado pelo terminal pré-sináptico despolariza a célula pós-sináptica, fazendo com que o potencial de sua membrana atinja o limiar. Em uma sinapse inibitória, o transmissor tende a manter o potencial da membrana da célula pós-sináptica abaixo do limiar.

Os neurônios corticais, por causa de suas interações sinápticas, formam *populações de neurônios*. Variáveis de estado de pulso e onda microscópicas são utilizadas para descrever a atividade dos neurônios individuais que contribuem para a população, e as variáveis de estado mesoscópicas (também de pulso e de onda) são utilizadas para descrever as atividades coletivas que os neurônios originam. Atividade de massa no cérebro é descrita por uma densidade de pulsos, em vez da frequência do pulso. Isto é feito por meio da gravação a partir do exterior da célula do disparo de pulsos de muitos neurônios simultaneamente. A mesma corrente que controla os disparos de neurônios é medida pela EEG, que não permite distinguir contribuições individuais. Felizmente, isso não é necessário [18].

A ação coletiva dos neurônios forma padrões de atividade que vão além do nível celular e se aproximam do nível do organismo. A formação de estados mesoscópicos é o primeiro passo para isso. Desta forma, o nível de atividade é decidido pela população, não por indivíduos [18].

5.2.2. Hodgkin-Huxley

A teoria de Hodgkin-Huxley (H-H) do potencial de ação, formulada há mais de 60 anos, permanece uma das grandes histórias de sucesso da biologia [29]. Juntamente com a rede neural artificial de McCulloch e Pitts [48], a teoria H-H provê a fundação para a neurociência computacional moderna. A teoria foi resultado de intensa colaboração experimental e teórica entre Alan Lloyd Hodgkin e Andrew Fielding Huxley, de 1938 até a publicação em 1952 de cinco artigos no *Journal of Physiology*. Os mecanismos iônicos responsáveis pela geração de um potencial de ação foram elucidados pelos trabalhos de Hodgkin e Huxley com o axônio gigante da lula.

Foram quatro os experimentos principais:

1. Cole e Curtis demonstraram que o potencial de ação é associado a um grande aumento na condutância da membrana [11];
2. Hodgkin e Huxley fizeram a primeira gravação intracelular de um potencial de ação [31] (Figura 5.6); Isto demonstrou que o potencial de ação excedia zero mV, contradizendo a hipótese de Bernstein [4] de que o aumento fundamental na permeabilidade da membrana é não-seletivo;

3. Hodgkin e Katz explicaram o pico do potencial de ação mostrando que era resultado de um aumento na permeabilidade do sódio [33];
4. Hodgkin, Huxley e Katz desenvolveram um circuito grampo de voltagem para habilitar medição quantitativa de correntes iônicas do axônio gigante da lula.

Figura 5.6. A primeira gravação intracelular de um potencial de ação do axônio da lula mostrou uma forma como esta. Adaptado de [68].

A lula é um invertebrado marinho que usa propulsão a jato para o movimento [61]. Seu axônio mais comprido também é o mais grosso (axônio gigante). Trata-se de uma fibra não-mielinizada com um diâmetro de aproximadamente 0.5 mm e vários centímetros de comprimento³. É uma das maiores células conhecidas. Portanto, constitui um sistema ideal para a realização de experimentos.

Hodgkin e Huxley mostraram que as despolarizações do axônio da lula engatilham uma corrente de entrada seguida de uma corrente de saída. Usando substituição iônica, eles demonstraram que esta corrente de rede poderia ser separada em dois componentes distintos: uma corrente de entrada rápida conduzida por íons Na^+ e uma corrente de saída de ativação mais lenta conduzida por íons K^+ . A partir de experimentos, eles concluíram que essas duas correntes resultam de mecanismos de permeabilidade independentes para Na^+ e K^+ com condutâncias que mudam como uma função do tempo e do potencial da membrana.

No entanto, sua contribuição mais memorável foi a representação empírica de dados experimentais em um modelo quantitativo [32], a primeira descrição completa da excitabilidade de uma única célula. Eles modelaram as mudanças suaves de corrente observadas em termos de poros ou canais que ou estavam abertos ou fechados, e usando previsões geradas por uma abordagem estatística para a probabilidade dos canais estarem abertos [29].

5.2.3. Plasticidade

Plasticidade corresponde à capacidade do cérebro em mudar a força da conexão das sinapses. As formas da plasticidade sináptica são potenciação de longo prazo (*long-term*

³As células dos vertebrados possuem diâmetros de poucos μm .

potentiation - LTP) e depressão de longo prazo (*long-term depression* - LTD). Ambas envolvem mudanças na sinapse que duram horas ou dias. Mais recentemente outros tipos de plasticidade foram propostos [57]: (1) plasticidade dependente do tempo do *spike* (*spike-timing dependent plasticity* - STDP), quando o tempo relativo dos *spikes* de entrada e saída determina a polaridade da mudança sináptica, (2) facilitação/depressão de curto prazo, quando a plasticidade é rápida mas não tem duração longa.

A LTP é uma das formas de plasticidade sináptica mais importante. Na sua forma mais simples, a LTP envolve um aumento na força da conexão sináptica entre dois neurônios causado por *disparos correlacionados* dos dois neurônios. A LTP é considerada a implementação biológica do famoso *postulado de Hebb* (também chamado de aprendizado hebbiano ou plasticidade hebbiana) [57]: “Se um neurônio A é consistentemente envolvido em causar o disparo de um outro neurônio B, então a força da conexão de A a B deve ser aumentada.” A LTP é encontrada em várias áreas do cérebro, incluindo o hipocampo e o neocortex [57].

A LTD envolve uma diminuição na força de uma conexão sináptica causada, por exemplo, por disparos *não correlacionados* entre os dois neurônios envolvidos. A LTD tem sido observada principalmente no cerebelo, ainda que coexista com a LTP no hipocampo, neocortex e outras áreas do cérebro [57].

Protocolos tradicionais que demonstram a LTP/LTD envolvem a estimulação de um neurônio pré-sináptico e um pós-sináptico simultaneamente. Esses protocolos manipulam a taxa de disparo dos neurônios pré e pós-sináptico mas não o tempo entre os *spikes* pré e pós-sinápticos. Estudos recentes revelaram que a *temporização precisa* dos *spikes* pré e pós-sinápticos pode determinar se a mudança na força sináptica é positiva ou negativa. Esta forma de plasticidade sináptica é conhecida como STDP [57].

Na STDP hebbiana, quando o *spike* pré-sináptico ocorre um pouco antes do *spike* pós-sináptico (1 a 40 ms) a sinapse é fortalecida. Quando ocorre um pouco depois (1 a 40 ms), a força sináptica é diminuída. A STDP hebbiana foi observada no cortex cerebral e no hipocampo do mamífero. O fenômeno oposto, a STDP anti-hebbiana, em que a sinapse é fortalecida quando o *spike* pré-sináptico ocorre depois do pós-sináptico e vice-versa, foi observada em algumas estruturas, como por exemplo, as sinapses inibitórias do cerebelo dos peixes elétricos [57].

LTP, LTD e STDP são plasticidades de longo prazo, pois as mudanças que elas causam podem durar horas e até dias. A plasticidade de curto prazo faz com que as sinapses correspondentes funcionem como filtros temporais nos padrões de *spikes* de entrada. Na depressão de curto prazo (*short-term depression* - STD) o efeito de cada *spike* sucessivo em um trem de *spikes* de entrada (sequência de *spikes*) é diminuído comparado com o *spike* precedente [57].

Portanto, se um neurônio recebe uma explosão de *spikes* como entrada, o primeiro *spike* tem o maior efeito, e cada *spike* sucessivo causa cada vez menos mudanças no potencial da membrana até que um ponto de equilíbrio seja alcançado e todos os *spikes* subsequentes tenham o mesmo efeito diminuído no neurônio pós-sináptico. A facilitação de curto prazo (*short-term facilitation* - STF) exibe o efeito oposto, em que cada *spike* sucessivo tem um efeito maior que seu predecessor, até que um ponto de saturação seja

alcançado. Tanto STD quanto STF têm papéis importantes em regular a dinâmica de redes corticais [57].

5.3. Neurodinâmica

Um sistema dinâmico é um sistema cujo estado varia com o tempo. Para proceder com o estudo da neurodinâmica, é necessário um *modelo matemático espaço-estado*: conjunto de *variáveis de estado* cujos valores devem conter informação suficiente para prever a evolução futura do sistema.

5.3.1. Espaço de estados

A equação espaço-estado descreve o movimento de um ponto em um espaço de estados N -dimensional. O espaço de estados pode ser um espaço euclidiano ou um subconjunto deste. Pode ser também um espaço não-euclidiano como um círculo, uma esfera, uma tora ou algum outro elemento diferenciável. O interesse, porém, está confinado em um espaço euclidiano. O espaço de estados é importante porque provê uma ferramenta visual/conceitual para analisar a dinâmica de um sistema não-linear. Mudanças no estado do sistema com o tempo t são representadas como uma curva no espaço de estados, com cada ponto na curva carregando (explícita ou implicitamente) um rótulo que grava o tempo da observação. Esta curva é chamada de *trajetória* ou *órbita* do sistema.

5.3.2. Atratores

Sistemas dissipativos são geralmente caracterizados pela presença de conjuntos de atração ou coletores de dimensionalidade mais baixa que a do espaço de estados. Por coletor entende-se uma superfície k -dimensional embutida em um espaço de estados N -dimensional, definido por um conjunto de equações. Esses coletores são chamados de *atratores*, pois eles são subconjuntos limitados para os quais regiões de condições iniciais de volume do espaço de estados não-zero convergem quando o tempo t aumenta. O coletor pode consistir de um único ponto no espaço de estados, neste caso um atrator de ponto. Alternativamente, pode ser na forma de uma órbita periódica, nesse caso um ciclo limite estável, no sentido de que trajetórias próximas aproximam-se dele assintoticamente (Figura 5.7).

Figura 5.7. Um atrator ciclo limite [77].

5.3.3. Rede de Hopfield

A rede (ou modelo) de Hopfield consiste de um conjunto de neurônios e de um conjunto correspondente de unidades de atraso, formando um sistema de retro-alimentação (*feedback*) de *loops* múltiplos (Figura 5.8). O número de *loops* de *feedback* é igual ao número de neurônios. Basicamente, a saída de cada neurônio é retro-alimentada, via unidade de atraso, a cada um dos outros neurônios da rede. Ou seja, não há auto-retro-alimentação [30].

Figura 5.8. Uma rede de Hopfield [78].

A convergência é geralmente assegurada, uma vez que Hopfield mostrou que os atratores deste sistema dinâmico não-linear são estáveis, não periódicos ou caóticos como em alguns outros sistemas [34].

5.4. Conjuntos-K de Freeman

Nas décadas passadas, testemunhou-se avanços significativos nas técnicas de monitoramento do cérebro, em parte devido aos avanços dos métodos experimentais [64, 22]. Abordagens de modelamento foram desenvolvidas para dar suporte às descobertas empíricas. Os cérebros têm sido descritos como sistemas dinâmicos complexos com trajetórias que evoluem de um horizonte atrator com colinas e vales [16, 17].

De acordo com a abordagem de sistemas dinâmicos, a trajetória dinâmica do cérebro pode repousar intermitentemente, por uma fração de segundo, em um vale [18, 35, 56]. O vale corresponde a uma recuperação da memória pelo sujeito, que pode ser medida por meio de um padrão de atividade espacialmente distribuído. Este padrão tem significado para o sujeito baseado em suas experiências e intenções anteriores.

Em relação à neurociência no nível mesoscópico [18, 19], a conexão teórica entre a atividade dos neurônios no nível microscópico em pequenas redes neurais e a atividade de grupos de células na escala mesoscópica ainda não é bem entendida [20]. Katzir-Katchalsky sugere tratar os grupos de células usando termodinâmica formando uma hierarquia [36]: conjuntos-K de Freeman.

O modelo de rede neural K é baseado em estudos de décadas de Walter Freeman sobre a dinâmica do processamento perceptual no cérebro [16, 17, 18, 20]. Os conjuntos-K de Freeman descrevem a dinâmica espaço-temporal do cérebro usando uma abordagem

hierárquica. Incluem os conjuntos- K_0 , K_1 , K_2 , K_3 , K_4 , K_5 e K_6 , que descrevem a hierarquia de redes neurais no cérebro, com o aumento da complexidade de estrutura e função [20, 39] (Figura 5.9). Os conjuntos- K procuram representar os padrões espaciais de fase e amplitude das oscilações geradas pela população neural [40, 41] e usam um conjunto de equações diferenciais ordinárias (EDO) com parâmetros distribuídos para descrever a hierarquia de populações neurais começando das micro-colunas até os hemisférios [37]. São utilizados em diversas aplicações, sendo que uma aplicação recente do conjunto- K_3 envolve agrupamento de bases de dados sociais [42, 60].

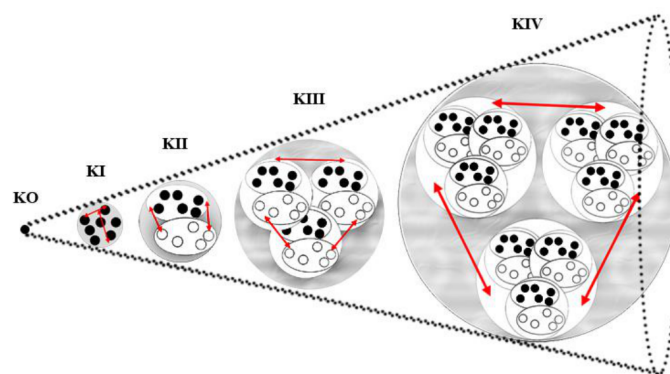


Figura 5.9. A hierarquia dos conjuntos- K , mostrando a progressão dos conjuntos de populações neurais do K_0 ao K_4 . Adaptado de [38].

5.5. Gravação e Estimulação da Atividade do Cérebro

5.5.1. Gravação da atividade cerebral

Nosso conhecimento sobre o funcionamento do cérebro aumentou dramaticamente nas últimas décadas devido ao desenvolvimento e refinamento de várias técnicas de gravação da atividade cerebral [22]. Tais avanços vão desde o estudo da atividade sináptica no nível microscópico ao refinamento de técnicas de imagem cerebral em um nível macroscópico. Sistemas modernos de aquisição de dados e novos *designs* de eletrodos permitiram a gravação simultânea de dezenas de neurônios em uma escala maior, e os computadores poderosos permitiram simulações e análise de dados mais complexas, dando assim origem ao campo da *neurociência computacional* [57].

Como já foi dito, a comunicação entre neurônios no cérebro ocorre por meio de *spikes* ou potenciais de ação, que são pulsos elétricos tudo-ou-nada produzidos quando o neurônio recebe uma quantidade suficiente de corrente de entrada de outros neurônios via conexões sinápticas. Portanto, algumas das primeiras tecnologias para registrar a atividade do cérebro foram baseadas na detecção de mudanças nos potenciais elétricos em neurônios (técnicas invasivas baseadas em eletrodos) ou em grandes populações de neurônios (técnicas não invasivas, como a EEG). Técnicas mais recentes são baseadas na detecção da atividade neural indiretamente por meio da medida da mudança no fluxo sanguíneo que resulta da atividade neural aumentada em uma região (fMRI) ou medindo as mudanças no campo magnético em volta do crânio causadas pela atividade neural (MEG).

5.5.1.1. Técnicas invasivas

Técnicas que permitem registrar neurônios individuais no cérebro são tipicamente invasivas, ou seja, envolvem alguma forma de cirurgia, em que uma parte do crânio é removida, um eletrodo ou implante é colocado no cérebro e a parte removida do crânio é recolocada. Normalmente são gravações realizadas em animais, tais como macacos e ratos. A gravação não causa dor, pois o cérebro não tem receptores internos para dor, mas a cirurgia e o processo de recuperação podem causar dor e envolver riscos como infecções [57].

