

UNIVERSIDADE DE SÃO PAULO
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto
Departamento de Computação e Matemática

**Extração de Conhecimento utilizando Aprendizado de
Máquina**

JOSÉ AUGUSTO BARANAUSKAS

Ribeirão Preto, SP
2016

UNIVERSIDADE DE SÃO PAULO
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto
Departamento de Computação e Matemática

**Extração de Conhecimento utilizando Aprendizado de
Máquina**

JOSÉ AUGUSTO BARANAUSKAS

Documento submetido à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da UNIVERSIDADE DE SÃO PAULO como parte dos requisitos para a obtenção de título de Livre-Docente na Área de Ciências de Computação, especialidade: Inteligência Artificial.

Ribeirão Preto, SP, maio de 2016

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE DOCUMENTO, POR MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Baranauskas, J.A.

Extração de Conhecimento utilizando Aprendizado de Máquina /

José Augusto Baranauskas — Ribeirão Preto, SP, 2016.

48p .: il.

Documento (Livre-Docente. Área de Ciências de Computação, especialidade: Inteligência Artificial) – Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da UNIVERSIDADE DE SÃO PAULO.

1. Aprendizado de Máquina 2. Extração de Conhecimento 3. Modelos Simbólicos

Dedicatória

Ao Cosmos.

Agradecimentos

Agradeço aos orientandos e ex-orientandos, pela troca de conhecimentos: Pedro, Oscar, Thais, Erica, Denise, Lucas, Rogério (*Mineiro*), Maria Izabela, Diego (*Testa*), Patrícia, Luciana, Caio, Juliana, Camila e Rodrigo. Agradeço ainda a oportunidade de tentar contagiá-los com a paixão pela Ciência.

Sou grato aos colegas que propiciaram discussões científicas, em especial, aquelas que me trouxeram dúvidas. Em especial, aos professores Gustavo Batista e Ronaldo Prati pelo caminho percorrido em conjunto.

Agradeço aos co-autores, que trilharam junto comigo o horizonte da Ciência. Em especial, à professora Alessandra Alaniz Macedo e aos professores Francisco de Assis Leone e Sérgio Ricardo Nozawa pela co-autoria de diversos dos trabalhos citados neste texto.

Minha profunda gratidão a todos meus ex-professores e mestres, em especial, à professora Maria Carolina Monard, ao professor André Carvalho e ao professor Adalberto Luis Val.

Meus agradecimentos pela oportunidade de conviver no ambiente acadêmico-científico junto ao Departamento de Computação e Matemática.

Agradeço a todos meus familiares, especialmente aos meus pais, João (*in memorian*) e Wilma, pela minha formação de caráter, pelas oportunidades proporcionadas à mim por meio de muito esforço por parte de ambos. Enorme é o carinho, amor e respeito dos meus sobrinhos Maria Augusta (*in memorian*), Ana Laura, Maria Angélica, Renato e Jhonas Gabriel. Obrigado, Luke (*in memorian*), pelos anos de alegria, aprendizado e amizade.

Meus agradecimentos à USP, CAPES, CNPq, FAPESP e FAPEAM.

Tenho convicção que agradei apenas uma pequena parcela das pessoas a quem devo muito e peço especial desculpa àquelas que omiti. Meu sincero obrigado a todos.

Sumário

Dedicatória	p. i
Agradecimentos	p. iii
Sumário	p. iv
Lista de Figuras	p. vii
Lista de Tabelas	p. ix
Normas e Convenções	p. xi
Resumo	p. xiii
Abstract	p. xv
Citação	p. xvii
1 Introdução	p. 1
1.1 O Programa de Doutorado	p. 1
1.2 Participação em Projetos Temáticos	p. 3
1.3 Considerações Finais	p. 3
2 Extração de Conhecimento	p. 5
2.1 Pré-Processamento	p. 6
2.1.1 Preparação de Dados	p. 6
2.1.2 Redução de Dados	p. 8

2.2	Mineração de Dados	p. 12
2.2.1	Aprendizado Multirrótulo	p. 14
2.2.1.1	Abordagens Independentes de Algoritmo	p. 15
2.2.1.2	Abordagens Dependentes de Algoritmo	p. 18
2.2.2	Ensembles	p. 20
2.2.2.1	Windowing	p. 22
2.2.2.2	Stacking	p. 23
2.2.2.3	Random Forests	p. 23
2.3	Pós-Processamento	p. 25
2.4	Considerações Finais	p. 27
3	Trabalhos Adicionais & em Colaboração	p. 29
3.1	Considerações Finais	p. 31
4	Conclusão	p. 33
4.1	Impacto das Contribuições	p. 33
4.2	Trabalhos Futuros	p. 35
4.3	Considerações Finais	p. 35
	Apêndice A – Mapeamento de Referências	p. 37
	Referências	p. 39

Lista de Figuras

2.1	O processo de Extração de Conhecimento utilizado nesta sistematização . . .	p. 6
2.2	Um exemplo de espaço de estados de atributos, contendo os subconjuntos possíveis para 4 atributos; círculos brancos indicam que o atributo em questão não foi selecionado; círculos escuros indicam seleção do atributo para o processo de indução. Cada estado no domínio de subconjuntos de atributos especifica quais atributos utilizar durante a busca. Nota-se que os estados são parcialmente ordenados onde os estados à direita acrescentam um atributo aos da esquerda.	p. 10
2.3	Exemplo de transformação baseada em rótulo	p. 16
2.4	Criação de novos rótulos	p. 16
2.5	Eliminação de exemplos com mais de um rótulo	p. 17
2.6	Transformação por eliminação de rótulos	p. 17
2.7	Transformação por decomposição de rótulos - método aditivo	p. 18
2.8	Transformação por decomposição de rótulos - método multiplicativo	p. 18
2.9	A região real de separação de classes, indicada pela linha diagonal, é aproximada de formas diferentes por quatro árvores de decisão em (a); a combinação das quatro árvores, em um <i>ensemble</i> , fornece uma região que se aproxima mais da diagonal em (b)	p. 23
2.10	Stacking	p. 24

Lista de Tabelas

2.1	Conjunto de exemplos T no formato atributo-valor, com n exemplos e m atributos. A linha i refere-se ao i -ésimo exemplo ($i = 1, 2, \dots, n$) e a entrada x_{ij} refere-se ao valor do j -ésimo ($j = 1, 2, \dots, m$) atributo X_j do exemplo i	p. 7
2.2	Conjunto de exemplos no formato atributo-valor para problemas multirrótulo	p. 14
A.1	Mapeamento entre referências	p. 37

Normas e Convenções

Este documento foi preparado com o formatador de textos \LaTeX . O sistema de citações de referências bibliográficas utiliza a classe *ieeetr* do \BIBTeX , que segue as recomendações do IEEE (*Institute of Electrical and Electronics Engineers*) para publicação em periódicos da instituição.

A formatação da capa, folha de rosto, folha de aprovação, resumo e *abstract* segue as “diretrizes para apresentação de dissertações e teses da USP”, disponível em <http://www.teses.usp.br>.

A formatação de sumário, lista de figuras e tabelas, lista de abreviaturas e siglas, espaçamento entre linhas, numeração de páginas e cabeçalhos de páginas segue a norma ABNT NBR 14724 para “Apresentação de trabalhos acadêmicos”.

A formatação de títulos e capítulos de seções segue a norma ABNT NBR 6024 para “Numeração progressiva das seções de um documento”.

Todas as formatações que seguem a norma ABNT foram geradas automaticamente utilizando as macros da classe *abntex* disponíveis em <http://abntex.codigolivre.org.br/>.

O mapeamento entre as referências no Memorial e na Documentação com as referências deste documento encontra-se no Apêndice A na página 37.

Resumo

A luta contra o erro tipográfico tem algo de homérico. Durante a revisão os erros se escondem, fazem-se positivamente invisíveis. Mas assim que o livro sai, tornam-se visibilíssimos, verdadeiros sacis a nos botar a língua em todas as páginas. Trata-se de um mistério que a ciência ainda não conseguiu decifrar. . .

— Monteiro Lobato

BARANAUSKAS, J.A.. **Extração de Conhecimento utilizando Aprendizado de Máquina.** Documento (Livre-Docente) – Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, SP, 2016.

Em fevereiro de 1898, logo após uma tempestade, uma amostra de pardais foi levada ao Laboratório de Anatomia do professor Hermon Carey Bumpus na Universidade de Brown, Rhode Island, EUA [1]. Ele encontrou 136 pardais (*Passer domesticus*) machucados no chão e decidiu coletar um conjunto características físicas dos mesmos, dentre elas o comprimento total, comprimento das asas, comprimentos do bico e cabeça, comprimento do úmero, comprimento da envergadura do esterno, dentre outros. Posteriormente, Bumpus observou que apenas 72 pardais sobreviveram e viu neste fato a oportunidade de analisar o processo de seleção natural no qual os pardais que sobreviveram o fizeram porque eles possuíam certas características físicas; por outro lado, os pardais que pereceram, pereceram não por acidente, mas porque eram fisicamente desqualificados. Diante da observação de processos naturais ou artificiais por meio da coleta de informações há duas perguntas que surgem: (i) É possível criar alguma hipótese a partir dos dados coletados? (ii) Dada uma hipótese, como saber se ela generaliza para dados futuros? As respostas a estas perguntas podem ser dadas pela pesquisa em Aprendizado de Máquina (AM) que tem como objetivo desenvolver algoritmos capazes de adquirir conhecimento de forma automática, baseando-se em experiências acumuladas por meio da solução bem sucedida em problemas anteriores. Os algoritmos de AM podem ser categorizados de acordo com o grau de compreensibilidade proporcionado ao ser humano em: (a) sistemas tipo caixa-preta que desenvolvem sua própria representação do conceito, isto é, sua representação interna pode não ser facilmente interpretada por humanos e (b) sistemas orientados a conhecimento que têm como objetivo a criação de estruturas simbólicas que sejam compreensíveis por humanos. De especial interesse neste trabalho estão os sistemas de aprendizado simbólico (orientados a conhecimento) que buscam aprender construindo representações de um conceito tipicamente na forma de alguma expressão lógica, árvore de decisão, regras ou rede semântica. Assim, este trabalho se concentra em sistemas que contribuem para a compreensão dos dados em contraste com indutores que visam apenas uma grande precisão. Um exemplo típico é o desenvolvimento de sistemas especialistas nos quais é importante que especialistas humanos possam, fácil e confiavelmente, verificar o conhecimento extraído e relacioná-lo ao seu próprio conhecimento. Além disso, algoritmos de aprendizado que induzem estruturas compreensíveis, contribuindo para a compreensão do domínio considerado, podem produzir conhecimento novo.