No caso de humanos, gravações invasivas são feitas apenas em instalações clínicas tais como durante cirurgia do cérebro ou quando pacientes estão sendo monitorados por causa de atividade anormal do cérebro (por exemplo, crises) antes da cirurgia. A maior vantagem das gravações invasivas é que elas permitem gravação de *spikes* (os sinais de saída dos neurônios) em uma escala de tempo de milisegundos. Isso contrasta com as técnicas não invasivas, que medem correlatos indiretos da atividade neural, tal como o fluxo do sangue, que ocorrem em uma escala de tempo maior (centenas de milisegundos). Técnicas invasivas são baseadas na tecnologia dos eletrodos.

Um *microeletrodo* é simplesmente um fio muito fino ou outro condutor elétrico usado para fazer contato com o tecido cerebral. Um eletrodo típico é feito de tungstênio ou de uma liga platina-irídio e é isolado exceto na ponta, que mede em torno de $1\ \mu\text{m}$ de diâmetro (o diâmetro do corpo celular de um neurônio está na faixa de dezenas de μm) [57].

Para registrar a atividade de um número grande de neurônios, os microeletrodos podem ser arranjados em uma estrutura de grade para formar uma *matriz multieletrodo* de $m \times n$ eletrodos, em que os valores de m e n normalmente estão entre 1 e 10 [57] (Figura 5.10). Estas matrizes foram desenvolvidas para gravações *in vitro* e *in vivo*. O foco aqui é para matrizes implantáveis para gravações *in vivo* porque elas são as mais relevantes para as BCIs. Os tipos mais comuns de matrizes implantáveis são matrizes de microfio, baseadas em silício, e de microeletrodos flexíveis.

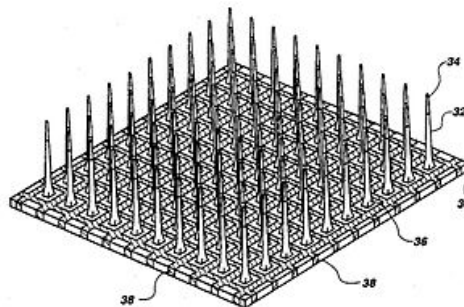


Figura 5.10. Exemplo de uma matriz multieletrodo (matriz de Utah de 10×10 eletrodos) [69].

A maior vantagem das matrizes multieletrodos sobre os sistemas mais convencionais de único eletrodo, é a resolução espacial aumentada; a habilidade de gravação simultânea de vários neurônios abre a porta para a extração de tipos de informação complexa como sinais de posição ou velocidade que podem ser úteis para controlar dispositivos protéticos. Matrizes implantáveis também têm suas desvantagens, especialmente

se o dispositivo implantado permanece no tecido cerebral por um tempo longo (como requerido pelo controle de longo prazo das próteses) [57].

A *eletrocorticografia* (ECoG) é uma técnica para gravar sinais cerebrais colocando eletrodos na superfície do cérebro. Necessita de uma incisão cirúrgica no crânio para implantar esses eletrodos (Figura 5.11). Tipicamente, uma grade ou tira de $m \times n$ eletrodos é implantada, em que $1 \leq m, n \leq 8$. Eletrodos da ECoG podem ter ponta de carbono, platina ou liga de ouro e são tipicamente de 2-5 mm de diâmetro. O espaço entre os eletrodos na grade é usualmente de 10 mm a 1 cm. Os eletrodos são flexíveis o suficiente para acomodar movimentos normais do cérebro [57].

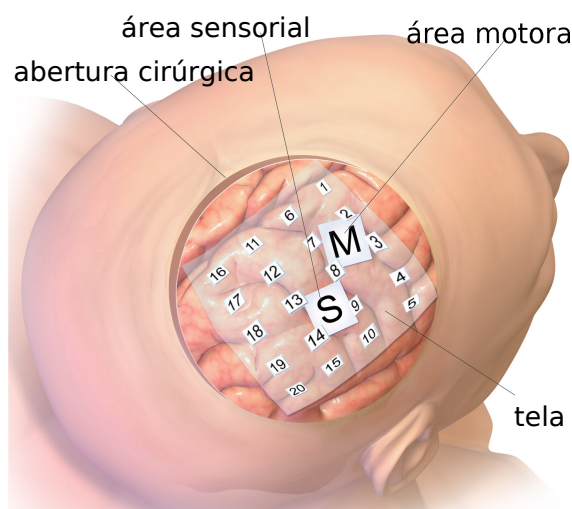


Figura 5.11. ECoG em humanos [7].

Ao contrário dos eletrodos de única célula e das matrizes de microeletrodos, os eletrodos de ECoG podem gravar flutuações elétricas causadas por atividade coerente de grandes populações de neurônios (várias dezenas de milhares). Enquanto os eletrodos de ECoG não medem diretamente *spikes*, o sinal gravado é diretamente relacionado às correntes de entrada recebidas pelos dendritos de neurônios corticais, particularmente nas camadas superiores do cortex cerebral.

A ECoG recentemente tem recebido atenção da comunidade de BCI como uma técnica parcialmente invasiva, entre matrizes multieletrodo invasivas e a EEG não invasiva. Ao contrário das matrizes multieletrodo, algumas formas de ECoG não penetram a barreira sangue-cérebro e são, portanto, mais seguras que matrizes implantadas dentro do cérebro [57].

Por causa de sua proximidade à atividade neural, a ECoG oferece [57]: (1) resolução espacial maior do que técnicas não invasivas como a EEG (dezenas de milímetros versus centímetros), (2) maior largura de banda espectral (0-200 Hz versus 0-40 Hz), (3) amplitude maior (50-100 μV versus dezenas de μV), e (4) consideravelmente menos vulnerabilidade a artefatos tais como atividade muscular e ruído ambiente.

As limitações da ECoG incluem [57]: (1) pode ser usada apenas em instalações cirúrgicas, (2) apenas porções cirurgicamente relevantes do cérebro podem ser gravadas,

e (3) pode haver interferência devido a drogas ou condições relacionadas ao paciente, tal como crises.

Uma desvantagem da ECoG é o tamanho relativamente grande de seus eletrodos (vários mm de diâmetro). Isso está sendo resolvido usando eletrodos microECoG, que são apenas uma fração de um milímetro de diâmetro e espaçados apenas de 2-3 mm em uma grade, permitindo a detecção da atividade neural em uma resolução muito melhor que a da ECoG tradicional. Isso abre a possibilidade de decodificar movimentos finos, como os movimentos de dedos individuais ou mesmo a fala, sem realmente penetrar o cérebro.

5.5.1.2. Técnicas não invasivas

Um pouco menos espetacular, mas também notável e de progresso constante, tem sido feito no nível mesoscópico intermediário [16, 18] na análise de EEG [22]. A EEG mede a atividade elétrica do cérebro em diferentes locais da cabeça, geralmente usando eletrodos colocados no couro cabeludo (veja Figura 5.3).

A EEG registra diferentes faixas de frequências [22, 57]:

- *ritmos α* (7,5-12,5 Hz): aparecem espontaneamente em adultos normais durante vigília, sob relaxamento e condições de inatividade mentais. São melhores vistos com os olhos fechados, mais pronunciados nos locais occipitais.
- *ritmos β* (12,5-30 Hz): são melhores definidos em locais centrais e frontais, com menor amplitude que ondas α . São reforçados em cálculos mentais, expectativa ou tensão ao longo de toda a superfície do couro cabeludo.
- *ritmos θ* (3,5-7,5 Hz): são típicos durante o sono profundo. Desempenham um papel importante na infância. No adulto acordado, alta atividade θ é considerada anormal e relacionada com distúrbios cerebrais, como a epilepsia.
- *ritmos δ* (0,5-3,5 Hz): também são característicos de estágios de sono profundo. Dependendo de sua morfologia, localização e ritmicidade, oscilações δ podem ser normais, tal como durante o sono de ondas lentas ou patológicas como em tumores cerebrais.
- *ritmos γ baixos* (30-60 Hz em EEG humano, 30-80 Hz em ECoG de animais): vários trabalhos têm mostrado correlações de atividade γ com diferentes processos sensoriais e cognitivos, notavelmente durante percepção visual, auditiva, somática e olfativa, bem como com a atenção.
- *ritmos γ altos* (variadamente definidos entre 80-120+ Hz): também chamados de ritmos ϵ , têm sido encontrados em ECoG humano e animal, em associação com potenciais de ação disparados em sequências regulares (“chattering”).

As principais vantagens da EEG sobre as outras técnicas de gravação são a sua *alta resolução temporal* e o fato de que pode ser gravado de forma *não invasiva* (por exemplo, sem a necessidade de uma cirurgia). Devido ao seu *custo relativamente baixo*, gravações

de EEG são amplamente utilizadas tanto em ambientes clínicos quanto em laboratórios de pesquisa. Isso faz da EEG uma ferramenta muito útil e acessível, o que é particularmente interessante para a análise dos processos cerebrais de alto nível, que surgem a partir da atividade de grupo de grandes *populações de neurônios*. Tais processos podem ser bem localizados no tempo, ou eles podem ser correlacionados com os padrões de variação de tempo, como oscilações cerebrais, que estão além da resolução de tempo das técnicas de imagem como a fMRI.

A desvantagem de EEGs não invasivas é o fato de que elas refletem a atividade média de um grande número de fontes distantes dos locais de gravação e, portanto, elas não têm uma resolução espacial ideal. Embora a forma de gravar sinais de EEG não se alterou, tanto quanto no caso de gravações microscópicas e macroscópicas, tem havido avanços significativos na metodologia de análise de dados de EEG. Na verdade, as gravações de EEG têm sido um desafio final para a maioria dos métodos de processamento de sinal, devido à sua alta complexidade, baixa relação sinal-ruído, não-linearidade e não-estacionariedade, características que tornam o processamento de sinais de EEG uma tarefa extremamente difícil [51].

O desenvolvimento e implementação de novos algoritmos que são especificamente projetados para sinais complexos como os EEG nos permitirá obter muito mais informações das que têm sido acessíveis com métodos anteriores e com a inspeção visual convencional de gravações, como é feita por eletroencefalógrafos treinados. Estes métodos abrem uma nova janela para o estudo de processos cognitivos de alto nível em seres humanos com técnicas não invasivas e sem grande despesa [57].



Figura 5.12. Exemplo de sistema MEG [73].

A *magnetoencefalografia* (MEG) mede os campos magnéticos produzidos pela atividade elétrica no cérebro. A Figura 5.12 mostra a instalação de uma MEG típica na qual um sujeito senta em uma cadeira e responde a estímulos em uma tela apertando botões em um dispositivo. Tanto sinais MEG quanto EEG originam do efeito de rede de correntes iônicas que fluem para os dendritos dos neurônios devido às entradas sinápticas provenientes de outros neurônios. Essas correntes produzem um campo magnético

orientado ortogonalmente (como ditado pela equações de Maxwell) [57].

Como a EEG, a MEG apresenta resolução temporal alta pois reflete diretamente a atividade neural, ao invés da atividade metabólica como é o caso de técnicas como a fMRI ou PET. Uma vantagem sobre a EEG é que os campos magnéticos produzidos pela atividade neural não são distorcidos pela matéria orgânica intermediária (como o crânio e o couro cabeludo), como é o caso dos campos elétricos medidos pela EEG. Portanto, a MEG tem resolução espacial melhor que a EEG e é independente da geometria da cabeça. Por outro lado, os sistemas MEG são consideravelmente mais caros que os sistemas de EEG, grandes, pesados e não portáteis e necessitam de uma sala isolada magneticamente para barrar interferências de sinais magnéticos externos, incluindo o campo magnético da Terra.

A imagem de ressonância magnética funcional (*functional magnetic resonance imaging* - fMRI) mede indiretamente a atividade neural no cérebro detectando mudanças no fluxo do sangue devido à atividade aumentada dos neurônios em regiões do cérebro durante tarefas específicas (Figura 5.13). Quando os neurônios tornam-se ativos, eles consomem mais oxigênio, trazido ao cérebro pela corrente sanguínea. A atividade neural engatilha uma dilatação das capilaridades locais resultando em um aumento no fluxo de entrada de sangue altamente oxigenado que substitui o sangue pobre em oxigênio. Esta resposta hemodinâmica é comparativamente lenta - aparece várias centenas de milisegundos depois da atividade neural e tem picos em 3-6 segundos, antes de voltar à linha de base em outros 20 segundos.

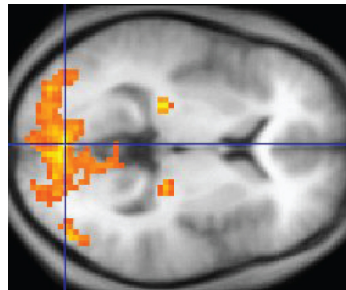


Figura 5.13. Uma imagem fMRI com áreas em amarelo mostrando atividade aumentada [75].

A tomografia de emissão de positron (*positron emission tomography* - PET) é uma técnica mais antiga para medir a atividade do cérebro indiretamente através da detecção da atividade metabólica. A PET mede as emissões de substâncias químicas radioativamente rotuladas e metabolicamente ativas que foram injetadas no fluxo sanguíneo de transporte ao cérebro (Figura 5.14).

5.5.2. Gravação e estimulação

Apesar da maioria das BCIs atuais apenas gravarem a atividade neural para controlar dispositivos e prover realimentação visual ou tátil, alguns pesquisadores estão explorando a possibilidade de, simultaneamente, gravar sinais neurais e prover realimentação direta por meio da estimulação neural. Duas abordagens possíveis que estão sendo exploradas incluem usar *matrizes de microeletrodos* e chips implantáveis mais sofisticados, como o

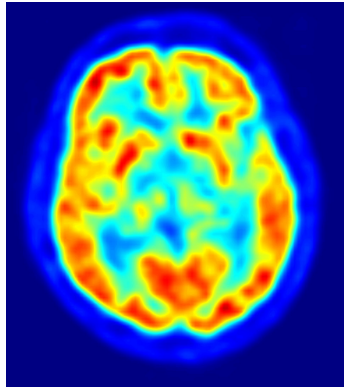


Figura 5.14. Scan PET do cérebro humano [74].

Neurochip, que implementa processamento de sinais e outros algoritmos, processamento da atividade neural e envio de estimulação dentro do próprio chip ao invés de depender de um computador.

Como já foi dito, os microeletrodos usados para gravação da atividade de *spiking* podem também ser usados para enviar corrente despolarizadora ou hiperpolarizadora para excitar ou inibir neurônios. Portanto, em uma matriz multieletrodo, alguns eletrodos podem ser separados para gravação e outros podem ser usados para estimulação.

O Neurochip é um exemplo de um circuito integrado que grava um ou mais neurônios, realiza processamento de sinal útil e outras computações e, baseado nos resultados dessas computações, envia estimulação adequada a um ou mais neurônios [47]. O chip é portanto uma unidade auto-contida, permitindo ao sujeito implantado circular livremente e se comportar naturalmente. O chip alimentado por bateria tem uma matriz de doze eletrodos de microfio de tungstênio móveis (diâmetro de $50 \mu m$; impedância de $0.5 M\Omega$; espaço intereletrodo de $500 \mu m$).

O chip contém um microprocessador que pode realizar o *spike sorting*⁴ em sinais de um conjunto de eletrodos e instruir um circuito estimulador a enviar pulsos elétricos via outro conjunto de eletrodos. Pequenos segmentos de sinais gravados e padrões de estimulação desejada podem ser armazenados na memória *on-chip*. O Neurochip tem sido usado em macacos para demonstrar que a ativação consistente de um grupo de neurônios em correlação com a ativação de um outro pode causar um fortalecimento das conexões entre os dois grupos de neurônios.

5.5.2.1. Gravações

Gravações de EEG no couro cabeludo são realizadas utilizando eletrodos de condutância elevadas colocados no topo da cabeça. Eletrodos são distribuídos em locais específicos, tipicamente utilizando o sistema 10-20, em que de 16-20 eletrodos estão separados por

⁴*Spike sorting* é o agrupamento de *spikes* em clusters baseado na similaridade de suas formas. Dado que, em princípio, cada neurônio tende a disparar *spikes* de um determinado formato, os clusters resultantes correspondem à atividade de diferentes neurônios. O resultado final é a determinação de qual *spike* corresponde a qual desses neurônios.