Palavras-chave: Aprendizado de Máquina, Extração de Conhecimento, Modelos Simbólicos.

Abstract

BARANAUSKAS, J.A.. **Knowledge Discovery using Machine Learning**. Manuscript (Livre-Docente) – Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, SP, 2016.

In February 1898, after an uncommonly severe storm, a sample of sparrows were brought to the Anatomical Laboratory of Brown University, Rhode Island, USA, led by Professor Hermon Carey Bumpus [1]. He found 136 sparrows (*Passer domesticus*) injured on the ground and then decided to collect a set of physical characteristics of them, among which the total length, the length of the wings, the length of the beak, the length of the head, humeral length, the length of sternum wingspan, among others. Later, Bumpus noted only 72 sparrows survived and saw in this fact the opportunity to analyze the process of natural selection in which the sparrows who survived did so because they had certain physical characteristics; on the other hand, the sparrows who perished, perished not by accident but because they were physically unfit. Given the observation of natural or artificial processes by collecting information, two questions arise: (i) Is it possible to create a hypothesis from the data collected? (ii) Given one hypothesis, how to know if it generalizes to future data? The answers to these questions can be given by Machine Learning (ML) research which aims to develop algorithms able to acquire knowledge automatically, based on experiences gained through the successful solution of previous problems. ML algorithms can be categorized according to the degree of comprehensibility provided to the human being: (a) black-box systems that develop their own concept of representation, that is, its internal representation can not be easily interpreted by humans and (b) knowledge-oriented systems that aim to create symbolic structures that are understandable by humans. Of particular interest in this work are the symbolic learning systems (knowledge-oriented) that aim to learn symbolic representations of a concept, typically in the form of logical expressions, decision trees, rules or semantic networks. In this way, the work described here focuses on systems that contribute to the understanding of the data in contrast to classifiers that target only high accuracy. A typical example is the development of expert systems in which it is important that human experts can, easily and reliably, verify the extracted knowledge and relate it to their own knowledge domain. In addition, learning algorithms that induce understandable structures, contributing to the understanding of the domain under consideration, can produce new knowledge.

Keywords: Machine Learning, Knowledge Discovery, Symbolic Models.

Citação

O físico britânico Michael Faraday alertou contra a tentação poderosa de procurar as evidências e aparências que estão a favor de nossos desejos e desconsiderar as que lhes fazem oposição [...]. Acolhemos com boa vontade o que concorda com nossas ideias, assim como resistimos com desgosto ao que se opõe a nós, enquanto todo preceito de bom senso exige exatamente o oposto.

— Carl Sagan (O mundo assombrado pelos demônios)

1 *Introdução*

Para mim está bem claro que a Ciência da Computação mudará nossas vidas, mas não por causa dos computadores e sim porque nos ajudará a entender os nossos próprios cérebros e a aprender qual é a natureza do conhecimento. O computador nos ensinará como aprender a pensar e a sentir. Este conhecimento mudará nossa visão da humanidade e nos capacitará a mudarmos a nós mesmos. A Ciência da Computação diz respeito à complexidade e nós somos as coisas mais complexas deste mundo.

— Marvin Minsky

Neste texto encontram-se as atividades de pesquisas realizadas desde que concluí meu programa de doutorado, em agosto de 2001, até a presente data. As atividades são descritas no contexto de projetos que coordenei ou com os quais eu colaborei, além de contar com a contribuição de parceiros pesquisadores e de alunos de iniciação científica e de alunos de mestrado que orientei no período.

1.1 O Programa de Doutorado

Realizei meu programa de doutorado, de fevereiro de 1997 a agosto de 2001, no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP), onde trabalhei sob a supervisão da Professora Maria Carolina Monard. O problema principal investigado em minha pesquisa era o de extrair conhecimento utilizando múltiplos algoritmos de Aprendizado de Máquina — AM. Dentre as contribuições do doutorado incluem-se [2]:

- Sobre seleção de atributos, concluiu-se que filtros deveriam ser considerados antes de se cogitar a utilização de *wrappers*, no caso de existirem muitos atributos para descrever os exemplos [3, 4, 5, 6, 7]. Sob a perspectiva de compreensibilidade sintática do conhecimento induzido, a análise sobre o impacto da seleção de atributos em um classificador simbólico mostrou um aumento do número de regras e do número de condições por regra. A seleção de atributos é uma atividade de pré-processamento de dados que seleciona um

subconjunto de atributos do conjunto original de exemplos. Existem, basicamente, três abordagens que são empregadas para a seleção de atributos: embutida, filtro e *wrapper*; as duas últimas investigadas no doutorado.

- A definição de um *framework* para composição de atributos guiada pelo conhecimento [8]. Os resultados de experimentos realizados utilizando a metodologia proposta em repositórios naturais (pré-processados) mostraram que, mesmo com o auxílio do especialista no domínio do conhecimento, não foi simples construir atributos derivados que fossem realmente relevantes para aprender o conceito embutido nos conjuntos de exemplos analisados. Entretanto, com dados do mundo real, a metodologia proposta mostrou potencial. A composição de atributos, também conhecida como indução construtiva, é outra atividade de pré-processamento de dados. Há várias abordagens de composição de atributos (guiada por dados, por hipótese, por conhecimento e multi-estratégia); aquela guiada pelo conhecimento foi investigada no doutorado.
- A definição de um *framework* de combinação de múltiplos classificadores simbólicos em um classificador final também simbólico [9, 10, 11]. A combinação de classificadores, simbólicos ou não, é uma atividade de mineração de dados. Na realidade, uma das preocupações das pesquisas em Aprendizado de Máquina simbólico é que os classificadores induzidos sejam fáceis de serem compreendidos pelos seres humanos. Para isso, deve-se escolher o indutor com viés mais adequado para cada tipo de situação, já que pesquisas da literatura de área mostraram que não existe o *melhor* indutor para todos os domínios. Aliada a essa escolha, é possível fazer uso de vários classificadores, combinando-os em um único classificador final, formando um *ensemble*. Os *ensembles* possuem a tendência de melhorar o desempenho na classificação de exemplos não vistos durante o processo de aprendizado. Entretanto, o emprego de *ensembles* dificulta a compreensão humana sobre o comportamento do classificador final, já que ele deixa de ser simbólico, mesmo assumindo que cada classificador individual que o compõe seja simbólico. Na realidade, a combinação de classificadores simbólicos — provenientes de diferentes indutores — em um classificador final também simbólico era (e ainda o é) um tópico novo de pesquisa, ainda com poucos resultados divulgados. Com o objetivo de preencher essa lacuna, foi proposto e desenvolvido no doutorado o *framework* XRULER. Inicialmente foi definido um formato padrão de regras, o qual forneceu uma perspectiva unificada sob a qual todo classificador simbólico poderia ser convertido e analisado [12, 13]. Dentre outros componentes, o *framework* XRULER possuía um algoritmo de cobertura que poderia ser aplicado ao conjunto de regras induzidas por diversos indutores simbólicos para se obter um classificador final também simbólico. Os resultados obtidos com o *framework* XRU-

LER mostraram aumento da precisão e redução do número de regras. Isso, na época, foi considerado um possível avanço no sentido de uma maior compreensibilidade, por parte seres humanos, no classificador final obtido. Os resultados do *framework* XRULER foram publicados em [14].

1.2 Participação em Projetos Temáticos

No final de 2008, participei da elaboração da proposta do projeto que se tornou, a partir de 2009 o INCT Adapta¹ que tinha coordenação geral do Prof. Dr. Adalberto Luis Val do Instituto Nacional de Pesquisas da Amazônia e minha coordenação institucional do laboratório de mineração de dados de bioinformática. Desde 2012 pude colaborar também com o projeto do NAP Aprendizado de Máquina em Análise de Dados (NAP AMAD), sob coordenação geral do Prof. Dr. André Carvalho, do ICMC-USP. Esses dois projeto serviram de nicho para o desenvolvimento da maior parte da pesquisa que realizei no período 2009 a 2015.

Neste sentido, investi esforços no projeto de algoritmos de aprendizado de máquina que explorassem o pré-processamento (redução) dos dados bem como na extração de modelos simbólicos nos domínios biomédico e biológico, em particular no contexto de algoritmos que permitam que o modelo gerado seja interpretado por um especialista humano no domínio do conhecimento.

1.3 Considerações Finais

Toda a pesquisa realizada tem como tema comum a aplicação de modelos (classificadores) simbólicos e, assim, tem apresentado resultados no contexto de melhorar a compreensão do comportamento dos algoritmos de aprendizado existentes bem como encontrar formas aprimoradas de aplicação desses algoritmos. Meu trabalho se concentra em aprendizado simbólico supervisionado, onde o termo *simbólico* indica que os classificadores, sempre que possível, devem ser legíveis e interpretáveis por humanos. O termo *supervisionado* sugere que algum processo, às vezes denominado *agente externo* ou *professor*, previamente rotulou os dados.

Neste texto, procuro apresentar minhas contribuições em termos de publicações, formação de recursos humanos, captação de recursos junto a agências de fomento à pesquisa, formação de grupo de pesquisa e fortalecimento da área a nível institucional e nacional.

O restante deste texto é organizado como segue. Na Seção 2 é detalhado o trabalho de

¹<http://adapta.inpa.gov.br/>

pesquisa diretamente relacionado à Extração de Conhecimento utilizando algoritmos de Aprendizado de Máquina. Na Seção 3 é detalhada a pesquisa realizada individualmente ou com a parceria de colaboradores. Na Seção 4 é discutido o impacto do trabalho reportado, contextualizando a continuidade da pesquisa e apresenta minhas considerações finais.

2 *Extração de Conhecimento*

Não é o mais forte que sobrevive, nem o mais inteligente, mas o que melhor se adapta às mudanças.

— Charles Robert Darwin

Em 2012, a companhia IBM estimou que o mundo criava todos os dias 2,5 exabytes; também estimou que 90% de toda informação tinha sido criada nos dois anos imediatamente anteriores. Os dados provêm de sensores utilizados para coletar dados climáticos, *posts* em *sites* de conteúdo social, registro de transações comerciais, sinais de GPS de celulares, etc¹.