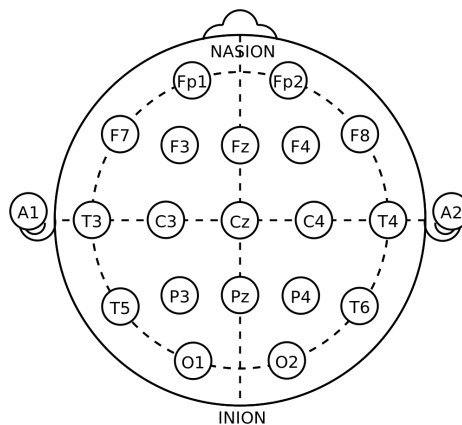


Figura 5.15. Sistema de EEG 10-20.

10-20% da distância total em torno da circunferência da cabeça (embora seja cada vez mais comum o uso de 32, 64, 128, ou 256 eletrodos) (Figura 5.15). Os eletrodos são colocados sobre a cabeça com gel a fim de aumentar a condutividade com o couro cabeludo. No início, os eletrodos eram colocados manualmente um por um, mas agora eles normalmente já vêm posicionados em um gorro, que pode ser facilmente encaixado em torno da cabeça do sujeito (veja a Figura 5.3). Isto é mais prático e menos demorado, considerando as possibilidades de equipamentos modernos, que permitem a gravação simultânea de mais de 200 canais [57].

A EEG pode ser gravada com referência a um eletrodo passivo comum - gravações monopolares (referenciais) - ou pode ser gravada diferencialmente entre pares de eletrodos contíguos - gravações bipolares. No último caso, existem várias maneiras de escolher os pares de eletrodos de acordo com montagens concebidas para visualizar a propagação de atividade em diferentes direções. Algumas montagens particulares podem ser muito úteis para visualizar as fontes de diferentes padrões de EEG.

Isto é o caso de pulsos cujas localizações, geralmente administrados por uma inversão de polaridade do sinal, podem ajudar no estudo de pacientes com epilepsia. Deve-se notar, no entanto, que a utilidade das diferentes montagens hoje em dia é limitada a uma visualização online dos dados (que costumavam ser armazenados em papel), uma vez que diferentes derivações podem ser calculadas *off-line* com o uso de computadores. Sinais de EEG são gravados com uma frequência de amostragem de 100 Hz ou superior. Sistemas de aquisição modernos podem lidar facilmente com altas taxas de amostragem, e é agora usual gravar EEGs com uma frequência de amostragem de 500 Hz ou maior para permitir o estudo de oscilações de alta frequência ou transições rápidas entre os diferentes eletrodos [57].

5.5.2.2. Artefatos

Devido à sua amplitude muito baixa, sinais de EEG são facilmente contaminados por fontes externas. Estes “artefatos,” inerentes de gravações de EEG do couro cabeludo, são produzidos por movimentos da cabeça, piscar dos olhos, eletrocardiograma, atividade

muscular, etc. Por exemplo, um artefato produzido pelo piscar pode parecer uma verdadeira atividade do cérebro para o olho ingênuo, mas um pesquisador especialista em EEG o reconhecerá facilmente como um artefato do piscar devido à sua morfologia e localização espacial nas posições frontais [57].

Piscar de olhos pode ser melhor identificado, colocando eletrodos perto dos olhos para medir eletrooculogramas. Outros tipos de artefatos são gerados por atividade muscular. Artefatos musculares podem, por exemplo, serem gerados por uma postura tensa, que normalmente está correlacionada com atividade de alta frequência. Estes artefatos podem ser eliminados, em princípio, por meio da utilização de filtros digitais padrões, mas, infelizmente, em alguns casos, isto não é possível porque coincidem com as frequências de interesse. Movimentos da cabeça estão correlacionados com atividade de frequência baixa, e neste caso, o EEG tipicamente mostra uma linha de base flutuante.

Esta atividade pode ser eliminada com um filtro passa-alta (normalmente fixado em 1 Hz), mas novamente, tal filtragem só é adequada quando frequências baixas não são de interesse. Um ponto importante é que os artefatos geralmente limitam a duração das gravações de EEG que podem ser consideradas como estacionárias (isto é, segmentos nos quais as características principais do sinal, tais como a sua média, variância, e espectro de potência não mudam). Com efeito, a duração relativamente curta de gravações de EEG estacionárias é um dos principais desafios para a sua análise [57].

O EEG do couro cabeludo pode ser gravado de forma não invasiva a um baixo custo relativo e tornou-se uma ferramenta de diagnóstico padrão na prática clínica. Em casos muito particulares, eletrodos são colocados dentro da cabeça para realizar gravações intracranianas. Isto é feito, por exemplo, em pacientes que sofrem de epilepsia refratária a medicamentos que são candidatos a cirurgia de epilepsia. O objetivo de gravações intracranianas em pacientes com epilepsia é localizar precisamente a área de início das crises de epilepsia, a fim de avaliar uma eventual ressecção cirúrgica do foco [57].

5.5.3. Potenciais evocados

Em muitos campos científicos, especialmente em física e engenharia, uma maneira muito útil de aprender sobre um sistema é estudar suas reações a perturbações. Na pesquisa do cérebro, é também uma estratégia comum para ver como os neurônios individuais ou grande conjunto de neurônios, conforme registro pela EEG, reagem a diferentes tipos de estímulos. *Potenciais evocados* (*evoked potentials* - EPs) são as alterações na atividade de EEG em curso, devido à estimulação. Eles também são usados como entradas bem definidas em estudos de ECoG. Eles são bloqueados no tempo para o estímulo e possuem um padrão característico de resposta que é mais ou menos reproduzível sob condições experimentais similares [22].

Os EPs são caracterizados pela sua polaridade e latência, por exemplo, *P100* significa uma deflexão positiva (*P* para positivo) que ocorre *100* ms após a estimulação. A gravação de EPs é feita da mesma maneira como o EEG, com o sistema de entrega do estímulo enviando disparadores para o sistema de aquisição, a fim de identificar os inícios e remissões do estímulo. Os EPs podem ser classificados como exógenos e endógenos. Exógenos são os obtidos pelas características físicas do estímulo externo, o qual é tipicamente visual, auditivo ou somatossensorial. EPs endógenos são extraídos por processos

cerebrais internos e respondem à significância do estímulo. EPs endógenos podem ser usados para estudar processos cognitivos.

Os EPs visuais são geralmente evocados por flashes de luz ou padrões visuais tais como um tabuleiro de xadrez. Há uma deflexão positiva a cerca de 100 ms após a apresentação do estímulo (P100), seguido por uma recuperação negativa em 200 ms (N200). Estes picos são mais bem definidos nos eletrodos occipitais, que são os mais próximos da área visual primária. O P100 também é observado nos eletrodos centrais e frontais mas não tão bem definidos e aparecendo mais tarde do que nos locais posteriores. EPs visuais podem ser utilizados na prática clínica para identificar lesões na via visual, tais como aqueles causados por neurite óptica e esclerose múltipla.

5.5.4. Potenciais relacionados a eventos (ERP)

Normalmente, o termo EP refere-se a respostas de EEG à estimulação sensorial. Sequências de estímulos podem ser organizadas em paradigmas e sujeitos podem ser convidados a executar diferentes tarefas. Potenciais relacionados a eventos (*event-related potentials* - ERPs) constituem uma categoria mais ampla de respostas que são desencadeadas por “eventos,” tais como o reconhecimento de um estímulo “alvo” ou a ausência de um estímulo em uma sequência (Figura 5.16).

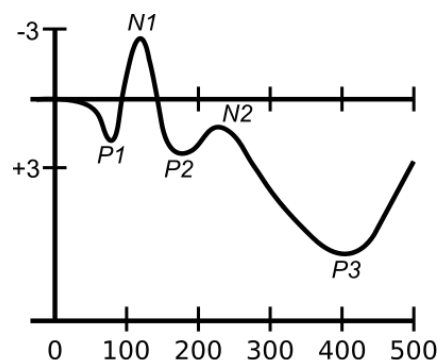


Figura 5.16. Uma forma de onda mostrando vários componentes ERP, incluindo o N100 e o P300. Note que o ERP é plotado com voltagens negativas crescentes, uma prática comum na pesquisa. No eixo x tem-se o tempo depois do estímulo (em ms) e no eixo y tem-se o potencial (em μV) [70].

O método mais comum para provocar ERPs é usando o *paradigma oddball*. Dois estímulos diferentes são distribuídos pseudo-aleatoriamente em uma sequência: um deles aparecendo frequentemente (estímulo padrão) e o outro sendo um estímulo alvo aparecendo com menos frequência e de forma inesperada. Estímulos padrão e alvo podem ser tons de diferentes frequências, ou figuras de cores diferentes, formas diferentes, etc. Os sujeitos são normalmente solicitados a contar o número de aparições alvo em uma sessão ou a pressionar um botão sempre que um estímulo alvo aparece [57].

A resposta ao estímulo não-alvo é qualitativamente semelhante a resposta a EPs visuais (em que não havia nenhuma tarefa). Tal como no caso dos EPs visuais padrão, o complexo P100-N200 pode ser observado tanto com estimulação não-alvo quanto com estimulação alvo. Estes picos estão essencialmente relacionados com o processamento sensorial primário, devido ao fato de eles não dependerem da tarefa, eles têm uma latên-

cia relativamente curta (100 ms), e eles são melhores definidos na área visual primária (lobo occipital). A estimulação alvo levou a um componente marcado positivo, o P300, aparecendo entre 300 e 500 ms, e é mais marcado nos locais centrais e posteriores.

Embora a localização do P300 no couro cabeludo seja bem conhecida, a localização das fontes do P300 no cérebro ainda é controversa. Uma vez que o P300 é dependente de tarefa, como tem uma latência relativamente longa, ele é tradicionalmente relacionado com os processos cognitivos tais como comparação do sinal, reconhecimento, tomada de decisão, atenção e atualização de memória. Muitos pesquisadores utilizaram o P300 para estudar processos cognitivos. Respostas P300 anormais podem refletir patologias em que a cognição é prejudicada, como tem sido mostrado em depressão, esquizofrenia, demência e outras.

O P300 também pode ser induzido usando um paradigma *oddball* passivo (isto é, uma sequência *oddball* sem qualquer tarefa). Neste caso, uma resposta parecida com P300 aparece sobre a estimulação alvo, refletindo a novidade do estímulo, em vez da execução de uma certa tarefa. Esta resposta foi chamada de P3a. É mais precoce do que o P300 clássico (também chamado P3b) e maior em áreas frontais e centrais.

5.6. Processamento de Sinais

5.6.1. Análise no domínio da frequência: transformada de Fourier

Jean Baptiste Joseph Fourier (1768-1830), um brilhante matemático francês, realizou estudo sobre a condução de calor. Levou ele, no entanto, quase duas décadas para publicar este trabalho, principalmente devido à proposta de uma nova - e no momento controversa - maneira de decompor sinais periódicos em somas ponderadas de funções seno e cosseno. Esta decomposição, nos dias atuais conhecida como *séries de Fourier*, foi sua maior contribuição para a ciência, em grande parte, transcendendo a sua aplicação original para a condução de calor [22].

Seguindo as idéias de Fourier, sinais como os registrados a partir de eletrodos de superfície de EEG do couro cabeludo, podem ser representados no *domínio do tempo* ou, alternativamente, em termos da sua decomposição em senos e cossenos no *domínio da frequência*. Tome-se por exemplo o sinal oscilatório da Figura 5.17a e suponha que se queira transmiti-lo a outra pessoa. Pode-se, em princípio, ditar todos os pontos de tempo da senóide um por um, ou alternativamente pode-se apenas dizer que é uma senóide com uma frequência de 10 Hz (ou seja, um ciclo repetindo-se a cada 100 ms), conforme representado no gráfico da frequência do painel do lado inferior esquerdo. Essas duas visões parecem análogas, embora também seja possível dizer que a representação em frequência parece ser mais compacta e simples.

Tome agora o exemplo da Figura 5.17b do lado direito. É muito difícil conseguir uma compreensão deste sinal a partir da representação do tempo no desenho superior. No entanto, a representação de frequência no desenho inferior fornece uma boa ideia da sua natureza: é apenas a superposição de três senóides de frequências diferentes. Este exemplo simples ilustra a idéia de porque usar representações de frequência. Basicamente, tentou-se obter uma imagem mais simples de algumas das características básicas do sinal, que são geralmente não óbvias a partir de representações de tempo ruidosas e

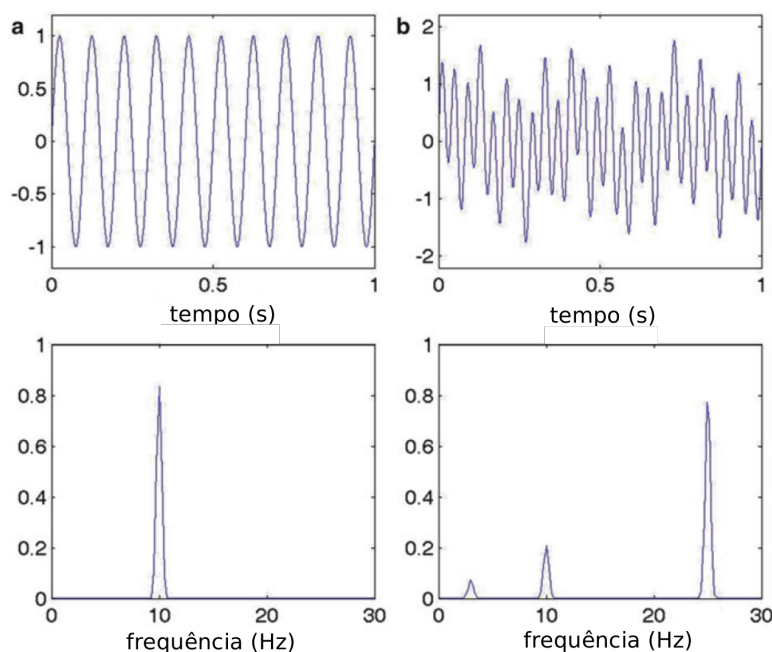


Figura 5.17. (a) Um sinal senoidal no domínio do tempo (acima) e da frequência (abaixo). (b) Um sinal quase-periódico. Neste caso, a transformada de Fourier fornece uma representação mais simples. No eixo das ordenadas (y) estão representadas as amplitudes dos sinais. Adaptado de [23].

complexas [22].

A representação de frequência de um sinal é dada por sua transformada de Fourier (*Fourier transform* - FT), que tem inúmeras aplicações em diferentes disciplinas científicas. No caso específico de sinais de EEG, é de longe a ferramenta quantitativa mais utilizada, especialmente após a introdução de um algoritmo eficiente e rápido para calculá-la, o *Fast Fourier Transform* (FFT) [12].

5.6.2. Análise no domínio do tempo-frequência

Sabe-se que uma das principais limitações da FT é que ela não tem resolução de tempo. Para calcular a FT, supomos que o sinal é estacionário e, conseqüentemente, que a atividade em diferentes frequências é constante ao longo de todo o sinal. Em muitas ocasiões, no entanto, os sinais têm características variáveis no tempo que não podem ser resolvidas com a FT. Este é o caso da música, discurso, sons de animais, dados de radar, e muitos outros sinais. Para sinais de EEG, esta limitação é crítica quando analisamos os processos que mudam no tempo, tais como a resposta a um determinado estímulo ou o desenvolvimento de uma crise de epilepsia.

Intuitivamente, podemos superar a falta de resolução de tempo da FT cortando os dados em pedaços e, em seguida, calculando o espectro de potência para cada pedaço ou, ainda melhor, usando uma janela de evolução no tempo para se concentrar em diferentes segmentos de dados. Este procedimento chama-se transformada de Fourier de curta duração (*short-time Fourier transform* - STFT) ou FT de janelas. Se a janela utilizada é uma gaussiana, é chamada transformada de Gabor, em honra de Denis Gabor, físico húngaro que foi o primeiro que desenvolveu essas ideias para analisar as variações de frequên-

cia de sons. Com o STFT é possível acompanhar a evolução no tempo das frequências diferentes e a exigência de estacionaridade é satisfeita ao considerar que os sinais são quase estacionários dentro de cada janela. De acordo com o *princípio da incerteza de Heisenberg*, é possível conseguir resolução de tempo a partir da FT por janelamento do dado.

Há ainda um ponto crítico a ser discutido para a implementação da STFT. Temos de decidir como escolher o tamanho da janela, ou mais especificamente, a taxa de decaimento da função gaussiana. A má notícia é que isto não é tão simples e, novamente, um compromisso deve ser considerado. Se a janela é muito estreita, ela vai dar uma resolução de tempo boa, mas as frequências não serão bem resolvidas (a resolução de frequência é inversamente proporcional ao comprimento dos dados). Pelo contrário, se a janela é muito grande teremos uma resolução de frequência boa, mas a localização no tempo será perdida. Há um compromisso entre a frequência e a resolução de tempo.

Em analogia com o princípio da incerteza de Heisenberg da mecânica quântica (que estabelece que não é possível determinar com acurácia a posição e a velocidade de uma dada partícula ao mesmo tempo), isso é chamado de princípio da incerteza da análise do sinal: frequência e resolução de tempo não podem ser feitos arbitrariamente pequenos, ao mesmo tempo. Em outras palavras, as localizações exatas no tempo e na frequência são mutuamente exclusivas porque precisamos de vários pontos de dados para definir uma frequência [22].

5.6.3. Análise de Ondaletas (*wavelets*)

Mostrou-se que uma questão-chave com a transformada de Fourier de curta duração (STFT) é a escolha do comprimento da janela, tendo em conta a limitação básica imposta pelo princípio da incerteza de análise de sinais. Janelas curtas fornecem boa resolução de tempo (mas resolução de frequência ruim) e, inversamente, longas janelas fornecem boa resolução de frequência (mas resolução de tempo ruim).

No final de 1970, Jean Morlet, um geofísico que trabalhava para uma empresa petrolífera francesa, percebeu que a STFT não era adequada para o estudo de seus dados sísmicos. Ele observou que um bom compromisso entre resolução de tempo e de frequência não foi possível porque os padrões de alta frequência tinham uma duração mais curta em comparação com os de baixa frequência. Assim, uma única janela para todas as frequências não funcionaria. Sua solução foi bastante simples: ele apenas tomou diferentes tamanhos de janela para diferentes frequências, ou mais precisamente, ele tomou uma função cosseno limitada por uma gaussiana (uma função de Gabor) e comprimiu-a ou esticou-a no tempo para obter as diferentes frequências. Em seguida, em vez de sempre ter o mesmo tamanho da janela, ele tinha a mesma forma de onda em diferentes escalas, isto é, com um tamanho variável. Com este truque simples, ele criou a base das ondalletas (*wavelets*)! Em nossos dias, ondalletas são reconhecidas como uma ferramenta muito poderosa de decomposição de sinal com um grande número de aplicações em diferentes campos, incluindo a análise de sinais do cérebro [22].

5.6.4. Análise no domínio do tempo - filtragem bayesiana

Os métodos no domínio de tempo discutidos não mantêm explicitamente estimativas das propriedades do sinal computado no tempo. A manutenção de uma representação de incerteza pode ser importante em BCI pois ações potencialmente desastrosas baseadas em estimativas pobres podem ser evitadas se a quantidade de incerteza associada com uma estimativa é considerada antes de se chegar a uma decisão. Técnicas de filtragem bayesiana provêm uma metodologia estatisticamente válida para estimar as propriedades do sinal e suas incertezas. Técnicas de filtragem espacial consideram como entradas sinais do cérebro gravados de diferentes posições (ou canais) e os transformam de uma ou várias formas. Possíveis metas incluem melhorar a atividade local, reduzir ruído comum entre canais, diminuir a dimensionalidade dos dados, identificar fontes ocultas, ou achar projeções que maximizem a discriminação entre classes diferentes [22].

5.6.5. Redução de artefatos

Artefatos em BCI são sinais indesejáveis que originam de fora do cérebro. Por exemplo, nas BCIs baseadas em EEG, encontra-se o ruído de linha de energia de 50/60 Hz e artefatos causados por movimentos de músculos e olhos. Alguns desses artefatos podem ser permitidos ou mesmo explorados como sinais de controle para certas aplicações tal como jogos ou novos modelos de interação humano-computador. Entretanto uma verdadeira BCI deve possuir a habilidade de eliminar ou reduzir tais artefatos de tal forma que os sinais usados para controlar um dispositivo origemem somente do cérebro. Técnicas de processamento de sinal podem ser usadas para esse fim [22].

5.7. Aprendizado de Máquina

O aprendizado de máquina (*Machine Learning* - ML) tem um papel importante no desenvolvimento de BCIs pois provê técnicas que podem aprender a mapear a atividade neural para comandos de controle apropriados. Algoritmos de ML podem ser divididos em duas classes: aprendizado *supervisionado* e aprendizado *não supervisionado*. No *aprendizado supervisionado*, dados de treinamento consistem de um conjunto de entradas e saídas correspondentes. A meta é aprender a função subjacente a partir dos dados de treinamento tal que novas entradas de teste são mapeadas às saídas corretas.

Se as saídas são classes discretas, o problema é chamado *classificação*. Se as saídas são contínuas, o problema é equivalente à *regressão*. Dada a ênfase em descobrir uma função subjacente, o aprendizado supervisionado é também chamado de *aproximação de função*. O *aprendizado não supervisionado*, por outro lado, enfatiza a descoberta de estrutura estatística escondida em dados não rotulados: os dados de treinamento consistem de entradas que são tipicamente vetores de alta dimensão, e a meta é aprender um modelo estatístico que pode ser compacto ou útil para a análise subsequente.

5.7.1. Aprendizado supervisionado

Classificação é o problema de atribuir um de N rótulos a um novo sinal de entrada, fornecidos dados de treinamento rotulados de entradas conhecidas e seus rótulos de saída correspondentes. *Regressão* é o problema de mapear sinais de entrada a um sinal de saída contínuo. Muitas BCIs baseadas em EEG, ECoG e fMRI dependem da classificação para

gerar saídas de controle discretas (e.g., mover um cursor para cima ou para baixo). As BCIs baseadas em gravações neurais, por outro lado, têm predominantemente utilizado regressão para gerar sinais de saída contínuos, tais como sinais de posição e velocidade para um dispositivo protético. Em geral, a escolha de se usar classificação ou regressão dependerá do tipo do sinal do cérebro gravado e do tipo da aplicação controlada.

5.7.2. Classificação

5.7.2.1. Binária

A tarefa do classificador é atribuir rótulos de classe $y \in Y$ a um vetor de características p -dimensional x . O caso mais simples é quando $y = [-1, +1]$, isto é, discriminação entre duas classes (rotuladas -1 e $+1$). Este caso é chamado de *classificação binária*. O problema de classificação binária reduz-se a achar uma fronteira entre as duas classes baseado nos dados de treinamento rotulados - a meta é achar uma fronteira tal que novos dados possam ser classificados corretamente. Os métodos diferem em como esta fronteira é calculada a partir dos dados de treinamento.

As *redes neurais* são inspiradas na biologia e buscam reproduzir as capacidades adaptativas das redes no cérebro na classificação de dados de entrada de uma maneira robusta. Um importante exemplo é o *perceptron* e sua generalização, o perceptron multi-camadas. O perceptron de camada única computa um hiperplano conforme a Eq. (1), em que o vetor w representa os “pesos sinápticos” que conectam as entradas ao neurônio e $-w_0$ representa o limiar de disparo do neurônio.

$$w^T x + w_0 = 0 \quad (1)$$

A saída do perceptron é mostrada pela Eq. (2).

$$y = \text{sign}(w^T x + w_0) \quad (2)$$

A Eq. (2) tem uma interpretação “neural”: a saída do neurônio é baseada no cálculo de uma soma ponderada de suas entradas ($w^T x = \sum_i w_i x_i$) e na comparação desta soma com um limiar $-w_0$; se a soma ponderada é maior que (ou igual a) o limiar $-w_0$, a saída do neurônio é 1 (um *spike*), de outra forma a saída é 0.

O perceptron adapta seus parâmetros de uma maneira *online*: dada uma entrada x e uma saída desejada y^d , se o erro da saída ($y - y^d$) é positivo, os pesos para entradas positivas são diminuídos, os pesos para entradas negativas são aumentados e o limiar é aumentado, todos por uma pequena quantidade.

O efeito de rede desta regra de “aprendizado” é reduzir o erro da saída para entradas similares no futuro. Se o erro da saída é negativo, os pesos para entradas positivas são aumentados, os pesos para entradas negativas são diminuídos e o limiar é diminuído. Ainda que este algoritmo adaptativo inspirado nos neurônios seja simples e elegante, é aplicável apenas a problemas de classificação nos quais os dados são linearmente separáveis.

Perceptrons multi-camadas (MLP) foram propostos como uma generalização não

linear dos perceptrons para dar conta de problemas de classificação mais difíceis. Os MLPs usam uma não-linearidade sigmoideal (“limiar suave”) ao invés da não-linearidade de limiar degrau para suas unidades neurais (Eq. (3)).

$$y = \text{sigmoid}(w^T x + w_0) \quad (3)$$

A saída da função sigmoide é um número entre 0 e 1, com valores próximos de 0 indicando elemento da classe 1 e valores próximos de 1 indicando elemento da classe 2. A razão para a escolha da sigmoide é que se trata de função diferenciável, permitindo uma regra de aprendizado conhecida como *backpropagation*. Redes neurais baseadas em *backpropagation* são bem sucedidas em várias tarefas de classificação incluindo classificação de dados de BCI e são disponíveis em pacotes de software para classificação, como por exemplo o Weka⁵ e o Matlab⁶. Apesar de poderosas, essas redes neurais sofrem do problema do *overfitting* aos dados de treinamento, resultando em generalização pobre. Assim, a técnica mais recente das máquinas de vetor de suporte (*support vector machines* - SVM) são preferidas como algoritmos de classificação em muitas BCIs.

O perceptron seleciona um hiperplano $w^T x + w_0 = 0$ para separar duas classes. Este hiperplano é apenas um entre um número potencialmente infinito de hiperplanos separando as duas classes de entrada. Pode ser mostrado que entre tais hiperplanos, a melhor generalização é alcançada selecionando o hiperplano com a maior separação (“margem”) entre as duas classes separáveis. A SVM é um classificador que acha um hiperplano separador para o qual a margem entre as amostras das duas classes é maximizada. SVMs lineares têm sido usadas em várias aplicações de BCI.

5.7.2.2. Métodos *ensemble*

Métodos *ensemble* para classificação combinam as saídas de vários classificadores (que discordam entre si em algumas entradas de treinamento) para produzir um classificador final com performance de generalização melhor que qualquer um dos classificadores individuais. Os métodos *ensemble* mais populares, *bagging* e *boosting*, trabalham selecionando diferentes subconjuntos dos dados de treinamento para gerar diferentes classificadores e então combinar suas saídas usando alguma forma de eleição.

5.7.2.3. Classificação multi-classe

Os classificadores apresentados até agora foram projetados para classificar dados em uma de duas classes. Em aplicações BCI, o número de sinais de saída desejado é muitas vezes maior que dois, necessitando de métodos para classificação multi-classes. Há várias estratégias para aplicar classificadores binários ao problema multi-classe.

Uma estratégia para classificação multi-classe é treinar vários classificadores binários e usar eleição majoritária. Dadas N_Y classes, um total de $N_Y(N_Y - 1)/2$ classificadores

⁵<http://www.cs.waikato.ac.nz/ml/weka/>

⁶<http://www.mathworks.com/products/matlab/>

binários são treinados, um para cada combinação binária de classes. Para classificação, uma dada entrada é alimentada a cada um destes classificadores e a classe com o maior número de votos - a classe selecionada pelo maior número de classificadores - é selecionada como saída. A desvantagem desta abordagem é o número relativamente grande de classificadores que precisam ser treinados e usados durante a classificação.

5.7.2.4. Avaliação

Em aplicações BCI, como em outras aplicações dos classificadores, é importante avaliar a acurácia e a performance de generalização do classificador escolhido. A seguir são apresentadas algumas das técnicas de avaliação mais importantes.

Quando se avalia a performance, é útil calcular a matriz de “confusão” $M_{N_Y \times N_Y}$, em que N_Y denota o número de classes. As linhas de M representam os rótulos da classe verdadeira e as colunas representam a saída do classificador. O caso da classificação binária ($N_Y = 2$) é mostrada na Tabela 5.1. As quatro entradas na matriz correspondem a: o número de *verdadeiros positivos* (VP) ou classificações positivas corretas, o número de *falsos negativos* (FN) ou classificações positivas perdidas (erros tipo II), o número de *falsos positivos* (FP) ou classificações positivas incorretas (erros tipo I) e o número de *verdadeiros negativos* (VN) ou rejeições corretas. Os elementos da diagonal M_{ii} da matriz representam o número de amostras classificadas corretamente. Os elementos fora da diagonal M_{ij} mostram quantas amostras da classe i foram classificadas erradamente como classe j .

Tabela 5.1. Matriz de confusão para problemas de duas classes [57].

Classe verdadeira	Classificação	
	Positivo	Negativo
Positivo	verdadeiros positivos (VP)	falsos negativos (FN)
Negativo	falsos positivos (FP)	verdadeiros negativos (VN)

Quando se varia algum parâmetro do classificador (e.g., um limiar), obtém-se diferentes números de verdadeiros positivos e falsos positivos. Um gráfico da proporção de verdadeiros positivos versus a proporção de falsos positivos, quando algum parâmetro do classificador é variado, é conhecido como *curva ROC* (*receiver operating characteristic*), um termo com origem na teoria de detecção de sinal.

A *acurácia da classificação ACC* é definida como a razão entre as amostras corretamente classificadas e o número total de amostras. Pode ser derivada da matriz de confusão M como mostrado na Eq. (4).

$$ACC = \frac{VP + VN}{VP + FN + FP + VN} \quad (4)$$

Pode-se então definir a *taxa de erro* como $err = 1 - ACC$. Quando o número de exemplos para cada classe é o mesmo, o *nível aleatório* é $ACC_0 = 1/N_Y$ em que N_Y denota o número de classes.

Uma outra medida de performance útil é o *coeficiente kappa* (Eq. (5)).

$$\kappa = \frac{ACC - ACC_0}{1 - ACC_0} \quad (5)$$

Por definição, o coeficiente kappa é independente do número de amostras por classe e do número de classes. $\kappa = 0$ significa performance de nível aleatório e $\kappa = 1$ significa classificação perfeita. $\kappa < 0$ significa que a performance da classificação é pior que a aleatória.

Para comparar a performance de BCIs, é importante considerar tanto a acurácia quanto a velocidade de uma BCI. Como uma BCI pode ser considerada como um canal de comunicação, pode-se usar ideias da teoria da informação e quantificar a performance da BCI em termos de *taxa de bit* ou *taxa de transferência da informação* (ITR), que é a quantidade de informação comunicada por um sistema por unidade de tempo. Esta medida captura tanto a velocidade quanto a acurácia.

Suponha que uma BCI ofereça N seleções (ou classes) possíveis em cada teste e cada classe tenha a mesma probabilidade de ser aquela que o usuário deseja. Suponha também que a probabilidade P da classe desejada ser realmente selecionada é sempre a mesma (note que $P = ACC$). Cada uma das outras (não desejadas) classes tem a mesma probabilidade de ser selecionada (isto é, $(1 - P)/(N - 1)$).

Então, usando ideias da teoria da informação, pode-se expressar a ITR (ou taxa de bit) como na Eq. (6), medido em bits/teste (dividindo B pela duração do teste em minutos fornece a taxa em bits/min).

$$B = \log_2(N) + P \log_2(P) + (1 - P) \log_2(1 - P)/(N - 1) \quad (6)$$

Para se estimar a taxa de erro err os classificadores são testados nos *dados de teste* que são diferentes dos dados usados para treinar o classificador. Uma abordagem é particionar simplesmente um conjunto de dados de entrada fornecido em dois subconjuntos, um para treinamento e um para teste (o método *hold out*), mas esta estratégia é sensível à forma como os dados são divididos. Uma estratégia mais sofisticada é a *validação cruzada de K pastas*: o conjunto de dados é separado em K subconjuntos de tamanhos aproximadamente iguais, dos quais $K - 1$ são usados para treinar o classificador e o subconjunto remanescente é usado para teste. O classificador é treinado e testado K vezes, resultando em K diferentes taxas de erro err_k .