Quando a quantidade de dados é muito grande (muitas vezes denominado *big data* na literatura), tentativas de extrair conhecimento manualmente da conjuntos de dados são, geralmente, improdutivas. Normalmente, recorre-se a uma metodologia computacional que automatize o processo de manipulação de dados, visando a extração de informação útil. Inicialmente, foi proposta uma metodologia por [15], composta por nove etapas e, mais tarde, outra por [16], composta por quatro etapas. No meu doutorado, fiz uma combinação e compactação dessas duas metodologias, que favorece a compreensão geral do processo de Extração de Conhecimento — EC — resultando nas seguintes três etapas: (i) pré-processamento; (ii) mineração e (iii) pós-processamento. O foco na minha pesquisa atual engloba essas três etapas do processo de EC.

Em resumo, a etapa de pré-processamento, na qual se preparam os dados para o processo de mineração, pode ser entendida como duas sub-etapas. Na sub-etapa de preparação, os dados são coletados e transformados para início do processo de mineração. A sub-etapa de redução, opcional quando a quantidade de dados é moderada, diminui a quantidade de dados de forma a viabilizar a aplicação da etapa de mineração em grandes bases. A etapa de mineração de dados procura por soluções que podem ter diferentes objetivos e complexidade. Por último, a etapa de pós-processamento ou análise das soluções obtidas é efetuada, consolidando os resultados obtidos em uma solução final que será apresentada ao usuário. É possível retornar de uma etapa para etapas anteriores, visando uma melhoria dos resultados que somente tenha sido percebida

¹<http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html> acesso em 23/05/2016.

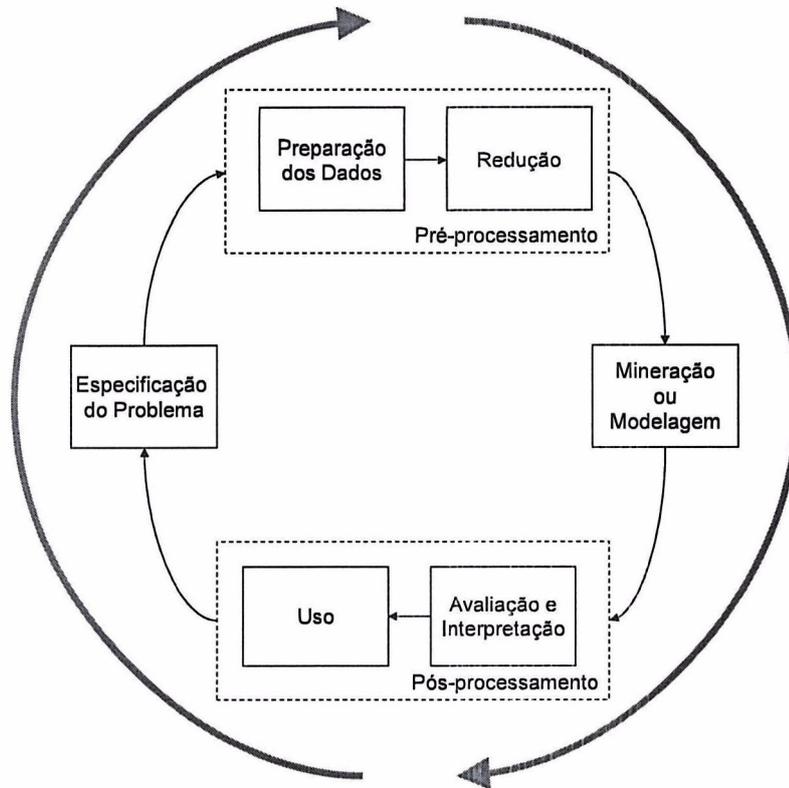


Figura 2.1: O processo de Extração de Conhecimento utilizado nesta sistematização

como possível em uma etapa mais adiantada, conforme esquema apresentado na Figura 2.1.

Nas seções seguintes há uma sistematização dessas etapas do processo de EC, apresentando minhas contribuições em termos de publicações e formação de recursos humanos do meu grupo de pesquisa. Apesar das contribuições estarem separadas dentre as etapas, ressalta-se que essa separação não é estanque ou isolada já que, em várias contribuições, mais de uma etapa foi utilizada. Neste caso, as publicações foram mencionadas na etapa mais aproximada em termos de contribuição da pesquisa desenvolvida.

2.1 Pré-Processamento

2.1.1 Preparação de Dados

Esta sub-etapa frequentemente recebe pouca atenção na literatura pelo fato de ser considerada muito específica pois, na prática, requer realizar um *reconhecimento do domínio da aplicação* considerando-se o conhecimento prévio relevante e os objetivos da aplicação.

Nesta sub-etapa, os dados brutos são transformados no formato atributo-valor definido na

Tabela 2.1, ou seja, no *formato padrão*. Embora outros formatos possam ser definidos, o formato padrão adotado coloca os dados originais em uma representação simples e uniforme que é universalmente aceitável para várias técnicas de Mineração de Dados que utilizando algoritmos de Aprendizado de Máquina.

Tabela 2.1: Conjunto de exemplos T no formato atributo-valor, com n exemplos e m atributos. A linha i refere-se ao i -ésimo exemplo ($i = 1, 2, \dots, n$) e a entrada x_{ij} refere-se ao valor do j -ésimo ($j = 1, 2, \dots, m$) atributo X_j do exemplo i .

Exemplo	X_1	X_2	\dots	X_m	Y
z_1	x_{11}	x_{12}	\dots	x_{1m}	y_1
z_2	x_{21}	x_{22}	\dots	x_{2m}	y_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
z_n	x_{n1}	x_{n2}	\dots	x_{nm}	y_n

Assim, exemplos são tuplas $z_i = (x_{i1}, x_{i2}, \dots, x_{im}, y_i) = (\vec{x}_i, y_i)$ também denotados por (x_i, y_i) , onde fica subentendido o fato que x_i é um vetor. A última coluna, $y_i = f(x_i)$, é a função que tenta-se prever a partir dos atributos. Observa-se que cada x_i é um elemento do conjunto $X_1 \times X_2 \times \dots \times X_m$ e y_i pertence a uma das k classes, isto é, $y_i \in \{C_1, C_2, \dots, C_k\}$ no caso de *classificação* ou y_i assume valores reais no caso de *regressão*.

Os dados brutos nem sempre são adequados para a etapa de Mineração de Dados. Várias *transformações* podem ser necessárias para produzir atributos e exemplos com um bom poder de predição ao final da etapa de pré-processamento. Dentre essas transformações, incluem-se:

- (i) **Definição de atributos:** com base nos dados brutos e no conhecimento prévio do domínio, são definidos os atributos importantes para atingir a meta do processo de EC;
- (ii) **Extração e integração:** os dados brutos podem se encontrar sob diferentes formas de armazenamento, tais como arquivos, base de dados, *clouds* e é necessário realizar a extração e integração dos dados provenientes de diferentes fontes em diferentes formatos, para o formato padrão;
- (iii) **Transformação de dados:** após a extração e integração, os dados brutos devem ser transformados em dados mais apropriados para EC, por exemplo, dados sobre vendas individuais podem ter sido armazenados, mas um resumo diário talvez seja mais indicado para a tarefa em questão, a transformação de tipos de dados para um algoritmo de aprendizado específico que pode não ser capaz de lidar com atributos do tipo data, etc;
- (iv) **Limpeza:** A qualidade dos dados é um ponto central em EC, uma vez que muitos indutores não são capazes de considerar informações adicionais durante a fase de aprendizado.

Há tarefas específicas do domínio, que são solucionadas por técnicas que utilizam o conhecimento do domínio, como verificação de consistência dos atributos. Há tarefas independentes do domínio, que podem ser automatizadas, tais como a decisão da estratégia para o tratamento de atributos com valores ausentes, remoção de ruído, etc;

- (v) **Composição de atributos:** Em alguns casos, existem transformações adicionais que podem apresentar um impacto muito grande nos resultados. Neste sentido, a composição de atributos é um fator determinante na qualidade dos resultados, muito maior do que o próprio método de mineração adotado para produzir os resultados. Em muitos casos, a composição de atributos é dependente do domínio da aplicação.

A transformação e a composição de atributos foram temas de pesquisa que envolveram dois alunos de iniciação científica:

- O bolsista FAPESP Rogério Nunes Lemos pesquisou transformação de dados ao desenvolver uma técnica de arredondamento de valores de contínuos, avaliando seu impacto na indução de árvores decisão, publicando os resultados no congresso de iniciação científica da USP [17] e no Congresso Brasileiro de Informática em Saúde [18] e
- A bolsista CNPq Thais Mayumi Oshiro, analisou a composição de atributos na identificação do sexo de pessoas não identificadas por meio de radiografias faciais, publicando seus resultados no *Workshop* de Informática Médica do Congresso da SBC [19].

2.1.2 Redução de Dados

Considerando a etapa de preparação de dados, é possível que uma grande quantidade de dados brutos resulte em um conjunto de exemplos, no formato padrão, de tamanho relativamente moderado. Neste caso, é possível aplicar algoritmos de mineração diretamente. Por outro lado, com dados atingindo a ordem de terabytes, petabytes ou exabytes, é muito simples exceder a capacidade de processamento de um algoritmo de aprendizado ou mesmo o tempo disponível para encontrar uma solução. Nesse caso, a etapa de redução de dados é necessária antes da utilização dos algoritmos de mineração.

A redução de dados é uma etapa mediadora entre a preparação de dados e a Mineração de Dados. Entretanto, é possível considerar técnicas que removem atributos como métodos para preparação de dados. De forma similar, métodos que transformam dados em um novo conjunto de atributos podem também ser considerados como métodos de preparação de dados.

Para a maioria das aplicações, o conjunto de exemplos no formato padrão possui muito mais exemplos do que atributos. A tendência, para um grande volume de dados, é que o número de exemplos cresça muito mais rapidamente do que o número de atributos. Dessa forma, a remoção de um atributo tem um impacto muito maior na redução de dados do que a remoção de um exemplo. Nesse caso, as abordagens de redução de dados são também denominadas seleção de atributos.