A taxa de erro total é calculada pela média dos err_k individuais (Eq. (7)).

$$err = \frac{1}{K} \sum_{k=1}^K err_k \quad (7)$$

Existem variações do procedimento acima. Por exemplo, validação cruzada *leave-one-out* é uma forma extrema da validação cruzada de K pastas em que K é feito igual ao número de amostras de treinamento. Em uma outra variação que procura minimizar os efeitos de determinadas partições dos dados, a validação cruzada de K pastas é repetida N

vezes, dando $N \cdot K$ taxas de erro individuais err_i , com a taxa de erro final sendo a média sobre estes $N \cdot K$ valores.

Em muitas aplicações, é comum separar o conjunto de dados de treinamento em três subconjuntos: um subconjunto de treinamento para achar os parâmetros do classificador, um subconjunto de validação para ajustar estes ou outros parâmetros do classificador e um subconjunto de teste para reportar a performance do classificador otimizado. Ainda que esses procedimentos sejam computacionalmente custosos, eles desempenham um papel importante no melhoramento da habilidade de generalização do classificador.

5.7.3. Regressão

Foi visto na seção anterior que a classificação envolve mapear entradas a uma de um número finito de classes. Isto pode ser considerado como um caso especial do problema de aproximação de função em que a saída é discreta. Quando a saída é contínua, isto é, um valor escalar real ou vetor, o problema é equivalente à *regressão*. Como foi no caso da classificação, é fornecido um conjunto de treinamento de N pares de vetores exemplo (u^m, d^m) , em que $m = 1, \dots, N$ e seria desejável aprender uma função que mapeia arbitrariamente vetores de entrada a saídas apropriadas. Discutir-se-á inicialmente a forma mais simples de regressão, a *regressão linear* antes de se estudar métodos de regressão não lineares e probabilísticos.

5.7.3.1. Regressão linear

A regressão linear assume que a função subjacente geradora dos dados é linear, isto é, o vetor de saída é uma função linear do vetor de entrada. Considere o caso especial em que a entrada u é um vetor K -dimensional (e.g., taxas de disparo de K neurônios) e a saída v é um valor escalar (e.g. posição final do efetor). Então, a saída é dada pela função linear da Eq. (8), em que w é o vetor de “pesos” ou *filtro linear* que é necessário determinar a partir dos dados de treinamento.

$$v = \sum_{i=1}^K w_i u_i = w^T u \quad (8)$$

5.7.3.2. Redes neurais e *backpropagation*

As redes neurais têm sido algoritmos populares para aproximação de função não linear desde a (re-)descoberta do algoritmo de aprendizado *backpropagation* (BP). Vai-se rever redes neurais sigmóides multi-camadas para regressão não linear e derivar o algoritmo BP. Quando técnicas de classificação foram discutidas, foi apresentado o perceptron, um tipo de rede neural em que cada “neurônio” utiliza uma função de saída limiar em uma soma ponderada de suas entradas [30]. A função limiar é útil para classificação mas para a regressão não linear, uma escolha popular é a função de saída *sigmoide* (ou *logística*) (Eqs. (9) e (10)).

$$v = g(w^T u) \quad (9)$$

$$g(x) = \frac{1}{1 + e^{-\beta x}} \quad (10)$$

A função sigmoide pode ser vista como uma versão mais suave da função limiar: ela espreme suas entradas ao intervalo 0 a 1, com o parâmetro β controlando a inclinação da função (valores maiores de β empurram a sigmoide para perto de uma função limiar). A sigmoide é também facilmente diferenciável, o que se torna importante quando se deriva a regra de aprendizado BP.

Para a regressão não linear, são interessantes redes contendo múltiplas camadas de neurônios, em que a saída de uma camada é alimentada como entrada da próxima camada de neurônios. O tipo mais comum de uma rede multi-camadas é uma rede de três camadas contendo uma camada de entrada, uma camada “escondida” e uma camada de saída.

Pode ser mostrado que, pelo menos teoricamente, tais redes podem aproximar funções não lineares arbitrárias, desde que haja neurônios suficientes na camada escondida. Foca-se portanto em tais redes (com uma única camada escondida).

Suponha que se tenha uma rede de três camadas (uma escondida) de neurônios sigmoidais, com a matriz V representando os pesos da camada de entrada para a camada escondida e a matriz W representando os pesos da camada escondida para a camada de saída. A saída do i -ésimo neurônio na camada de saída pode ser descrita como na Eq. (11).

$$v_i = g\left(\sum_j W_{ji} g\left(\sum_k V_{kj} u_k\right)\right) \quad (11)$$

Como no caso da regressão linear, a meta é mais uma vez minimizar o erro entre o vetor saída desejada nos dados de treinamento e o vetor de saída real produzido pela rede. O algoritmo BP foi proposto como uma solução a este problema.

Pode ser visto que os erros de saída ($d_i - v_i$), em que d_i representa a saída desejada, influenciam a atualização da camada interna de pesos e são apropriadamente modulados por derivadas da função de ativação não linear (a sigmoide) em cada camada. Os erros são então “propagados de volta” para a camada mais baixa. Este procedimento de aprendizado pode ser generalizado para um número arbitrário de camadas, incluindo as redes “profundas” (*deep*) contendo um grande número de camadas escondidas, ainda que tais redes sejam propensas ao *overfitting* dos dados de treinamento, resultando em generalização pobre. A maioria das aplicações BCI tendem a usar redes com uma camada escondida tais como a descrita, com o número de neurônios na camada escondida determinado usando validação cruzada.

5.8. Aplicações

5.8.1. Respostas úteis do cérebro

Uma das propriedades mais importantes dos circuitos neurais é sua plasticidade, permitindo respostas dos neurônios para adaptação como uma função das entradas. Em muitos

casos, esta plasticidade é modulada pelas recompensas recebidas pelo animal. Um exemplo comportamental bem conhecido desta plasticidade é o *condicionamento pavloviano* (ou *clássico*): um cachorro que originalmente saliva em resposta a comida, começa a salivar em resposta a uma campainha depois que a campainha é emparelhada de forma consistente com o estímulo da comida. Neste exemplo, a campainha é chamada de estímulo condicionado e a salivação é a resposta condicionada [57].

Em contraste, no *condicionamento instrumental* (ou *operante*), o animal recebe uma recompensa apenas depois de completar uma ação apropriada, e.g., pressionar uma alavanca. Neste caso, depois da recompensa ter sido emparelhada com a ação de pressionar a alavanca, a ação de pressionar a alavanca torna-se a resposta condicionada.

Respostas condicionadas são também vistas em únicos neurônios e em redes. Em uma das primeiras demonstrações de BCI, Eberhard Fetz na Universidade de Washington utilizou a ideia de condicionamento para demonstrar que a atividade de um único neurônio no córtex motor do primata pode ser condicionado para controlar a agulha de um medidor analógico. O movimento da agulha era diretamente acoplado à taxa de disparo do neurônio: quando a agulha cruzava um limiar, o macaco era recompensado.

Depois de vários testes, o macaco aprendeu a consistentemente mover a agulha para passar o limiar aumentando a taxa de disparo do neurônio sob gravação. Este é um exemplo do condicionamento operante em que a ação (movimento da agulha) que produz recompensa é acoplada à atividade aumentada no neurônio sob gravação (a resposta condicionada).

Respostas condicionadas podem também ser obtidas em grandes populações de neurônios. Por exemplo, depois de várias sessões de treinamento, sujeitos humanos podem controlar a potência em bandas de frequência particulares em sinais de EEG gravados do couro cabeludo. Nestes experimentos, a potência é acoplada ao movimento de um cursor em uma tela de computador usando uma função de mapeamento fixo e a meta é mover o cursor em uma direção desejada para atingir um alvo.

O sujeito gradualmente aprende a controlar o movimento do cursor modulando a potência na(s) banda(s) de frequência usada(s) na função de mapeamento. Neste caso, o condicionamento ocorre no nível da população neural e a resposta condicionada envolve as atividades de grande número de neurônios sendo modulados em concerto para gerar o aumento ou diminuição apropriado na potência na banda de frequência desejada.

Em resumo, tanto as respostas de neurônio único quanto de redes de neurônios podem ser moduladas como uma consequência do acoplamento da atividade neural com ações externas (tais como movimentar o cursor) e recompensas que são contingentes à execução de ações apropriadas (atingir um alvo).

Os neurônios no córtex motor primário codificam para vários atributos de movimento tais como a direção do movimento de um membro (braço ou perna), velocidades, forças, etc. Em vários experimentos importantes, Georgopoulos e colegas [27] mostraram que o movimento é representado usando um código de população. Por exemplo, no caso da direção do movimento, os neurônios na população disparam de acordo com o quão perto sua direção preferida de movimento está em relação à direção real do movimento. A direção real do movimento pode ser predita, por exemplo, por uma combinação pon-

derada das direções preferidas dos neurônios, o peso para cada neurônio sendo a taxa de disparo do neurônio. Este método da decodificação da direção do movimento é chamado de decodificação do *vetor de população*.

O fato de que variáveis relacionadas a movimento podem ser extraídas das atividades de populações de neurônios foi um importante achado para a BCI porque permite supor que a mesma atividade motora da população poderia ser usada para controlar o movimento de membros artificiais e outros dispositivos. Algumas das demonstrações mais impressionantes de BCI em animais baseiam-se no uso de técnicas de regressão para mapear a atividade motora da população a sinais de controle apropriados para dispositivos protéticos.

Um terceiro tipo de resposta do cérebro que é muito usado para BCI em humanos é a atividade neural produzida quando um sujeito voluntariamente imagina fazendo um movimento particular (*imagética motora*). Imaginar um movimento produz atividade neural similar espaço-temporalmente à atividade gerada durante o movimento real, mas menor em magnitude. Vários algoritmos de aprendizado de máquina (tipicamente, classificadores) podem ser aplicados para discriminar entre dois ou mais tipos de movimentos imaginados, permitindo cada atividade imaginada ser mapeada a um sinal de controle particular (e.g., mover o cursor para cima). A resposta inicialmente fraca devido ao movimento imaginado torna-se mais robusta quando o sujeito recebe retroalimentação enquanto aprende a controlar o cursor. Em alguns sujeitos, a atividade imaginada durante o controle do cursor pode até exceder a atividade observada durante o movimento real.

Similar aos movimentos imaginados, pode-se pedir a um sujeito humano que realize uma tarefa cognitiva como uma aritmética mental ou a visualização de um rosto. Se as tarefas cognitivas forem suficientemente distintas, as áreas do cérebro que são ativadas também serão diferentes e a ativação resultante do cérebro, medida por exemplo pela EEG, pode ser discriminada usando um classificador treinado com um conjunto inicial de dados coletado do sujeito. Cada tarefa cognitiva é mapeada a um sinal de controle (e.g., a realização da aritmética mental é mapeada para mover o cursor para cima, etc.). A abordagem portanto é baseada fortemente em ser capaz de discriminar com confiança os padrões de atividade para tarefas cognitivas diferentes, fazendo a escolha das tarefas cognitivas uma decisão de projeto experimental importante e complicada.

Outra classe de sinais do cérebro útil para BCI é baseada na atividade estereotipada gerada pelo cérebro em resposta a tipos especiais de estímulos. Um exemplo particularmente importante é o sinal P300 (ou P3) observado em gravações de EEG, chamado assim porque é uma deflexão positiva no sinal de EEG que ocorre aproximadamente 300 ms depois do estímulo. Como já foi visto, o P300 é um exemplo de um “potencial relacionado a evento” (ERP) ou “potencial evocado” (EP) - ele é evocado pela ocorrência de um estímulo raro ou imprevisível. É geralmente observado mais fortemente no lobo parietal, ainda que alguns componentes também originam nos lobos temporal e frontal. Os mecanismos neurais exatos responsáveis pelo P300 ainda não estão claros: várias estruturas do cérebro, como córtex parietal, o giro cingulado e o córtex temporoparietal e estruturas límbicas (hipocampo, amígdala) têm sido implicados como substratos para o P300.

Outros tipos comuns de EP incluem o potencial visualmente evocado de estado estável (SSVEP), o N100 e o N400. O SSVEP é a resposta obtida em populações de

neurônios no córtex visual quando o sujeito está fixando em um estímulo visual (e.g., tabuleiro de xadrez) piscando em uma frequência particular (e.g., 15 Hz). O sinal do cérebro associado, gravado por exemplo, usando EEG, exibe picos no espectro de potência na frequência do estímulo e suas harmônicas. Se frequências diferentes estão associadas com escolhas diferentes, uma BCI pode decodificar a escolha do sujeito detectando onde os picos estão.

O N100 (ou N1) é um potencial negativo que ocorre aproximadamente 100 ms depois de um estímulo imprevisível e é tipicamente distribuído pelas regiões frontal e central da cabeça. É usualmente seguido por uma onda positiva (P200) resultando no “complexo N100-P200”. O N100 ocorre por exemplo em resposta a um barulho alto e repentino, mas não se o som é criado pelo sujeito.

O N400 é um outro exemplo de uma deflexão negativa no potencial que tem pico aproximadamente 400 ms depois de determinados tipos de entradas incongruentes mas potencialmente significativas, tal como palavras inapropriadas semanticamente ocorrendo em uma sentença durante a fala. É tipicamente distribuído nas posições central e parietal do couro cabeludo. O N400 é similar a um outro tipo de potencial chamado de *potencial de erro* (ErrP) evocado quando um erro é observado depois da realização de uma ação.

5.8.2. Aplicações médicas

5.8.2.1. Restauração

O campo da BCI originou do objetivo de ajudar pessoas paralisadas e deficientes. Portanto não é surpresa que algumas das maiores aplicações de BCIs sejam em tecnologia médica, particularmente restabelecendo função sensorial e motora.

Uma das BCIs mais usadas é o implante coclear para os surdos (Figura 5.18). O implante coclear é um exemplo de BCI para restauração sensorial, da mesma forma que implantes retiniais desenvolvidos para os cegos.

Não há muita pesquisa nos outros dois tipos possíveis de BCIs puramente sensoriais, as BCIs para somatosensação e as BCIs para olfato e paladar. No caso do primeiro, a necessidade da BCI é minimizada porque normalmente é possível restaurar a sensação tátil por meio de enxerto de pele. Entretanto, há um interesse considerável em estimulação somatosensorial como um componente de BCIs bidirecionais para permitir que indivíduos paralisados e amputados possam, por exemplo, sentir objetos segurados ou tocados por dispositivos protéticos.

No caso de BCIs para olfato e paladar, têm havido esforços para construir “narizes artificiais” e chips que podem sentir vários tipos de odores, mas estes dispositivos são construídos mais para segurança e robótica e menos para BCIs. A falta de interesse em desenvolver BCIs para olfato e paladar é devida mais à falta de uma grande população de indivíduos necessitando de tais BCIs, comparado à população de pessoas com deficiência visual ou auditiva [57].

Uma outra grande motivação para pesquisa de BCI nas duas últimas décadas tem sido o objetivo de desenvolver dispositivos protéticos para indivíduos paralisados e amputados que podem ser controlados usando sinais neurais. Talvez o mais próximo de ser

Figura 5.18. Ilustração de um implante coclear. Adaptado de [7].

comercializado estão os braços protéticos que podem ser controlados por sinais de nervos intactos (BCI semi-invasiva). Mais adiante no futuro estão braços e mãos protéticos que podem ser controlados diretamente usando neurônios corticais - os primeiros protótipos de tais BCIs estão correntemente sendo testados em macacos e humanos (BCIs invasivas).

Provavelmente o mais desafiador sejam próteses de pernas controlados por sinais do cérebro. Neste caso, o sistema de BCI/prótese precisa ser capaz de manter a estabilidade e permitir ao usuário manter o equilíbrio enquanto obedece a comandos do cérebro e provê retroalimentação por meio da estimulação de neurônios somatosensoriais apropriadamente. Uma abordagem baseada em BCIs hierárquicas baseado em uma mistura de autonomia e controle do usuário pode prover a forma mais flexível de controlar próteses de membros inferiores.

Similarmente, a estimulação profunda do cérebro (DBS) tem sido usada não apenas para tratar os sintomas do mal de Parkinson mas também para aliviar a dor crônica e a depressão. Finalmente, BCIs que podem gravar memórias e estimular centros de memória apropriados do cérebro poderiam ajudar a combater o prejuízo da memória de doenças tais como a doença de Alzheimer, embora o desenvolvimento de tais BCIs exigirá uma compreensão mais profunda de como as memórias são criadas e armazenadas no cérebro do que o que se conhece hoje.

5.8.2.2. Reabilitação

Uma outra aplicação potencialmente significativa de BCIs é em reabilitar pacientes em recuperação de derrame, cirurgia ou outra condição neurológica. A BCI seria parte de um

sistema de retroalimentação fechado que converte sinais do cérebro em estímulo em uma tela de computador ou em movimentos de um dispositivo de reabilitação. Tal sistema de retroalimentação neural pode habilitar pacientes a aprender a gerar o tipo apropriado de atividade neural para acelerar sua reabilitação.

Uma grande motivação para o desenvolvimento de BCIs baseadas em EEG não invasivas tem sido a restauração da comunicação para pacientes paralisados mas conscientes sofrendo de doenças motoras progressivas tal como a esclerose lateral amiotrófica (ALS, também conhecida como mal de Lou Gehrig). Nos casos em que os pacientes são incapazes mesmo de piscar ou sugar um canudo para indicar uma resposta “sim” ou “não”, uma BCI torna-se o único modo possível de comunicação.

Uma abordagem para restaurar a comunicação é construir uma BCI para controlar um cursor em um sistema de menu, permitindo ao paciente selecionar uma opção de um conjunto de escolhas. Um sistema de menu permite a composição de sentenças longas ou sequências de comandos. O cursor em tal sistema pode ser controlado por exemplo, via controle voluntário dos potenciais oscilatórios ou potenciais corticais lentos (não invasivos) ou também por métodos invasivos [57].

Alternativamente, um método evocado por estímulo tal como o soletrador BCI P300 pode ser usado para selecionar letras para soletrar palavras. Tanto o soletrador quanto abordagens baseadas em cursor podem ser bem lentas e tediosas para o paciente. Uma BCI mais natural para comunicação implica o aproveitamento dos centros da fala do cérebro. Alguns resultados preliminares em decodificar fonemas a partir da atividade neural gravada da região da fala (área de Broca) do córtex cerebral foram publicados [6], mas um entendimento mais aprofundado do processamento da fala no cérebro é necessário antes de uma BCI ser desenvolvida para traduzir pensamentos linguísticos.

Pacientes paralisados são algumas vezes capazes de controlar uma *cadeira de rodas* usando partes de seu corpo que ainda estão sob controle voluntário. Outros podem ser capazes de usar a fala para emitir comandos a uma cadeira de rodas semi-autônoma. Pode-se controlar uma cadeira de rodas diretamente usando sinais do cérebro?

A abordagem mais simples é usar uma BCI para selecionar comandos de alto nível (e.g., ir para a cozinha, ir para o quarto, etc.) e dotar a cadeira de rodas com conhecimento e autonomia suficientes para ser capaz de executar estes comandos de uma maneira autônoma. Os comandos de alto nível podem ser selecionados usando uma BCI síncrona, tal como a BCI baseada no P300. Esta abordagem pode ser flexível e adaptativa às necessidades do usuário usando uma BCI hierárquica.

Uma abordagem diferente [24] baseia-se no conceito de *controle compartilhado* (Figura 5.19). Nesta abordagem, o usuário continuamente gera comandos para o robô que são então probabilisticamente combinados com comportamentos pré-programados. A cadeira de rodas tem sensores como o *scanner* na faixa do laser. Se a meta do usuário é se mover suavemente no ambiente, a informação dos sensores da cadeira de rodas pode ser usada para construir um “filtro contextual” na forma de uma distribuição de probabilidade $P_{Env}(C)$ sobre um conjunto de possíveis comandos de direção mental, e.g., $C = [esquerda, direita, frente]$. O sistema BCI baseada em EEG estima as probabilidades $P_{EEG}(C)$ para diferentes comandos mentais a partir dos sinais do cérebro do usuário.

Figura 5.19. Controle BCI de uma cadeira de rodas inteligente [76].

Ainda que os resultados preliminares sejam animadores, uma cadeira de rodas controlada por BCI prática para o uso do dia-a-dia continua uma promessa devido à falta de um sistema de gravação (EEG ou outra modalidade) confiável, fácil de usar e portátil, assim como a falta de cadeiras de rodas robóticas robustas, e semi-autônomas, que podem funcionar seguramente em ambientes humanos.

5.8.3. Outras aplicações

5.8.3.1. Navegação na web e em mundos virtuais

Tem havido um crescimento no número de aplicações não médicas da tecnologia BCI. Muitas destas aplicações foram impulsionadas por fatores comerciais, tais como o potencial para uma nova interface para jogos e entretenimento. A maioria destas aplicações ainda está em sua infância e sendo investigada em laboratórios de pesquisa, embora algumas tenham sido aplicadas a problemas do mundo real, tais como triagem de grandes quantidades de imagens e detecção de mentiras.

Um exemplo de uma interface web-browser controlada por BCI é Nessi (*Neural Signal Surfing Interface* [2]), que permite ao usuário selecionar qualquer *link* em uma página web e acessar serviços baseados em web. Nessi é um software independente de plataforma e código aberto que pode ser usado com diferentes tipos de BCI. Uma demonstração usa uma BCI de duas classes baseada em potenciais corticais lentos (SCP). Quadros vermelhos ou verdes são colocados em volta de *links* em uma página web: quadros vermelhos eram selecionados produzindo deslocamentos SCP negativos e quadros verdes eram selecionados por deslocamentos SCP positivos.

Retroalimentação era provida na forma de um cursor que era movido para cima para uma meta vermelha ou para baixo para uma meta verde usando a BCI baseada em

SCP. O usuário tinha apenas de observar a cor do quadro do *link* desejado para saber que tipo de resposta do cérebro produzir, podendo portanto sucessivamente o conjunto de itens selecionáveis via decisões binárias até que o *link* desejado fosse selecionado.

5.8.3.2. Avatares robóticos

Telepresença controlada pelo cérebro, ou a idéia de controlar um avatar robótico remoto diretamente com a mente humana, tem sido objeto de filmes de Hollywood, como *Avatar* e *Surrogates*, mas os avanços na robótica e tecnologia BCI estão trazendo a idéia para mais próximo da realidade. Já se discutiu os esforços de investigação em curso para construir BCIs que podem controlar cadeiras de rodas robóticas. Uma linha paralela de pesquisa tem como alvo o desenvolvimento de robôs de assistência e avatares que podem ser controlados remotamente por meio de sinais cerebrais. Além de telepresença, esses robôs podem ajudar as pessoas paralisadas ou com deficiência na realização de várias tarefas na vida do dia-a-dia, tais como a obtenção de um copo de água da cozinha ou a busca de um frasco no armário de remédios [57].

Uma abordagem para avatares robóticos tem se concentrado em sistemas BCI baseados em EEG para controle de robôs humanóides [1]. Em uma das primeiras manifestações de um “avatar” controlado pelo cérebro, uma BCI baseada em P300 foi utilizada para comandar um robô humanóide para ir para locais desejados e buscar objetos desejados [1]. O usuário tinha uma visão do olho do robô do ambiente, o que proporcionou uma experiência imersiva. O robô tinha a capacidade de mover-se de forma autônoma e pegar/liberar objetos. O robô também possuía alguns recursos de visão computacional, tais como ser capaz de segmentar objetos que ele viu em uma mesa e usar visão para navegar para um destino.

Esforços mais recentes têm focado em fazer a BCI mais adaptável às necessidades do usuário usando BCIs hierárquicas para aprender novos comandos para o robô. Futuros avatares robóticos controlados pelo cérebro podem permitir mais controle refinado, talvez com base em gravações invasivas, bem como retroalimentação mais rica do robô, incluindo retroalimentação auditiva e tátil e, finalmente estimulação direta de áreas sensoriais do cérebro com base nas leituras dos sensores do robô.

5.8.3.3. Busca de imagens em alta quantidade

O cérebro humano é extremamente habilidoso no processamento visual em relação aos atuais sistemas de visão computacional. Uma aplicação interessante de BCIs é aproveitar as capacidades de processamento de imagem do cérebro para uma rápida busca visual de grandes conjuntos de dados de imagem. A idéia, explorada por Sajda e colegas [62], é usar a análise de um único teste para rapidamente detectar assinaturas neurais correlacionadas com o reconhecimento visual. Suponha que o objetivo seja classificar imagens (por exemplo, imagens de satélite) de tal modo que as imagens com maior probabilidade de conter objetos de interesse (por exemplo, tanques) são colocadas no início da sequência de imagens para um exame mais aprofundado.

Gerson e colegas [28] desenvolveram uma BCI baseada em EEG em tempo real para triagem de tais imagens usando o paradigma da apresentação visual serial rápida (RSVP). A sua técnica, denominada *visão computacional corticalmente acoplada* (CCCV) baseia-se no paradigma *oddball* para provocar um P300: uma imagem alvo que ocorre em uma sequência de imagens distratoras não-alvo vai causar uma resposta P300.

Em cada teste, é apresentada ao sujeito, uma sequência contínua de 100 imagens, cada imagem durando 100 ms. A sequência continha 2 imagens alvo com 1 ou mais pessoas em uma cena natural. A aparição repentina de uma imagem alvo na sequência tipicamente provocou um P300, que foi detectado por um classificador. A saída do classificador foi usada para repriorizar a sequência de imagens, colocando as imagens alvo detectadas na frente da pilha de imagens.

5.8.3.4. Detecção de mentiras e aplicações na lei

Uma aplicação de BCIs que provocou um interesse (e polêmica) considerável em comunidades de justiça penal e lei é a detecção de mentira ou detecção de posse de conhecimento culpado. A técnica tradicional é o polígrafo, que mede as reações corporais de sujeitos, tais como alterações na pressão sanguínea, a condutividade da pele, e frequência cardíaca enquanto ele ou ela responde a uma série de perguntas durante um interrogatório. A premissa é que respostas enganosas irão produzir respostas fisiológicas diferentes daquelas associadas com respostas verdadeiras [57].

Embora a poligrafia seja usada por muitas agências de aplicação da lei, é geralmente considerada pouco confiável pela maioria dos cientistas porque mede a ansiedade, em vez do engano, e seus níveis de acurácia são considerados pouco melhores do que o acaso.

Para superar as deficiências do polígrafo, os pesquisadores de BCI têm explorado o uso de respostas do cérebro como um modo para detectar se um sujeito anteriormente encontrou ou possuiu conhecimento sobre uma pessoa específica, lugar ou objeto. O desafio é projetar uma BCI para a *detecção de memória* que poderia ser usada diretamente para interrogar os cérebros de suspeitos e testemunhas. O objetivo é encontrar prova neural, se existir, de reconhecimento de uma pessoa, lugar ou objeto vinculado a uma cena de crime.

Um dos primeiros exemplos de uma BCI “detectora de mentiras,” baseado no potencial relacionado a evento (ERP) P300 foi investigado por Farwell e Donchin [13]. Neste paradigma, o sujeito é solicitado a discriminar entre alvos predeterminados e estímulos irrelevantes. Incorporado entre os estímulos irrelevantes, está um conjunto de itens de diagnóstico chamado de “sondas”, que são indistinguíveis a partir de itens irrelevantes se o sujeito não possuir conhecimento culpado. Para os indivíduos que possuem conhecimento culpado, as sondas são percebidas de forma diferente dos itens irrelevantes e são susceptíveis de provocar um P300, o qual pode ser detectado por uma BCI.

5.8.3.5. Monitoramento de alerta

Uma aplicação potencialmente importante de BCIs é a monitoração do estado de alerta dos seres humanos durante a execução de tarefas críticas, mas potencialmente monótonas, como direção de veículos ou vigilância. Muitos acidentes catastróficos são causados anualmente por condutores que estão cansados, sonolentos, ou até mesmo dormindo ao volante. Esses acidentes podem ser evitados monitorando os sinais do cérebro para qualquer transição de um estado de alerta e de vigília para um estado que indica falta de atenção. Embora os estados de sono ou sonolência possam ser detectados pelo monitoramento do fechamento das pálpebras, tal detecção pode ocorrer tarde demais para evitar um acidente [57].

Detecção baseada em cérebro de alerta diminuída também tem aplicações na educação e aprendizagem em que tal detecção poderia ser usada para medir o grau em que um estudante está envolvido durante uma aula.

Os pesquisadores têm procurado achar correlatos de uma diminuição da atenção e alerta em sinais cerebrais, especialmente EEG. Conhece-se, por algum tempo, que um aumento na potência em certas bandas de frequência (tais como alfa, 8-13 Hz) em correlatos de EEG com uma diminuição na concentração, conforme medido por taxas de erro mais elevadas na detecção de tarefas [57].

5.8.3.6. Educação e aprendizado

Já foi discutido como técnicas de BCIs não invasivas são úteis para medir o nível de alerta e a carga cognitiva durante a realização de uma tarefa. Ideias similares podem ser aplicadas para avaliar o grau de envolvimento, atenção e carga cognitiva de um estudante ouvindo uma palestra ou fazendo um exercício. Por exemplo, a empresa NeuroSky desenvolveu um aplicativo BCI que tenta medir o nível de atenção do usuário durante um exercício de matemática. A BCI é baseada no *headset* da NeuroSky, o MindWave, que mede EEG a partir de um eletrodo seco frontal [57].

Em um estudo recente, Szafir e Mutlu [65] também usam um eletrodo frontal na posição Fp1 no sistema 10-20 para monitorar o nível de atenção de um aluno enquanto o aluno ouvia um conto popular japonês sendo recitado por um robô humanóide. Durante os 10 minutos da história sendo contada pelo robô, sempre que o sistema detectava (a partir do sinal de EEG) que o nível de atenção do aluno caía, o robô levantava a sua voz ou executava movimentos do braço para recuperar a atenção do aluno. Os pesquisadores descobriram que os alunos que ouviam a história do robô, cujo comportamento era contingente à BCI de detecção de atenção, eram muito melhores nas respostas a perguntas sobre o conto popular, respondendo uma média de 9 das 14 perguntas corretamente, em comparação com os alunos para quem o robô não exibiu comportamento contingente à atenção.

Os primeiros resultados discutidos acima, se verificados em estudos aprofundados subsequentes, indicam que as BCIs poderiam potencialmente fornecer retroalimentação valiosa para educadores, bem como para alunos, permitindo medidas adequadas a serem tomadas para estratégias educacionais, paradigmas de interação e lições de acordo com o

estado de atenção e necessidades correntes de cada estudante. Ser capaz de detectar um envolvimento do aluno ou um nível de atenção pode ser especialmente útil para esforços educacionais *on-line* (tais como os buscados por Khan Academy, Coursera, EdX, e Udacity) em que não há nenhum professor humano para medir o envolvimento do aluno quando o aluno está assistindo material apresentado em um vídeo *on-line*.

Os alunos podem ainda usar a BCI como um dispositivo de apoio para melhorar a sua concentração e performance. A BCI pode também ser útil para ajudar os alunos com transtornos de déficit de atenção ao apanhar lapsos de atenção e redirecionar o foco. Assim que avanços na neurociência fornecem uma compreensão mais aprofundada dos mecanismos envolvidos na aprendizagem e compreensão, pode-se esperar novas BCIs a ser desenvolvidas que aproveitem esses avanços e acelerem a aprendizagem, adaptando-se ao estilo e velocidade de aprendizado de cada estudante. Os professores e os pais poderiam potencialmente determinar o grau em que um aluno aprendeu um conceito particular diretamente a partir de mudanças em sinais do cérebro, fornecendo uma alternativa aos testes padronizados para medir a competência e a aprendizagem [57].