Segundo [20], uma forma conveniente para representar as abordagens para a seleção de atributos é a busca heurística, na qual cada estado no espaço de busca é composto por um subconjunto de possíveis atributos. Desse modo, qualquer método de seleção de atributos pode ser caracterizado como um processo de busca heurística. Por exemplo, na Figura 2.2 na próxima página é ilustrado o espaço de busca para quatro atributos, representados por uma sequência de quatro círculos [21]. Pode-se observar que existe uma ordem parcial entre os estados, pois cada um deles possui um atributo a mais (círculos escuros) que o estado anterior, sendo o estado inicial, mais à esquerda, o conjunto vazio de atributos, representado pelos quatro círculos em branco. Essa abordagem é geralmente conhecida como seleção *forward*. Já a abordagem que inicia o ponto de partida com o subconjunto contendo todos os atributos e sucessivamente os remove, é denominada de eliminação *backward*. Também podem ser empregadas variações de ambas abordagens, selecionando-se um estado inicial em algum ponto do espaço de busca e movendo-se a partir desse ponto — seleção *outward*.

A seleção de atributos úteis é, intrinsecamente, um dos problemas principais em AM. Embora a maioria dos algoritmos de aprendizado tente selecionar atributos, ou então atribuir graus de importância a eles, análises teóricas e estudos experimentais indicam que muitos algoritmos apresentam um comportamento ruim com a presença de um grande número de atributos irrelevantes. Por exemplo, o número de exemplos necessário para o algoritmo K -NN (K vizinhos mais próximos) atingir um dado nível de precisão parece crescer exponencialmente com o número de atributos irrelevantes, independente do conceito a ser aprendido. Mesmo os indutores de árvores de decisão, que explicitamente selecionam alguns atributos em detrimento de outros, exibem este comportamento para alguns conceitos [22]. Outros algoritmos, tais como *Naïve Bayes*, são robustos com relação à presença de atributos irrelevantes, mas podem ser extremamente sensíveis em domínios contendo atributos fortemente correlacionados, mesmo se os atributos são relevantes. Isto pode ser explicado pelo fato que esses algoritmos assumem independência entre os atributos. Assim, isso sugere a necessidade de métodos adicionais para selecionar um subconjunto útil de atributos quando muitos deles estão disponíveis [21]. Isso sugere a necessidade de métodos adicionais para redução de dados quando muitos atributos estão disponíveis [23].

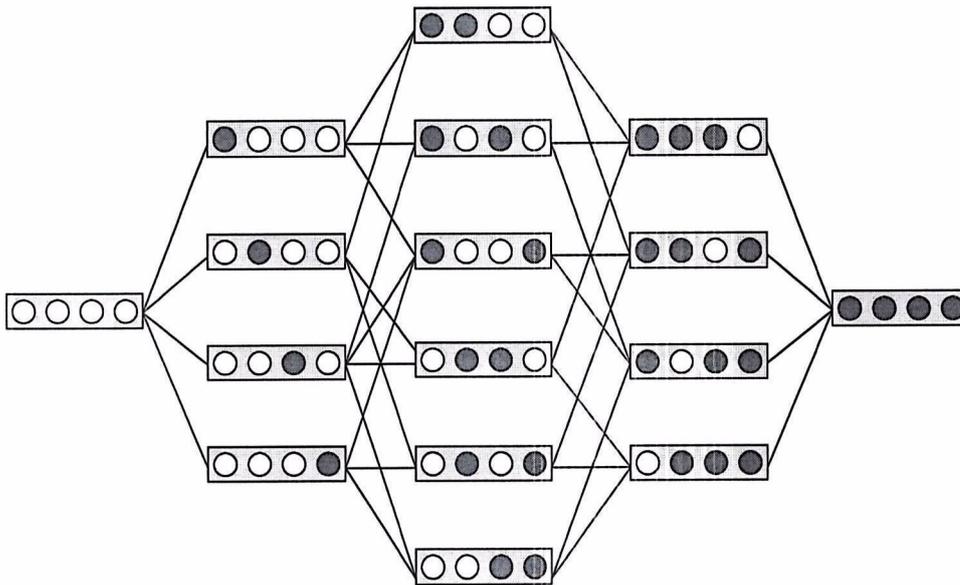


Figura 2.2: Um exemplo de espaço de estados de atributos, contendo os subconjuntos possíveis para 4 atributos; círculos brancos indicam que o atributo em questão não foi selecionado; círculos escuros indicam seleção do atributo para o processo de indução. Cada estado no domínio de subconjuntos de atributos especifica quais atributos utilizar durante a busca. Nota-se que os estados são parcialmente ordenados onde os estados à direita acrescentam um atributo aos da esquerda.

Por exemplo, os domínios médico e biológico impõem obstáculos difíceis para os algoritmos de aprendizado tais como alta dimensionalidade, uma grande ou uma pequena quantidade de exemplos, muitos valores possíveis para a classe, classes desbalanceadas, etc. Isso pode explicar o fato de que a pesquisa a respeito de seleção de atributos vem sendo amplamente trabalhada na literatura atualmente [24, 25, 26, 27, 28].

Há, basicamente, três abordagens que podem ser utilizadas para redução de dados [20, 29, 30, 31, 32]):

- (i) **embutida:** alguns indutores são capazes de realizar sua própria seleção de atributos de forma dinâmica, enquanto procuram por um classificador, sendo a redução de dados parte integral desses indutores.

Em geral, a maioria dos algoritmos ávidos (*eager*) [33] possuem uma abordagem embutida para a seleção de atributos. Por exemplo, métodos de particionamento recursivo, tais como árvores de decisão, efetuam uma busca gulosa através do espaço de árvores. A cada passo, eles usam uma função de avaliação para selecionar o atributo que tem a melhor capacidade de discriminar entre as classes. Eles particionam o conjunto de treinamento baseados nesse atributo e repetem o processo para cada subconjunto, estendendo a

árvore até que nenhuma discriminação adicional seja possível. Este método é usado pelo indutor C4.5 [34].

Métodos de separar-e-conquistar para indução de regras também possuem seleção embutida de atributos. Estes métodos usam uma função de avaliação para selecionar o atributo que ajuda a distinguir uma classe C das outras; então eles adicionam o teste resultante em uma única regra conjuntiva para essa classe C . Eles repetem esse processo até que a regra exclua todos os exemplos de outras classes e então removem os exemplos da classe C que a regra cobre, repetindo esse processo nos exemplos de treinamento remanescentes. Este método é empregado pelo indutor CN2 [35, 36].

- (ii) **filtro**: processo separado, o qual ocorre antes da aplicação do algoritmo de indução propriamente dito. A ideia é filtrar atributos irrelevantes, segundo algum critério, antes de iniciar a indução [37]. Esse passo de pré-processamento considera características gerais do conjunto de exemplos para selecionar alguns atributos e excluir outros. Sendo assim, métodos de filtros são independentes do algoritmo de indução que, simplesmente, receberá como entrada o conjunto de exemplos contendo apenas os atributos selecionados pelo filtro.

Segundo [20], um dos esquemas mais simples de filtragem é a avaliação de cada atributo individualmente, baseada na sua correlação com a classe, escolhendo os p atributos que fornecem o melhor valor. Esse método é comumente empregado em tarefas de categorização de textos, geralmente combinando filtro com algoritmos *Naïve Bayes* [38] ou K -NN [39, 40], os quais têm mostrado bons resultados empíricos.

Um ponto principal dessa abordagem é que os filtros ignoram totalmente os efeitos do subconjunto de atributos no desempenho do indutor ao qual os dados reduzidos serão submetidos. Por esse motivo, [29] propõe que filtros sejam substituídos pela abordagem *wrapper*, a qual considera o próprio indutor no processo de seleção.

- (iii) **wrapper**: a abordagem *wrapper* gera um subconjunto de atributos como candidato, executa o indutor com apenas esses atributos no conjunto de treinamento e usa a precisão do classificador extraído para avaliar o subconjunto de atributos em questão. Este processo é repetido para cada subconjunto candidato, até que o critério de parada seja satisfeito.

O algoritmo de seleção de atributos existe como um *wrapper* ao redor do indutor e é responsável por conduzir a busca por um bom subconjunto de atributos. A qualidade de um subconjunto candidato é avaliada utilizando o próprio indutor como uma caixa-preta. O objetivo da busca é encontrar o subconjunto (nó) com a melhor qualidade, utilizando uma função heurística para guiá-la.

Por exemplo, a busca pode ser conduzida no espaço do subconjunto de atributos, com os operadores adicionar ou remover, utilizando como busca o método *hill-climbing* ou o método *best-first* assim com direções *forward* e *backward* como direção da busca. Em geral, *cross-validation* é o método tradicional usado para estimar a precisão de um nó [41].

Um argumento a favor da abordagem *wrapper* é que o mesmo algoritmo de indução que vai utilizar o subconjunto de atributos selecionados deve prover uma estimativa melhor de precisão que outro algoritmo, o qual pode possuir um *bias* de indução totalmente diferente [29]. Por outro lado, essa abordagem pode ser computacionalmente dispendiosa, uma vez que o indutor deve ser executado para cada subconjunto de atributos considerado.

Existem também, na literatura, abordagens híbridas (filtro e *wrapper*), tentando otimizar a eficiência do processo de seleção de atributos [32, 42, 43, 44].

Em qualquer dessas abordagens, o conjunto de exemplos inicial no formato padrão é analisado e processado para produzir um subconjunto do conjunto original, mas com menos atributos. Isso deve ser efetuado de forma relativamente rápida (em relação ao prazo para término da tarefa em questão), reduzindo o número de atributos ou a quantidade de valores de atributos de forma aceitável à etapa de mineração.

Nesta linha de pesquisa, pode envolver a orientação de um aluno de mestrado:

- O aluno Oscar Picchi Netto (bolsista CAPES) desenvolveu um algoritmo de filtro iterativo utilizando árvores de decisão, publicando os resultados em dois artigos junto a dois *Workshops* de Informática Médica do Congresso da SBC [22, 45].