5.8.3.7. Segurança, identificação e autenticação

As BCIs estão começando a ser aplicadas a problemas de segurança tal como a identificação biométrica para a recuperação de informações de bancos de dados e autenticação para controle de acesso (e.g., para segurança de aeroporto, login de conta, ou banco eletrônico). Como um exemplo, a atividade do ritmo alfa distintivo de um sinal de EEG individual tem sido proposta como uma assinatura biométrica para identificação. Em um estudo [55], sujeitos foram solicitados a relaxar e fechar os olhos enquanto o EEG era registrado a partir de eletrodos O2 e Cz no sistema 10-20. O sinal bipolar obtido da diferença entre O2 e Cz foi filtrado por passagem de banda na banda de frequência 7.5-12.5 Hz (banda alfa) usando FFT e FFT inversa [57].

Enquanto a identificação envolve reconhecer uma pessoa de um grande grupo de indivíduos, o problema de autenticação envolve a verificação se a pessoa que alega uma identidade é realmente essa pessoa ou um impostor. Pesquisadores estão começando a explorar a utilização de BCIs baseadas em EEG para autenticação. Em um estudo realizado por Marcel e Millan [45], os indivíduos foram convidados a executar uma de 3 tarefas mentais (imaginação de movimentos da mão esquerda ou direita e geração de palavra). Os sinais de EEG foram espacialmente filtrados usando um filtro laplaciano e características espectrais de potência na banda 8-32 Hz foram extraídas usando a FFT. Essas características foram utilizadas para construir um modelo probabilístico dos dados [57].

5.8.3.8. Amplificação física com exoesqueletos

Muitos vilões das histórias em quadrinhos querem amplificar o poder do corpo humano para alcançar força sobre-humana (por exemplo, Dr. Octopus em *Homem-Aranha*). Exoesqueletos motorizados oferecem os meios para atingir amplificação do corpo humano além do que a evolução nos dotou. Enquanto os pesquisadores têm explorado os mecanismos de controle para exoesqueletos com base em sinais de movimentos auto-gerados ou

sinais musculares (EMG), os pesquisadores de BCI começaram recentemente a explorar a utilização de sinais cerebrais para controlar diretamente um exoesqueleto (Figura 5.20).



Figura 5.20. Exoesqueleto das forças armadas americanas [72].

Como exemplo, o projeto europeu Mindwalker procura usar sinais de EEG gravados a partir de eletrodos secos de design personalizado e redes neurais recorrentes para controlar um exoesqueleto robótico anexado às pernas do sujeito. O duplo objetivo do projeto é habilitar pessoas com lesões na medula espinhal a alcançar mobilidade e a ajudar na recuperação de astronautas depois de uma missão prolongada no espaço.

Uma série de empresas como a Cyberdyne, EKSO Bionics, e Raytheon têm desenvolvido exoesqueletos motorizados que amplificam a força dos usuários, permitindo-lhes levantar e transportar até 200 libras de peso (90.7 Kg) com pouco ou nenhum esforço. No futuro, exoesqueletos de corpo inteiro poderiam ser utilizados por equipes de salvamento, bombeiros, e soldados para se mover mais rápido, saltar mais alto, carregar cargas mais pesadas, e executar outras proezas físicas que não podem ser executadas por um organismo humano normal. Estes exoesqueletos poderiam potencialmente ser controlados por sinais cerebrais, e a retroalimentação do exoesqueleto poderia ser utilizada para estimular diretamente centros somatossensoriais apropriados no cérebro para permitir controle preciso, na medida em que o exoesqueleto possa ficar incorporado como parte do mapa do corpo no cérebro do usuário [57].

5.8.3.9. Amplificação mnemônica e cognitiva

O enredo do filme *Johnny Mnemonic* gira em torno do protagonista agindo como um mensageiro com dados secretos implantados em seu cérebro. Outros roteiros de ficção

científica têm contado com máquinas que podem injetar seletivamente ou apagar memórias. Essas habilidades ainda têm de ser demonstradas em uma BCI, mas os pesquisadores começaram recentemente a explorar a possibilidade de restauração da memória e ampliação das funções cognitivas via gravação e estimulação neurais.

Embora ainda a serem testados em seres humanos, os implantes de memória oferecem um raio de esperança para aqueles que sofrem de doença de Alzheimer, amnésia, e outros distúrbios de memória devastadores. Além disso, a capacidade de armazenar e amplificar certas memórias abre a porta para novas formas de valorização da memória e amplificação cognitiva para indivíduos sãos. Por exemplo, memórias podem ser armazenadas *off-line* (por exemplo, na “nuvem”) e recuperadas quando necessárias, por meio de implantes sem fio. Embora seres humanos hoje rotineiramente usam a internet, livros, smartphones, computadores e outros dispositivos como armazenamentos de memória externa, os implantes de memória fariam acesso a tais informações essencialmente de forma suave, permitindo armazenamento e recuperação apenas por meio do pensamento [57].

5.8.3.10. Jogos e entretenimento

Muitos paradigmas BCI tradicionais (por exemplo, controle de cursor) têm um gostinho de jogo. Para aplicações médicas, como a seleção de menu ou reabilitação com base em retroalimentação neural, usar um paradigma de interação tipo-jogo ajuda na manutenção do interesse do paciente. Esses aplicativos não foram projetados com fins de entretenimento em mente, mas o jogo para os indivíduos sãos não deixa de ser uma das áreas de BCI de aplicações não-médicas que cresce com mais rapidez. Uma das razões para este crescimento é o enorme mercado que existe atualmente para *video-games*, diminuindo o mercado para aplicações médicas de BCIs [57].

Uma segunda razão é que ao contrário de aplicações médicas, como cadeiras de rodas ou próteses controladas por BCI, o desempenho deficiente de uma BCI em um jogo pode incomodar um usuário mas normalmente não causa danos corporais ou prejuízo para o usuário ou indivíduos nas proximidades, diminuindo assim as preocupações de responsabilidade. Finalmente, as BCIs podem ser utilizadas em jogos como uma interface que melhora outras interfaces mais tradicionais, como *joysticks*, *gamepads*, sistemas de reconhecimento de gestos, e assim por diante. Assim, ao contrário de aplicações médicas das BCIs, tais como sistemas de comunicação para pacientes *locked-in*, as BCIs para jogos podem depender ativamente de uma mistura de sinais cerebrais (por exemplo, sinais de EEG), sinais musculares (EMG), e movimentos de mão/corpo para alcançar um novo modo de interação humano-computador [57].

Em um dos primeiros estudos a explorar esta direção, Cheung e colegas [10] demonstraram que os sujeitos poderiam controlar o movimento bidimensional de um cursor usando um *joystick* simultaneamente com imagética motora de mão em uma BCI baseada em EEG. Sujeitos aprenderam a usar imagética para controlar o movimento de cima para baixo do cursor e simultaneamente utilizar o *joystick* para o controle do movimento esquerda-direita do cursor. Estes resultados sugerem que pode ser possível utilizar BCIs para aumentar a capacidade motora normal em indivíduos sãos.

Um grande número de jogos controlados pelo cérebro foi introduzido ao longo da

última década. Brainball foi um jogo BCI em que os usuários aprenderam a controlar o seu nível de relaxamento, controlando seu ritmo alfa. MindGame é um jogo mais recente baseado no P300, que envolve o movimento de um personagem em um jogo tridimensional de tabuleiro. Outros aplicativos de jogos têm contado com SSVEP e imagética motora (por exemplo, o BCI-PacMac), bem como navegação virtual baseada em EEG [63].

Uma demonstração interessante de controle em tempo real de um dispositivo de jogos físico envolveu uma máquina de pinball controlada por BCI, em que o *paddle* era controlado por uma BCI de 2 classes baseada em imagética (por exemplo, imagética motora de mão esquerda e direita). Os parâmetros BCI estavam sintonizados individualmente para cada usuário. Os pesquisadores relataram que o jogo era percebido como altamente imersivo e motivador.

Houve também uma versão controlada por BCI do popular videogame Tetris [5]. O jogo de BCI baseada em EEG baseia-se em um conjunto “natural” de controles: o jogador usa imagética motora de mão esquerda ou direita para mover uma peça Tetris caindo para a esquerda ou a direita respectivamente, e rotação mental para girar a peça no sentido horário e imagética motora do pé para soltar a peça. Um classificador de 4 classes (3 comandos de imagética motora e rotação mental) foi treinado em uma fase de calibração *off-line* e, em seguida, aplicado *on-line* durante a fase do jogo para conseguir o controle de uma peça caindo [57].

Vários sistemas comerciais apareceram recentemente no mercado que tentam medir sinais tipo EEG do couro cabeludo. Estes sistemas normalmente usam um pequeno número de eletrodos secos (em contraste aos eletrodos de EEG tradicionais “molhados” que exigem gel para fazer contato com o couro cabeludo). As medições feitas por estes eletrodos secos são usadas para controlar objetos em uma tela de computador ou objetos reais tal como uma bola. Exemplos incluem sistemas fabricados pela Emotiv (*headset* EPOC) e Neurosky (*headset* Mindwave), e brinquedos como o Mindflex da Mattel.

Estes sistemas novos são muito mais baratos do que sistemas de EEG baseados em gel tradicionais usados na pesquisa e clínicas e eles são mais fáceis de usar e operar. No entanto, um problema deste novo sistema é que não há nenhuma garantia de que estão capturando os verdadeiros sinais de EEG. Em ambientes não controlados, tais sistemas podem capturar uma mistura de atividade EEG e EMG causada por ativação muscular da face e do pescoço, movimentos dos olhos, mudanças na resistência da pele, ou em alguns casos, mesmo ruído elétrico. Por outro lado, como mencionado acima, o uso de um EEG/EMG híbrido ou outro tipo de sinal voluntariamente gerado pode ser bom para aplicações de jogos, se constituir em um novo e potencial modo de entretenimento de controle em um jogo.

5.8.4. Pesquisa desenvolvida no laboratório BioCom da USP

No laboratório de Computação Bio-inspirada (BioCom) da Universidade de São Paulo (USP), campus de São Carlos, existe um grupo de neurodinâmica computacional, no qual se inserem vários projetos relacionados às BCI. Alunos pesquisam temas, que vão desde reconhecimento de faces até BCI de imagética motora, usando os conjuntos-K de Freeman. Outros alunos pesquisam a interpretação de sinais de EEG utilizando diversas técnicas, desde algoritmos tradicionais de aprendizado de máquina até redes complexas,

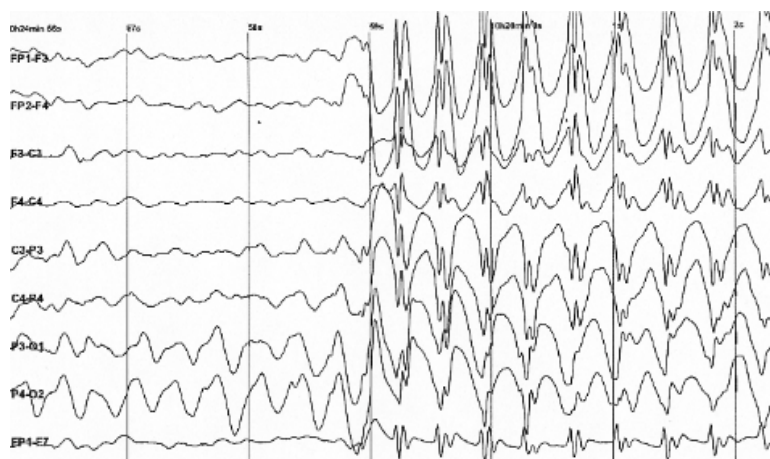


Figura 5.21. Vários canais de EEG com *spikes* de 3 Hz num sujeito com epilepsia. O EEG mostra atividade anormal a partir da metade do quadro [71].

para prevenção de doenças, como a epilepsia. Este grupo conta com o apoio de pesquisadores da Faculdade de Medicina da USP de Ribeirão Preto, que fornecem bases de dados de EEG de epilepsia. Além dessas, bases de dados disponíveis publicamente, como as das *BCI competitions*⁷, são utilizadas. A seguir, alguns dos trabalhos do grupo de neurodinâmica computacional desenvolvidos no BioCom:

Redes complexas para análise de sinais de epilepsia: Como a epilepsia é uma condição com alta incidência é natural tentar utilizar redes complexas para detectar sinais da epilepsia. Pretende-se converter sinais de EEG (Figura 5.21) em redes complexas e utilizar as medidas de caracterização das redes para detectar crises.

BCI baseadas em sinais eletrofisiológicos para o tratamento de epilepsias: Uma das possíveis aplicações de BCI é voltada ao monitoramento das atividades cerebrais de pacientes com epilepsias, visando a detecção de atividades convulsivas. A forma mais comum para analisar a atividade cerebral é por meio de sinais de EEG. Utilização dos conjuntos-K de Freeman.

Preenchimento automático de laudos médicos por meio do processamento de EEG utilizando aprendizado de máquina: O objetivo desse trabalho é desenvolver um método computacional para o preenchimento automático de laudos médicos de forma padronizada a partir do processamento de exames de EEG utilizando redes neurais artificiais [50].

Aplicações de BCI em robótica: Durante muito tempo apenas o uso de técnicas cirúrgicas foram utilizadas para controle de próteses e braços robóticos. Novos estudos mostram que a EEG pode ser utilizada para esse propósito. Desta forma, propõe-se a investigação de formas de aprendizado que, com as informações obtidas pela EEG, possam permitir o controle de algum dispositivo robótico.

Desenvolvimento de um modelo computacional para apoio ao diagnóstico de transtornos psicóticos: O diagnóstico não envolve apenas decidir o que é verdade sobre um paciente, mas também os dados que são necessários para determinar o que é verdadeiro. O objetivo do projeto é por meio da identificação e exploração de diferenças nas

⁷<http://www.bbc.de/competition/>

características estruturais e funcionais do cérebro desenvolver um modelo computacional capaz de apresentar apoio à decisão de diagnóstico para esquizofrenia, transtorno bipolar e transtorno depressivo.

Conjuntos-K de redes neurais e sua aplicação na classificação de imagética motora: A imagética motora deixa um rastro no sinal da EEG que torna possível a identificação e classificação dos diferentes movimentos motores [52]. Um simulador para os conjuntos-K foi construído em Java e está disponível publicamente [53].

Reconhecimento de faces utilizando um modelo conexionista baseado em populações de neurônios: Reconhecimento de faces consiste em, a partir de uma imagem, identificar ou verificar um ou mais indivíduos através de um banco de dados de faces utilizando os conjuntos-K [9].

5.9. Conclusão

Este curso apresentou os conceitos básicos de Interfaces Cérebro-Computador (BCI), desde a neurociência básica, focando em populações de neurônios, até a plasticidade sináptica. Foram discutidos conceitos relacionados à neurodinâmica, apresentando os conjuntos-K de Freeman, modelo de rede neural biologicamente plausível [59], que se baseia na população de neurônios e não no neurônio individual. Apresentou-se elementos das BCI, como formas de registro de sinais do cérebro, dando ênfase à eletroencefalografia, técnica não-invasiva muito usada em pesquisa com sujeitos normais. Discutiu-se também conceitos de processamento de sinais e de aprendizado de máquina. Por fim, apresentou-se diversas aplicações de BCI, em áreas médicas e não médicas.

Agradecimento

O autor agradece aos revisores anônimos da versão preliminar deste texto, que com seus comentários construtivos, possibilitaram uma versão final muito mais aprimorada. Quaisquer erros remanescentes são de inteira responsabilidade do autor.

Referências

- [1] Bell, C. J., Shenoy, P., Chalodhorn, R., and Rao, R. P. N. (2008) “Control of a humanoid robot by a non-invasive brain-computer interface in humans.” *J Neural Eng.* 5(2):214–20.
- [2] Bensch, M., Karim, A. A., Mellinger, J., Hinterberger, T., Tangermann, M., Bogdan, M., Rosenstiel, W., and Birbaumer, N. (2007) “Nessi: An EEG-Controlled Web Browser for Severely Paralyzed Patients.” *Comput Intell Neurosci.* 2007(Art ID 71863), 5 pp.
- [3] Berger, T. W., Hampson, R. E., Song, D., Goonawardena, A., Marmarelis, V. Z., and Deadwyler, S. A. (2011) “A cortical neural prosthesis for restoring and enhancing memory.” *J. Neural Eng.* 8 046017 (11pp).
- [4] Bernstein, J. (1902) “Untersuchungen zur Thermodynamik der bioelektrischen Ströme”. *J. Pflügers Arch. Ges. Physiol.* 92:521–562.
- [5] Blankertz, B., Tangermann, M., Vidaurre, C., Fazli, S., Sannelli, C., Haufe, S., Maeder, C., Ramsey, L., Sturm, I., Curio, G., and Müller, K.-R. (2010) “The Berlin brain-computer interface: non-medical uses of BCI technology.” *Frontiers on Neurosc.* Dec, 4(Art 198).