2.2 Mineração de Dados

Existem muitos algoritmos disponíveis para a Mineração de Dados. Independente do algoritmo escolhido, a Mineração de Dados adequada pode exigir muita experimentação de forma a ajustar o método para o melhor desempenho ou outras características desejáveis, tais como simplicidade da solução. Os experimentos podem ser vistos como uma busca em um espaço de estados consistindo de combinações de parâmetros e opções. Os nós deste espaço, representando combinações, podem ser testados em sequência e a variação que fornece o melhor resultado é selecionada. Embora possam existir muitos parâmetros a serem ajustados, a experiência sugere que poucos parâmetros resultam em um grande impacto sobre o resultado final. Esta sub-etapa pode vista em três componentes:

- (i) **função:** envolve a decisão sobre o *objetivo* da solução obtida pelo algoritmo de mineração, por exemplo, resumo, classificação, regressão, categorização ou mesmo visualização. Associada à escolha da função, encontra-se a decisão sobre *representação* da hipótese. As representações mais comuns incluem árvores de decisão, regras, modelos lineares, modelos não lineares (*e.g.*, redes neurais), modelos baseados em exemplos (*e.g.*, *K-NN* e raciocínio baseado em casos), modelos de dependência probabilística, tais como redes Bayesianas e modelos relacionais. A representação da solução determina tanto a capacidade de representação do conceito embutido nos dados, como a facilidade de interpretação do modelo em termos humanos. Geralmente, quanto mais complexa é a solução, melhor ela se ajusta aos dados, mas também pode ser mais difícil de interpretar;
- (ii) **algoritmo:** envolve a escolha dos algoritmos a serem usados, bem como a decisão de quais parâmetros podem ser apropriados, de forma a compatibilizar o método de mineração ao processo geral de Extração de Conhecimento (por exemplo, o usuário pode estar mais interessado em entender o modelo do que na sua capacidade de predição) e
- (iii) **mineração:** busca por padrões de interesse em uma forma de representação particular ou em um conjunto de tais representações, incluindo árvores ou regras de classificação, regressão, categorização, visualização, entre outras. O termo *Mineração de Dados* refere-se ao processo de resolução de problemas que encontra uma descrição lógica ou matemática, eventualmente de natureza complexa, de padrões e regularidades em um conjunto de exemplos. De forma mais simples, mineração é a aplicação de algoritmos que extraem informações potencialmente úteis dos dados.

Na etapa de mineração, em grande parte de minha pesquisa, são utilizados algoritmos de Aprendizado de Máquina — AM. Formalmente, em Aprendizado de Máquina supervisionado, um exemplo é um par $(x_i, f(x_i))$ onde x_i é a entrada e $f(x_i)$ é a saída, ou seja, seguindo o formato padrão definido na Tabela 2.1 na página 7. A tarefa de um indutor (ou algoritmo de aprendizado) é, dado um conjunto de exemplos, induzir uma função h que aproxima f , normalmente desconhecida [46]. Neste caso, h é chamada uma *hipótese* (modelo ou classificador) sobre a função objetivo f , ou seja, $h(x_i) \approx f(x_i)$.

Essa foi a abordagem adotada por mim em termos de implementações nesta frente de trabalho e envolveu vários alunos sob minha orientação, com mais detalhes apresentados nas seções seguintes.

2.2.1 Aprendizado Multirrótulo

A classificação de único rótulo refere-se à classificação tradicional, na qual há somente um único rótulo a ser predito. O princípio básico da classificação multirrótulo é similar ao da classificação convencional, sendo diferenciada no número de rótulos a serem preditos, no qual há dois ou mais. Por exemplo, um artigo no jornal pode ser classificado nas categorias música e cultura, uma proteína pode ser classificada como tendo mais de uma função.

Na Tabela 2.2 é mostrada uma modificação do formato atributo-valor para tratar problemas multirrótulo, estendendo a definição do formato padrão mostrado na Tabela 2.1 na página 7. Como antes, dados são caracterizados por n exemplos z_1, z_2, \dots, z_n , cada um contendo m atributos X_1, X_2, \dots, X_m e c rótulos Y_1, Y_2, \dots, Y_c . Nesta tabela, a linha i refere-se ao i -ésimo exemplo ($i = 1, 2, \dots, n$), a entrada x_{ij} refere-se ao valor do j -ésimo ($j = 1, 2, \dots, m$) atributo X_j do exemplo i e a saída y_{ki} refere-se ao valor do k -ésimo ($k = 1, 2, \dots, c$) rótulo (classe) Y_k do exemplo i .

Como pode ser observado, neste caso, exemplos são tuplas $\vec{z}_i = (x_{i1}, x_{i2}, \dots, x_{im}, y_{i1}, y_{i2}, \dots, y_{ic}) = (\vec{x}_i, \vec{y}_i)$ também denotados por $z_i = (x_i, y_i)$, onde fica subentendido o fato que o z_i , x_i e y_i são vetores. Observa-se que cada y_i é um elemento do conjunto $Y_1 \times Y_2 \times \dots \times Y_c$.

Diferentes abordagens têm sido propostas na literatura para tratar problemas de classificação multirrótulo. Em algumas delas, classificadores único rótulo podem ser combinados para tratar problemas de classificação multirrótulo. Outras abordagens modificam classificadores único rótulo, por meio de adaptações em seus algoritmos, para permitir a utilização em problemas multirrótulo. Algumas abordagens presentes na literatura para tratar problemas de classificação multirrótulo são:

Tabela 2.2: Conjunto de exemplos no formato atributo-valor para problemas multirrótulo

Exemplo	X_1	X_2	\dots	X_m	Y_1	Y_2	\dots	Y_c
z_1	x_{11}	x_{12}	\dots	x_{1m}	y_{11}	y_{12}	\dots	y_{1c}
z_2	x_{21}	x_{22}	\dots	x_{2m}	y_{21}	y_{22}	\dots	y_{2c}
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
z_n	x_{n1}	x_{n2}	\dots	x_{nm}	y_{n1}	y_{n2}	\dots	y_{nc}

2.2.1.1 Abordagens Independentes de Algoritmo

A abordagem independente de algoritmo lida com o problema transformando-o em um conjunto de problemas único rótulo, isto é, o conjunto de dados multirrótulo é transformado em um ou mais conjuntos de dados único-rótulo, dependendo do tipo de transformação. Após essa transformação é realizada a aplicação de algum algoritmo de classificação no(s) conjunto(s) de dados único-rótulo, para assim induzir um ou um conjunto classificadores para prever todos rótulos de um novo exemplo. Essa transformação pode ser realizada baseando-se em rótulo ou em exemplo.

(i) Transformação Baseada em Rótulo:

Nesse tipo de transformação, o conjunto de exemplos original é dividido em conjuntos de exemplos no qual cada um contém todos os atributos e seus valores para cada exemplo, mas contendo apenas um dos rótulos a ser predito. Portanto, são utilizados c classificadores, sendo c o número de rótulos do problema e cada classificador gerado é treinado para distinguir um rótulo contra todos os demais rótulos envolvidos. Essa técnica é também chamada de técnica binária ou *One-versus-All* (OVA) ou *Binary-Relevance* (BR) [47]. Porém, essa técnica assume que os rótulos são independentes entre si, algo que nem sempre é verdade, já que ignorar as possíveis relações entre os rótulos pode resultar em uma baixa capacidade de generalização [48].

Um exemplo desta técnica encontra-se na Figura 2.3 na qual é apresentado um problema multirrótulo de classificação de funções de proteínas com três rótulos “Estrutural”, “Hormonal” e “Reguladora”. O problema é dividido em três problemas binários, gerando três classificadores sendo que o i -ésimo classificador ($i = 1, 2, 3$) é treinado para considerar os exemplos pertencentes ao i -ésimo rótulo como positivos e os outros como negativos a fim de distinguir o i -ésimo rótulo dos demais. Essa técnica pode ser computacionalmente dispendiosa, dependendo no número de rótulos do problema. No trabalho de [49] é proposta uma extensão do método BR, denominada BR+, no qual considera as relações entre os rótulos.

(ii) Transformação Baseada em Exemplos

Nesse tipo de transformação, o problema multirrótulo é transformado em um ou mais problemas de único rótulo, baseando-se no conjunto de rótulos associados a cada exemplo. Três diferentes estratégias são conhecidas para esse tipo de transformação, sendo uma delas baseada na criação de rótulos. Essa estratégia pode ser chamada de *Label-Powerset* [47] e se baseia na combinação de mais de um rótulo para criar um novo rótulo;

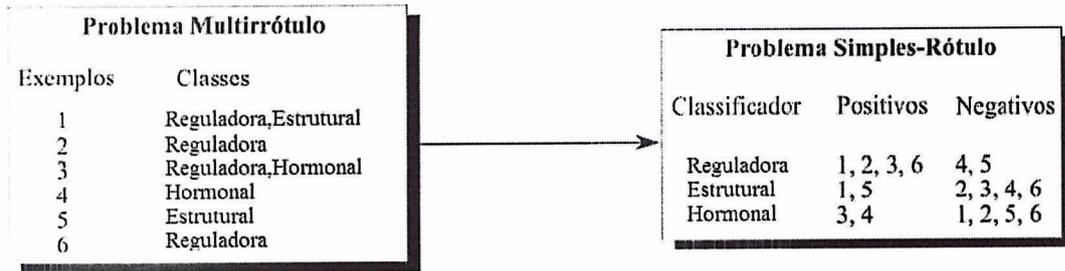


Figura 2.3: Exemplo de transformação baseada em rótulo

porém o número de rótulos pode aumentar consideravelmente e alguns podem terminar com poucos exemplos. Pode-se observar na Figura 2.4 que os rótulos criados “Reguladora_Estrutural” e “Reguladora_Hormonal” têm apenas um exemplo.

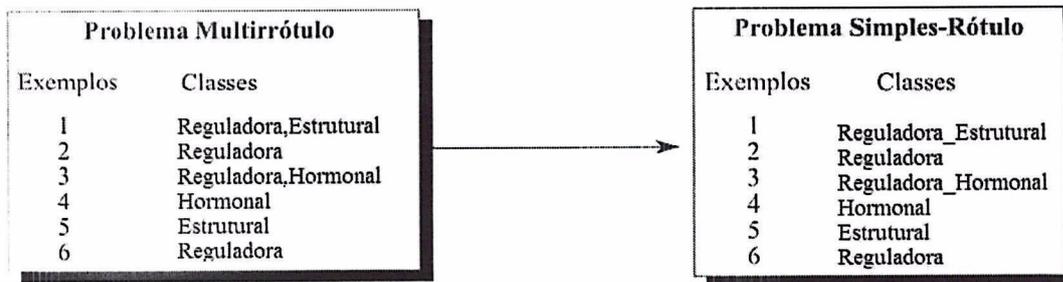


Figura 2.4: Criação de novos rótulos

No trabalho de [50] é apresentado o algoritmo RAKEL (RANDOM k-labELsets) que constrói um *ensemble* de classificadores *Label-Powerset* (LP) e cada classificador LP é treinado com um pequeno subconjunto k aleatório de rótulos. Esse método leva em conta as relações dos rótulos e, ao mesmo tempo, evita os problemas do LP citados acima. O método RAKEL apresentou melhor desempenho em comparação aos métodos BR e LP.

Outra estratégia é a eliminação de exemplos, sendo a estratégia mais simples e menos eficaz que se baseia-se na retirada de exemplos que contenham mais de um rótulo; assim o problema multirrótulo deixa de existir, como ilustrado na Figura 2.5. Porém, ocorre uma perda de informação que pode ser relevante para o problema abordado.

A terceira estratégia é a conversão de exemplos, sendo baseada na conversão de exemplos multirrótulos em exemplos único rótulo e existem duas variações dessa estratégia: eliminação e decomposição. Na primeira variação, todos os exemplos com mais de um rótulo são convertidos em exemplos único rótulo, escolhendo um dos rótulos associados ao exemplo e simplesmente eliminando os demais, gerando assim uma perda de

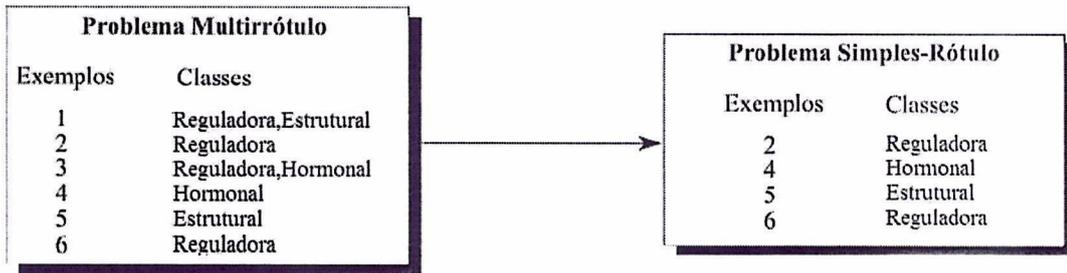


Figura 2.5: Eliminação de exemplos com mais de um rótulo

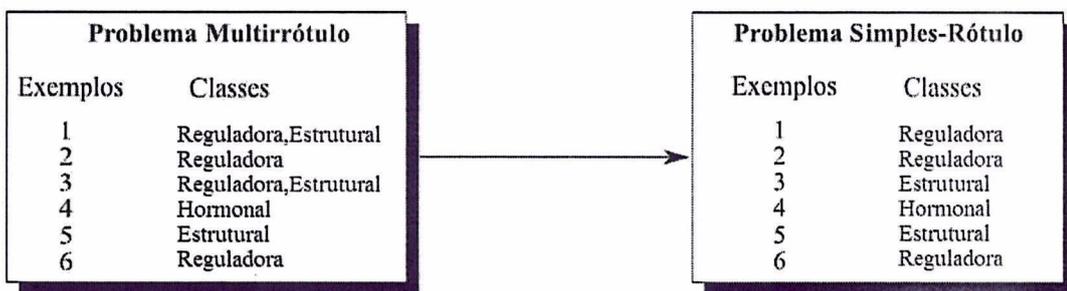


Figura 2.6: Transformação por eliminação de rótulos

informação, como pode ser observado na Figura 2.6. Essa escolha pode ser feita de modo determinista ou aleatória. Na segunda variação, a conversão dos exemplo com mais de um rótulo divide o problema multirrótulo com c rótulos e N exemplos em K conjuntos de problemas de único rótulo. O valor de K varia de 1 (quando nenhum exemplo possui mais de que um rótulo), a $(c - 1)N$, se todos os exemplos possuem $c - 1$ rótulos. Esse processo decomposição pode ser dividido em dois métodos: aditivo e multiplicativo. O método aditivo, ilustrado na Figura 2.7, considera que para cada exemplo, cada um dos possíveis rótulos será o rótulo positivo em sequência; esse método é também conhecido como *cross-training* [46]. Por exemplo, se os rótulos “Reguladora” e “Estrutural” aparecem nos exemplos multirrótulo, quando o classificador para o rótulo “Reguladora” for treinado, todos os exemplos multirrótulos que possuem o rótulo “Reguladora” se tornam exemplos único-rótulo para o rótulo “Reguladora” e o mesmo acontece para os outros rótulos.

No método multiplicativo é realizada uma combinação de todos os possíveis problemas único rótulo. O número de classificadores nesse método é igual ao $\prod l_i$, que é o produto do número de rótulos presentes em cada exemplo i . É ilustrado na Figura 2.8 um exemplo desse método no qual o número de classificadores é dado pelo produto $2 \times 1 \times 2 \times 1 \times 1 \times 1 = 4$ no qual cada número corresponde à quantidade de rótulos que cada exemplo

possui.

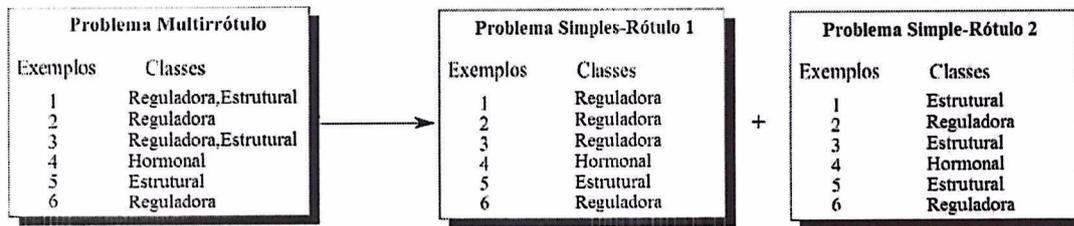


Figura 2.7: Transformação por decomposição de rótulos - método aditivo

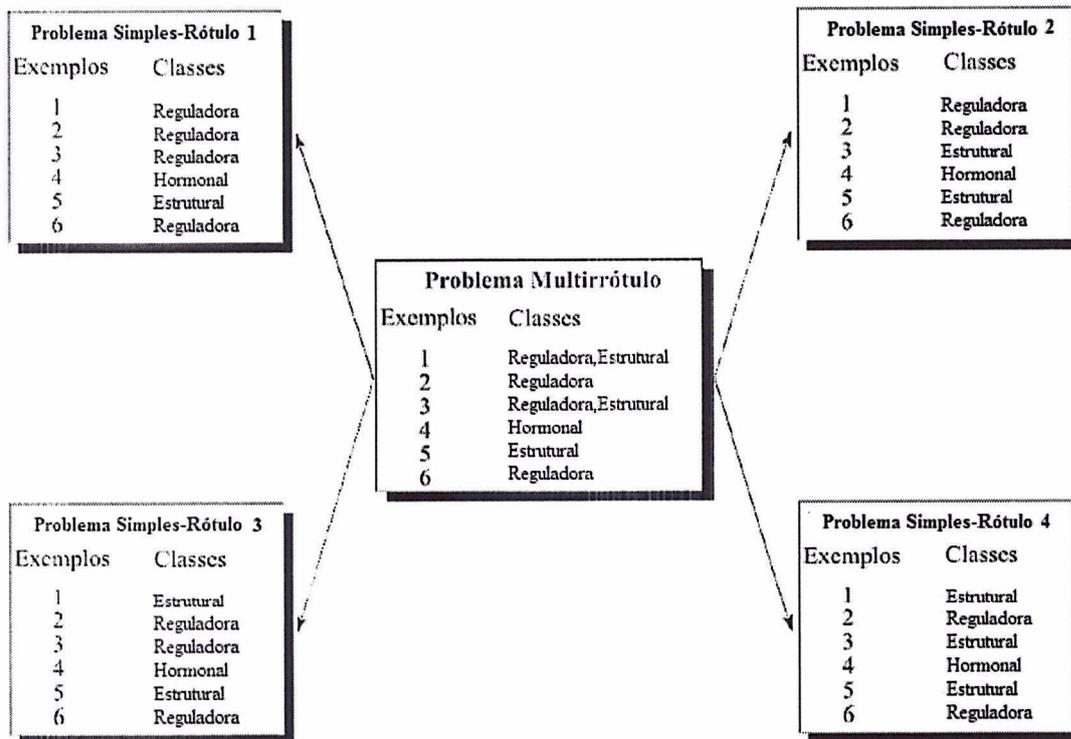


Figura 2.8: Transformação por decomposição de rótulos - método multiplicativo

2.2.1.2 Abordagens Dependentes de Algoritmo

A abordagem dependente de algoritmo modifica internamente o algoritmo dos classificadores tradicionais (único rótulo) para que possam ser utilizados em problemas multirrótulos.

No trabalho de [51] é apresentado um estudo utilizando árvores de decisão para classificação hierárquica multirrótulo para analisar informações de *S. cerevisiae* e tentar prever novas funções gênicas. Para analisar esses dados foram desenvolvidas estratégias de reamostragem e modifi-

cações no algoritmo C4.5 [34]. Foi feita uma alteração na formula da entropia, utilizando a soma das entropias de todos os rótulos.

Em [52] são propostas duas versões de um Sistema Imunológico Artificial (AIS), que é um paradigma de inteligência computacional relativamente recente para predizer funções de proteínas descritas na ontologia *Gene Ontology* (GO). A abordagem proposta chamada MHCAIS (*Multi-label Hierarchical Classification with an Artificial Immune System*) é um algoritmo de classificação adaptado para problema multirrótulo e hierárquico. A primeira versão do MHCAIS constrói um classificador global para predizer todos os rótulos, enquanto a segunda versão constrói um classificador local para predizer cada rótulo. Em ambas as versões o classificador é expresso com um conjunto de regras *IF-THEN* que tem a vantagem de representar o conhecimento de forma compreensível para usuários biólogos.

No trabalho de [53] para tratar problemas multirrótulo é proposta uma extensão do AD-Trees (*Alternating Decision Trees*) chamado ADTBoost.MH que combina os algoritmos ADT-boost [54] e do AdaBoost.MH [55]. O AdaBoost.MH é a versão para multirrótulo do AdaBoost [56, 57, 58].