- [6] Blakely, T., Miller, K. J., Rao, R. P. N., Holmes, M. D., and Ojemann J. G. (2008) “Localization and classification of phonemes using high spatial resolution electrocorticography (ECoG) grids.” *Conf Proc IEEE Eng Med Biol Soc*, 2008:4964–67.
- [7] Blausen gallery 2014. Illustration of a cochlear implant: Blausen.com staff. Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762. - Own work
- [8] Brunner P. and Schalk, G. (2009) “Brain-Computer Interaction,” In: D. D. Schmorrow et al. (Eds.): *Augmented Cognition*, HCII 2009, LNAI 5638:719–723, 2009. Springer-Verlag Berlin Heidelberg.
- [9] Carlos Junior, L. F. M. and Rosa, J. L. G. (2014), “Face Recognition through a Chaotic Neural Network Model,” *Proceedings of the IJCNN 2014 Intl Joint Conf on Neural Networks*, IEEE World Cong on Comput Intell (IEEE WCCI), Beijing, China, July 6-11, 2014:859-863.
- [10] Cheung, W., Sarma, D., Scherer, R., and Rao, R. P. N. (2012) “Simultaneous brain-computer interfacing and motor control: expanding the reach of non-invasive BCIs.” *Conf Proc IEEE Eng Med Biol Soc*. 2012:6715–8.
- [11] Cole, K. S. and Curtis, H. J. (1939) “Electric Impedance of the Squid Giant Axon during Activity”, *Gen. Physiol.* 22:649–670.
- [12] Cooley, J. W. and Tukey, J. W. (1965) “An algorithm for the machine calculation of complex Fourier series.” *Math Comput* 19:297–301.
- [13] Farwell, L. A. and Donchin, E. (1991) “The truth will out: interrogative polygraphy (“Lie Detection”) with event-related brain potentials”, *Psychophys*, 28(5):531–547.
- [14] Fatourech, M., Bashashati, A., Ward, R. K., and Birch, G. E. (2007) “EMG and EOG artifacts in brain computer interface systems: A survey.” *Clin Neurophys* 118(3):480–494.
- [15] Nicolas Ferrando, Lois Lammerhuber (credit) (2009) Brain Computer Interface / g-tech medical engineering, <https://www.flickr.com/photos/arselectronica/4306147303>, Taken on May 27, 2009.
- [16] Freeman, W. J. (1975) *Mass action in the nervous system - Examination of the Neurophysiological Basis of Adaptive Behavior through the EEG*, Academic Press, New York San Francisco London.
- [17] Freeman, W. J. (1991) “The physiology of perception,” *Scientific American*, 264:78.
- [18] Freeman, W. J. (1999) *How Brains Make Up Their Minds*, Weidenfeld & Nicolson, London.
- [19] Freeman, W. J. (2000) *Mesoscopic Brain Dynamics*, Springer-Verlag London Limited.
- [20] Freeman, W. J. and Erwin, H. (2008) “Freeman K-set,” *Scholarpedia* 3(2):3238. http://www.scholarpedia.org/article/Freeman_K-set.
- [21] Freeman, W. J. and Kozma, R. (2011) “Neuropercolation + Neurodynamics: Dynamical Memory Neural Networks in Biological Systems and Computer Embodiments,” Tutorial 6, *IEEE IJCNN 2011 - Intl. Joint Conf. on Neural Networks*, San Jose, CA, July 31.

- [22] Freeman, W. J. and Quian-Quiroga, R. (2013) *Imaging Brain Function With EEG - Advanced Temporal and Spatial Analysis of Electroencephalographic Signals*. Springer.
- [23] Freeman, W. J. and Quian-Quiroga, R. (2013) *Imaging Brain Function With EEG - Advanced Temporal and Spatial Analysis of Electroencephalographic Signals*. Springer. Chapter 2 - Frequency Analysis, available from https://www.researchgate.net/279353557_fig1_Figure-1-1-a-A-sinusoidal-signal-in-the-time-upper-plot-and-frequency-bottom-plot.
- [24] Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J., and Millán, J. del R. (2008) “A brain-actuated wheelchair: Asynchronous and non-invasive Brain-computer interfaces for continuous control of robots.” *Clin Neurophys*, 119(9):2159–2169.
- [25] Hugo Gamboa (author) (2005) One second of EEG signal. http://commons.wikimedia.org/wiki/File:Eeg_raw.svg. Dez 2005.
- [26] Gao, S., Wang, Y., Gao, X., and Hong, B. (2014) “Visual and Auditory Brain-Computer Interfaces,” *IEEE Trans on Biom Eng*, 61(5):1436–1447.
- [27] Georgopoulos, A. P., Kettner, R. E., and Schwartz, A. B. (1988) “Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population.” *J of Neurosc* 8(8):2928–37.
- [28] Gerson, A. D., Parra, L. C., and Sajda, P. (2006) “Cortically Coupled Computer Vision for Rapid Image Search,” *IEEE Tr: on Neural Systems and Rehab. Eng.*, 14(2):174–179.
- [29] Häuser, M. (2000) “The Hodgkin-Huxley theory of the action potential.” *Nature - neuroscience supplement*, 3, November:1165.
- [30] Haykin, S. (2009) *Neural Networks and Learning Machines*. 3rd. Ed. Pearson.
- [31] Hodgkin, A. L. and Huxley, A. F. (1939) “Action potentials recorded from inside a nerve fibre.” *Nature* 144:710–11.
- [32] Hodgkin, A. L. and Huxley, A. F. (1952) “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *J. Physiol.* 117:500–544.
- [33] Hodgkin, A. L. and Katz, B. (1949) “The effect of sodium ions on the electrical activity of the giant axon of the squid.” *J. Physiol.* (Lond.) 108:37–77.
- [34] Hopfield, J. J. (1982) “Neural networks and physical systems with emergent collective computational abilities.” *Proc National Academy of Sciences of the USA*, 79(8):2554–2558.
- [35] Kaneko, K. and Tsuda, I. (2001) *Complex Systems: Chaos and Beyond. A Constructive Approach with Applications in Life Sciences*, Springer, Berlin NY.
- [36] Katchalsky, A. K., Rowland, V., and Blumenthal, R. (1974) *Dynamic patterns of brain cell assemblies*, MIT Press.
- [37] Kozma, R. (2007) “Neuropercolation,” *Scholarpedia* 2(8):1360. <http://www.scholarpedia.org/article/Neuropercolation>.
- [38] Kozma, R., Aghazarian, H., Huntsberger, T., Tunstel, E., and Freeman, W. J. (2007) “Computational aspects of cognition and consciousness in intelligent devices,” *IEEE Comput Intell Mag*, Aug:53–64.

- [39] Kozma, R. and Freeman, W. J. (2001) “Chaotic Resonance - Methods and Applications for Robust Classification of Noisy and Variable Patterns,” *Int. J. Bifurcation & Chaos*, 11(6):1607–1629.
- [40] Kozma, R. and Freeman, W. J. (2003) “Basic Principles of the KIV Model and its application to the Navigation Problem,” *J of Integrative Neurosc*, 2(1):125–146.
- [41] Kozma, R. and Freeman, W. J. (2009) “The KIV Model of Intentional Dynamics and Decision Making,” *Neural Networks*, 22(3):277–285.
- [42] Kozma, R., Rosa, J. L. G., and Piazzentin, D. R. M. (2013), “Cognitive Clustering Algorithm for Efficient Cybersecurity Applications,” *Proc IJCNN 2013 Intl Joint Conf on Neural Networks*, Dallas, TX, Aug 4-9, 2013, 471–478.
- [43] Kuffler, S. W., Nicholls, J. G., and Martin, A. R. (1984) *From Neuron to Brain: A Cellular Approach to the Function of the Nervous System*, 2nd. Edition, Sinauer Associates Inc. Sunderland, MA.
- [44] Lopes da Silva, F. H. and Pijn, J. P. (2003) “EEG and MEG Analysis,” In: M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks*, 2nd. edition, A Bradford Book. The MIT Press, 387–391.
- [45] Marcel, S. and Millán, J. del R. (2007) “Person authentication using brainwaves (EEG) and maximum a posteriori model adaptation.” *IEEE Trans Pattern Anal Mach Intell.*; 29(4):743–52.
- [46] Matthews, F., Pearlmutter, B. A., Ward, T. E., Soraghan, C., and Markham, C. (2008) “Hemodynamics for Brain-Computer Interfaces,” *IEEE Signal Proc Mag*, 25(1):87–94.
- [47] Mavooria, J., Jackson, A., Diorio, C., and Fetz, E. (2005) “An autonomous implantable computer for neural recording and stimulation in unrestrained primates.” *J of Neurosc Methods* 148:71–77.
- [48] McCulloch, W. S. and Pitts, W. (1943) “A logical calculus of the ideas immanent in nervous activity.” *Bull of Math Biophysics*, 5:115-133.
- [49] Millán, J. del R. (2003) “Brain-Computer Interfaces,” In: M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks*, 2nd. ed, A Bradford Book. The MIT Press, 178–181.
- [50] Oliva, J. T. and Rosa, J. L. G. (2016), “Predictive Models for Differentiation between Normal and Abnormal EEG through Cross-Correlation and Machine Learning Techniques,” Springer LNCS SOTA (State-of-the-Art) Vol on *Machine Learning for Health Informatics*, 2016.
- [51] Oppenheim, A. V. and Schaffer, R. W. (1975). *Digital Signal Processing*. Prentice Hall.
- [52] Piazzentin, D. R. M. and Rosa, J. L. G. (2014), “Motor imagery classification for Brain-Computer Interfaces through a chaotic neural network,” *Proc IJCNN 2014 Intl Joint Conf on Neural Networks*, IEEE World Cong on Comput Intell (IEEE WCCI), Beijing, China, July 6-11, 2014:4103-4108.
- [53] Piazzentin, D. R. M. and Rosa, J. L. G. (2015), “A simulator for Freeman K-sets in Java,” *Proc IJCNN 2015 Intl Joint Conf on Neural Networks*, Killarney, Ireland, July 12-17, 2015:1–8.

- [54] Postelnicu, C. C., Talabã, D., and Toma, M. I. (2010) “Brain Computer Interfaces for Medical Applications”, *Bull of the Transylvania Univ of Brasov*, vol. 3 (52), Series I: Eng Sciences. 99–106.
- [55] Poulos, M., Rangoussi, M., Chrissicopoulos, V., and Evangelou, A. (1999) “Person identification based on parametric processing on the EEG.” In: *Proc. of the Sixth Intl. Conf. on Electronics, Curcuits and Systems (ICECS99)*, Pafos, Cyprus; 1:283–86.
- [56] Rabinovich, M. I., Huerta, R., Varona, P., and Afraimovich, V. S. (2008) “Transient cognitive dynamics, metastability, and decision making,” *PLoS comput biology*, vol. 4(5), e1000072.
- [57] Rao, R. P. N. (2013) *Brain-Computer Interfacing - An Introduction*, Cambridge Univ Press.
- [58] Rosa, J. L. G. (2011) *Fundamentos da Inteligência Artificial*, Editora LTC.
- [59] Rosa, J. L. G. (2013), “Biologically Plausible Artificial Neural Networks” (chapter 2) in Kenji Suzuki (ed.), *Artificial Neural Networks - Architectures and Applications*, ISBN 978-953-51-0935-8, Pub: InTech.
- [60] Rosa, J. L. G. and Piazzentin, D. R. M. (2016), “A New Cognitive Filtering Approach based on Freeman K3 Neural Network,” *Applied Intelligence 2016*, Springer. DOI 10.1007/s10489-016-0772-4. 20 pages. First online: 09 March 2016 <http://link.springer.com/article/10.1007%2Fs10489-016-0772-4>.
- [61] Saa, A. (2004) “O Modelo de Hodgkin-Huxley”. vigo.ime.unicamp.br/~fismat/2004-I-NC-aula7.pdf.
- [62] Sajda, P., Pohlmeier, E., Wang, J., Parra, L. C., Christoforou, C., Dmochowski, J., Hanna, B., Bahlmann, C., Singh, M. K., and Chang, S.-F. (2010) “In a Blink of an Eye and a Switch of a Transistor: Cortically Coupled Computer Vision.” *Proc of the IEEE*, 98(3): 462–478.
- [63] Scherer, R., Lee, F., Schlögl, A., Leeb, R., Bischof, H., and Pfurtscheller, G. (2008) “Towards self-paced brain-computer communication: Navigation through virtual worlds.” *IEEE Trans Biomed Eng*, 55(2):675–82.
- [64] Stam, C. J., Breakspear, M., van Cappellen van Walsum, A. M., and van Dijk, B. W. (2003) “Nonlinear synchronization in EEG and whole-head recordings of healthy subjects,” *Hum Brain Mapp*, 19:63–78.
- [65] Szafir, D. and Mutlu, B. (2012) “Pay attention! Designing adaptive agents that monitor and improve user engagement.” In: *Proc. ACM SIGCHI Conf. on Human Factors in Comp. Systems (CHI 2012)*. 11–20.
- [66] Vidal, J. J. (1973) “Toward direct brain-computer communication,” *Annu. Rev. Biophys. Bioeng.*, 2:157–180.
- [67] Vidal, J. J. (1977) “Real-time detection of brain events in EEG,” *Proc. IEEE*, 65:633–664.
- [68] By Original by en>User:Chris 73, updated by en>User:Diberri, converted to SVG by tiZom [CC BY-SA 3.0 (<http://creativecommons.org/licenses/by-sa/3.0>)], via Wikimedia Commons. https://commons.wikimedia.org/wiki/File%3AAction_potential_basic_shape.svg.

- [69] By Richard A. Normann (US Patent #5,215,088) [Public domain], via Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Utah_array_pat5215088.jpg
- [70] By Mononomic (talk) 16:12, 21 December 2008 (UTC) (en::File:Constudevent.gif) [GFDL (<http://www.gnu.org/copyleft/fdl.html>) or CC-BY-SA-3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>)], via Wikimedia Commons. <https://commons.wikimedia.org/wiki/File%3AComponentsofERP.svg>.
- [71] Uploaded from the German Wikipedia, uploaded into the German Wikipedia by Der Lange 11/6/2005, created by himself. <http://commons.wikimedia.org/w/index.php?title=File:Spike-waves.png&action=edit§ion=2>.
- [72] By Daren Reehl (<http://www.army.mil/-images/2007/01/07/1718/>) [Public domain or Public domain], via Wikimedia Commons. https://commons.wikimedia.org/wiki/File%3AAUS_Army_powered_armor.jpg.
- [73] By Unknown NIMH author [Public domain], via Wikimedia Commons. https://commons.wikimedia.org/wiki/File%3ANIMH_MEG.jpg.
- [74] By Jens Maus (<http://jens-maus.de/>) (Own work) [Public domain], via Wikimedia Commons. <https://commons.wikimedia.org/wiki/File%3APET-image.jpg>.
- [75] By OpenStax College [CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0/>)], via Wikimedia Commons. https://commons.wikimedia.org/wiki/File%3A1206_FMRI.jpg.
- [76] Nissan and EPFL to Push Development For Thought Controlled Vehicles. <https://thinktechuk.wordpress.com/category/bci-wheelchairs/>.
- [77] Author: User:XaosBits at English Wikipedia (2006) Description: Van de Pol phase portrait. Created with Mathematica and touched up with Illustrator. Permission: (Reusing this file) This image is licensed under the Creative Commons Attribution License v. 2.5.
- [78] Author: Zawersh at English Wikipedia (2006) Description: A Hopfield net with four nodes. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2.