No trabalho de [59] ferramenta MuLAM foi desenvolvida baseada biblioteca Weka [60], contendo vários métodos de modificações de algoritmos, como o ML-kNN [61] (*Multi-Label k-Nearest Neighbours*), BPMLL [62] (*Back-Propagation Multi-Label Learning*), entre vários outros. O algoritmo ML-KNN é baseado no algoritmo *K-NN*: para cada exemplo, os rótulos que são associados com *k* exemplos vizinhos são recuperados e é realizada uma contagem dos vizinhos associados a cada rótulo; então o principio *maximum posteriori* é utilizado para definir os rótulos de um novo exemplo. O algoritmo BPMLL é uma adaptação do popular algoritmo *back-propagation* para aprendizado multirrótulo, sendo a principal modificação desse algoritmo é a introdução de uma nova função de erro que considera múltiplos rótulos.

Em [63] uma ferramenta chamada Clus foi desenvolvida usando conceitos de *Predictive Clustering Trees* (PCT), no qual árvores de decisão são construídas onde cada nó corresponde a um grupo de exemplos do conjunto de exemplos. O PCT é uma abordagem de clusterização que adapta a indução de árvores de decisão para a clusterização. Esse procedimento usado para construção da PCT é similar a outros algoritmos de indução de árvores de decisão, como C4.5 [34] ou CART [64].

No trabalho de [65], Clus-HMC refere-se ao uso do Clus como um sistema de classificação multirrótulo hierárquico, que aprende uma árvore para classificar todos os rótulos usando a distância euclidiana ponderada e Clus-SC gera uma árvore de decisão para cada rótulo.

Na linha de pesquisa de aprendizado multirrótulo utilizando abordagem dependente do algoritmo, pode envolver a orientação de uma aluna de mestrado:

- Erica Akemi Tanaka que publicou seus resultados iniciais no *Workshop* de Informática Médica, evento do Congresso da SBC [66] e os resultados finais no *Journal of Biomedical Informatics* [67].

2.2.2 Ensembles

O viés (*bias*) e a variância podem ser vistos como componentes do erro de um classificador [68, 69, 70, 71]. Assim, métodos que reduzam o *bias* ou a variância, ou ambos, constituem uma área de interesse em AM. Um desses métodos, a construção de *ensembles*, consiste em induzir vários classificadores e então combiná-los em um único classificador final, geralmente utilizando algum mecanismo de votação [72]. Em geral, *ensembles* mostraram ser eficazes quando aplicados a indutores que são, de certa forma, instáveis, tais como indução de regras, árvores de decisão ou redes neurais [73].

Em termos formais, um *ensemble* h^* consiste em um conjunto de L classificadores individuais $\{h_1, h_2, \dots, h_L\}$, cujas previsões são combinadas para determinar o rótulo de um novo exemplo. Na sua forma mais simples, para um problema de k classes $\{C_1, C_2, \dots, C_k\}$, a combinação é efetuada utilizando o voto majoritário, dado por (2.1), na qual o operador $\|E\|$ retorna 1 se a expressão E for verdadeira e zero caso contrário. No voto majoritário, a classe predita com maior frequência por todos os classificadores $\{h_1, h_2, \dots, h_L\}$ é a classe predita pelo classificador final h^* . Para problemas de regressão, h^* é, normalmente, a média dos valores obtidos por cada hipótese individual, dada por (2.2).

$$h^*(x) = \arg \max_{c \in \{C_1, C_2, \dots, C_k\}} \sum_{l=1}^L \|h_l(x) = c\| \quad (2.1)$$

$$h^*(x) = \frac{1}{L} \sum_{l=1}^L h_l(x) \quad (2.2)$$

Frequentemente, um *ensemble* é mais preciso que qualquer uma das hipóteses individuais contida nele, isto é, o uso de múltiplas hipóteses (para classificação ou regressão) mostra uma melhoria na precisão ao rotular exemplos que não se encontram no conjunto de treinamento.

Um fato interessante é que um *ensemble* pode ser mais preciso que cada um dos classi-

ficadores que o compõe somente se cada classificador individual discorda de um outro. Uma explicação simples para esse fato é dada pelo seguinte exemplo. Suponha que um *ensemble* seja composto por três classificadores $\{h_1, h_2, h_3\}$ e considere um novo exemplo x a ser rotulado. Se os três classificadores são idênticos, então quando $h_1(x)$ erra, $h_2(x)$ e $h_3(x)$ também erram. Todavia, se os erros dos três classificadores não estão correlacionados, então quando $h_1(x)$ está errado, $h_2(x)$ e $h_3(x)$ podem estar certos. Se isso ocorrer, por meio do voto majoritário, o novo exemplo x será corretamente classificado pelo *ensemble*².

Existem algumas explicações prováveis para o fato que não é possível encontrar um classificador único que tenha um desempenho tão bom quanto um *ensemble* [72]:

- (i) os dados de treinamento podem não conter informação suficiente para a escolha da melhor e única hipótese h a partir do espaço completo de hipóteses H . A maioria dos algoritmos de AM considera espaços de hipóteses muito grandes e, dessa forma, mesmo eliminando hipóteses que classificam incorretamente os exemplos de treinamento, ainda restam muitas outras hipóteses. Essas hipóteses restantes aparentam ser igualmente precisas com relação aos dados de treinamento. A combinação desse conjunto de hipóteses restante em *ensembles*, normalmente, permite que diferentes regiões do espaço de exemplos possam ser cobertas;
- (ii) os algoritmos de aprendizado podem não ser capazes de resolver um problema difícil, que é exposto a eles. Por exemplo, o problema de encontrar a menor árvore de decisão, que seja consistente com os exemplos de treinamento, é um problema *NP*. Na prática, os algoritmos de indução de árvores de decisão empregam heurísticas para guiar o processo de busca por árvores menores. De forma similar, encontrar os pesos da menor rede neural, consistente com os exemplos de treinamento, é também um problema *NP*. Assim, os algoritmos de redes neurais empregam métodos de busca local, tais como gradiente descendente, para encontrar pesos ótimos locais. Uma consequência dessa imperfeição na busca é que — mesmo se a combinação dos exemplos de treinamento, com o conhecimento prévio do domínio, determinem a melhor e única hipótese — pode ser impossível encontrá-la. Portanto, *ensembles* podem ser vistos como uma forma de compensar a imperfeição de busca dos indutores e
- (iii) o espaço de hipóteses H pode não conter a função objetivo f . Ao invés disso, H pode incluir várias aproximações igualmente boas para f . Pela combinação dessas aproximações é possível representar classificadores que estejam fora de H . Uma forma de entender isso

²Se os erros de L classificadores forem independentes e menores que $1/2$ então a probabilidade que o voto majoritário erre é dada pela área sob a distribuição binomial na qual $L/2$ classificadores estão errados.

consiste em visualizar as fronteiras de regiões construídas pelos algoritmos de Aprendizado de Máquina. Uma região é uma superfície, tal que os exemplos que se localizam dentro são rotulados com uma classe diferente das classes dos exemplos que se localizam do lado de fora da região. Por exemplo, as regiões construídas por árvores de decisão univariadas são hiperplanos paralelos aos eixos. Se a fronteira entre duas classes for um hiperplano oblíquo aos eixos, os indutores de árvores de decisão devem aproximar o hiperplano oblíquo por uma série de hiperplanos paralelos aos eixos (formando uma ‘escada’). Essas aproximações em escada são equivalentes a uma árvore de decisão muito complexa, como mostrado no exemplo da Figura 2.9 na próxima página. Entretanto, para induzir essa árvore tão extensa, no espaço de hipóteses H , seriam necessários muitos exemplos de treinamento, que nem sempre estão disponíveis. Dessa forma, o uso de *ensembles* pode propiciar uma forma de ultrapassar as deficiências de representação do espaço de hipóteses.

Existe uma diversidade considerável de métodos usados para compor *ensembles*, alguns dos quais efetuam a manipulação dos atributos (por exemplo, cada classificador individual tem acesso a um subconjunto dos atributos originais); a manipulação da classe (por exemplo, problemas com muitas classes podem ser vistos como vários problemas com classes binárias); inserção de um componente aleatório (por exemplo, os pesos aleatórios definidos inicialmente em uma rede neural ou a escolha entre, diga-se, 20 dos melhores testes para um nó em uma árvore de decisão); uso de amostragem, entre outros [74, 75, 72].

A seguir descrevo três métodos usados para compor *ensembles* referentes a trabalhos que orientei.

2.2.2.1 Windowing

Windowing pode ser visto como um precursor dos *ensembles*, já que ele não combina vários classificadores em um classificador final mas obtém um classificador final por meio de iterações sucessivas, descartando os classificadores anteriores. *Windowing* seleciona um subconjunto de exemplos (uma *window*) do conjunto de treinamento e induz uma hipótese h_1 a partir dele [34]. A hipótese h_1 é então usada para classificar os exemplos remanescentes do conjunto de treinamento, isto é, aqueles exemplos que não foram incluídos na *window*. Se existem exemplos classificados incorretamente, uma seleção deles é adicionada à *window* inicial e uma segunda hipótese h_2 é construída, a partir da *window* aumentada. Este ciclo é repetido até que a hipótese, construída a partir da *window* atual, classifique corretamente todos os exemplos de treinamento fora da *window* ou o número de ciclos exceda um valor L pré-definido.

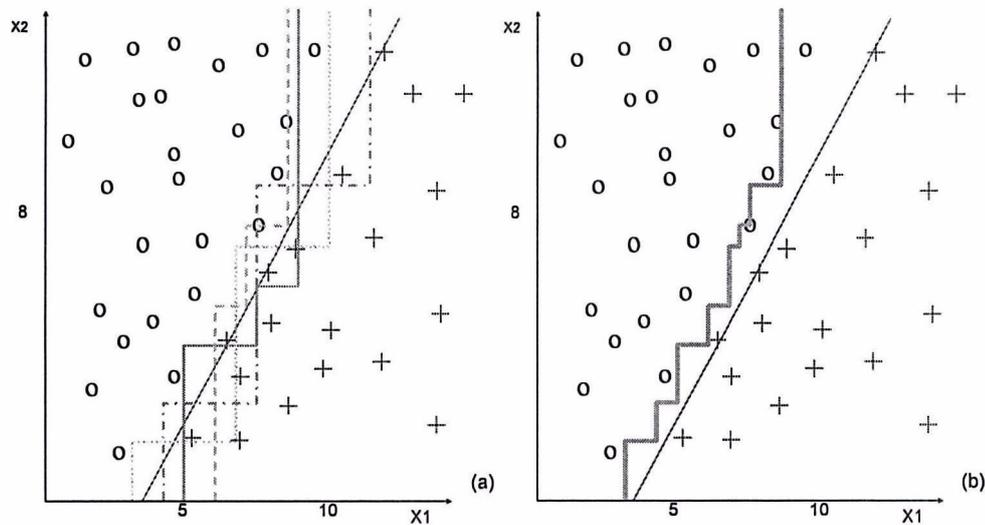


Figura 2.9: A região real de separação de classes, indicada pela linha diagonal, é aproximada de formas diferentes por quatro árvores de decisão em (a); a combinação das quatro árvores, em um *ensemble*, fornece uma região que se aproxima mais da diagonal em (b)

2.2.2.2 Stacking

No método *Stacking*, as descrições dos exemplos de treinamento são estendidas para incluir os resultados da classificação de exemplos com uma seleção inicial de classificadores. As novas descrições são então analisadas para compor novos classificadores [76].

Mais precisamente, suponha L diferentes algoritmos A_1, A_2, \dots, A_L e um conjunto de n exemplos de nível-0 $\{(x_1, y_2), (x_2, y_2), \dots, (x_n, y_n)\}$. Cada algoritmo é aplicado ao conjunto de treinamento nível-0, induzindo os classificadores $\{h_1, h_2, \dots, h_L\}$. Após induzido, cada classificador é usado para rotular os exemplos. Isso implica que, para cada exemplo x_i , uma nova tupla é formada, composta pela classe obtida por cada uma das hipóteses e a classe verdadeira do exemplo, isto é, $(h_1(x_i), h_2(x_i), \dots, h_L(x_i), y_i)$. Essas tuplas constituem o conjunto de treinamento de nível-1, cujos atributos são as classes preditas por cada um dos L classificadores. Com isso, é possível aplicar outro algoritmo de aprendizado (meta algoritmo) aos exemplos de nível-1 para aprender h^* , conforme esquema mostrado na Figura 2.10 na página seguinte.

2.2.2.3 Random Forests

Como já descrito, considere um conjunto de treinamento T com m atributos e n exemplos, seja T_k uma amostra *bootstrap* [77] do conjunto de treinamento a partir de T com reposição, contendo n exemplos e usando a atributos aleatórios ($a \leq m$) em cada nó das árvores.

Uma *Random Forest* é definida formalmente como um classificador composto por uma

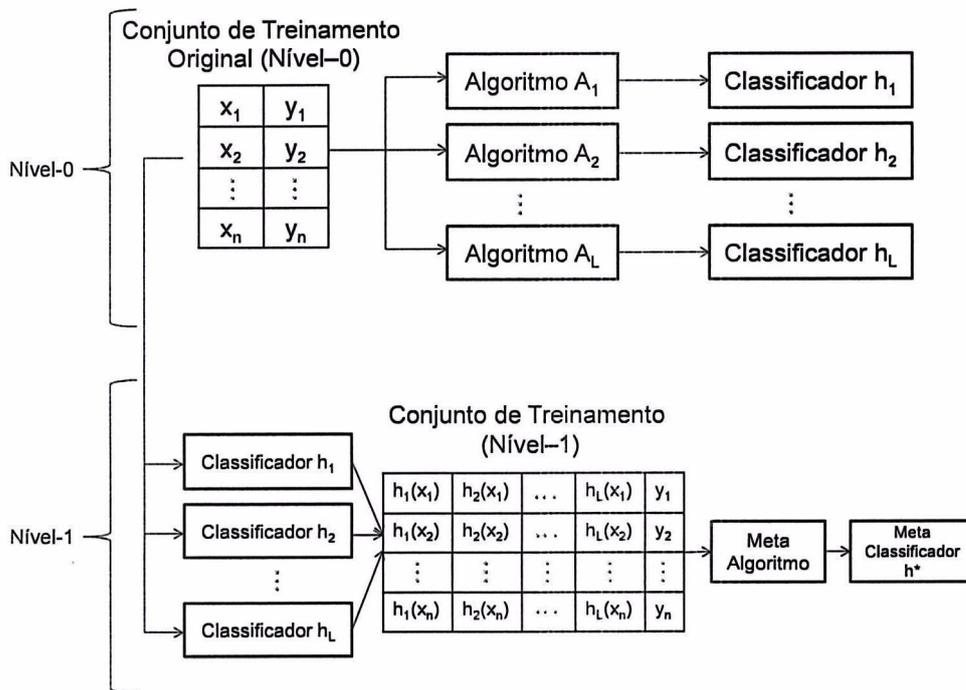


Figura 2.10: Stacking

coleção de árvores $\{h_k(x)\}, k = 1, 2, \dots, L$, onde T_k são amostras aleatórias independentes e identicamente distribuídas e cada árvore vota na classe mais popular para a entrada x [78, 79, 80].

Portanto, *Random Forests* aplicam o mesmo método que o *bagging* [81] para produzir amostras aleatórias de conjuntos de treinamento (amostras *bootstraps*) para cada árvore. Breiman [78] justifica o uso do método *bagging* em *Random Forests* por duas razões: o uso do *bagging* parece melhorar o desempenho quando atributos aleatórios são usados; *bagging* pode ser usado para fornecer estimativas contínuas do erro de generalização do conjunto combinado de árvores, assim como estimativas para força e correlação, usando o estimador *out-of-bag*.

O erro de classificação da floresta depende da força das árvores individuais da floresta e da correlação entre quaisquer duas árvores na floresta [78, 82, 83, 84], a saber:

- Correlação entre as árvores da floresta: duas medidas de aleatoriedade (uso do *bagging* e seleção aleatória de atributos) fazem com que as árvores sejam diferentes e, portanto, diminui a correlação entre elas. A baixa correlação tende a diminuir a taxa do erro de classificação.
- Força da árvore individual na floresta: pode ser interpretada como uma medida de desempenho para cada árvore. Uma árvore com uma taxa de erro baixa é um classificador forte.

Assim, aumentando a força das árvores individuais, reduz-se a taxa de erro da floresta.

Nesta linha de trabalho utilizando *ensembles* pude envolver a orientação de dois alunos de mestrado e uma aluna de iniciação científica:

- Pedro Santoro Perez, inicialmente bolsista CAPES e posteriormente bolsista FAPESP que, no mestrado, estendeu o algoritmo de *Windowing* tendo publicado seus resultados iniciais no *ECML PKDD Workshop on Knowledge Discovery in Health Care* [85] e os resultados finais da sua pesquisa no periódico *Knowledge-Based Systems* [86];
- Thais Mayumi Oshiro que investigou no mestrado os aspectos de *Random Forests* publicando seus resultados na *International Conference on Intelligent Data Engineering and Automated Learning* [87] e na *International Conference on Machine Learning and Data Mining in Pattern Recognition* [88] e
- Maria Izabela Ruz Caffé que em sua iniciação científica investigou o comportamento de *Stacking* publicando os resultados no *Workshop de Informática Médica do Congresso da Sociedade Brasileira de Computação*, cujo artigo foi premiado como melhor artigo completo do evento [89]. A aluna foi convidada para publicar uma versão estendida do artigo no periódico *Journal of Health Informatics* [90].

2.3 Pós-Processamento

Ao final da etapa de mineração, tem início o pós-processamento. Nesta etapa, ocorre a *interpretação dos resultados*, que consiste na avaliação dos padrões descobertos, visualização dos padrões extraídos, remoção de padrões irrelevantes ou redundantes e tradução de padrões úteis em termos inteligíveis pelos usuários. A seguir, tem-se o *uso do conhecimento extraído*, que consiste na incorporação do conhecimento no desempenho do sistema, tomando ações baseadas no conhecimento ou simplesmente documentando e relatando para as partes interessadas o conhecimento obtido, bem como remoção de conflitos potenciais com conhecimento previamente tido como correto (ou extraído).

A seguir enumero algumas das contribuições em termos de publicações que se encaixam na etapa de pós-processamento principalmente devido ao fato de serem pesquisas com foco na visualização e interpretação dos modelos induzidos por especialistas humanos. Como já mencionado no início desta seção, apesar dessas contribuições estarem posicionadas nesta etapa final, essa separação em relação às demais etapas do processo de EC não é isolada, considerando que em vários dos trabalhos, mais de uma etapa foi utilizada.

- A pesquisa de iniciação científica da aluna Patrícia Miranda Fugimoto fez uma comparação entre o modelo de regressão logística utilizada no sistema padrão de pontuação de índice de gravidade, o *Trauma and Injury Severity Score* (TRISS) muito utilizado por médicos em todo o mundo para calcular a probabilidade de sobrevivência de pacientes traumatizados com modelos simbólicos, ou seja, árvores de decisão que permitem a interpretação e validação por especialistas humanos, utilizando dados do Hospital das Clínicas de Ribeirão Preto. A árvore permitiu classificar e prever a condição de alta dos pacientes traumatizados, além de permitir interpretação e validação por especialistas humanos sobre o modelo induzido. Os resultados deste trabalho foram publicados no Congresso da Academia Trinacional de Ciências [91];
- A aluna de iniciação científica Luciana Domene Furlan Sales, pesquisou a partir de um modelo utilizado em estudos de trauma no mundo, TRISS (*TRAuma and Injury Severity Score*), que avalia a qualidade do desempenho de um hospital no cuidado com o trauma, calculando a probabilidade de sobrevivência de um paciente traumatizado. A pesquisa desta aluna avaliou a utilidade e as limitações que a metodologia TRISS apresenta no Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto. Para isso, foram utilizados dados de pacientes traumatizados e métricas como sensibilidade e especificidade [92], dentre outras, para analisar se o limiar de 50% da probabilidade de sobrevivência classificava corretamente os casos de óbitos e não-óbitos. As análises efetuadas constataram que a metodologia TRISS, nos dados analisados, de 2005 a 2007, era ineficiente para identificar os óbitos devidamente, apresentando uma alta taxa de falso-negativos. Os resultados foram publicados no Congresso Brasileiro de Informática em Saúde [93];
- O trabalho de iniciação científica da aluna Erica Akemi Tanaka construiu uma arquitetura de *software* composta por três camadas, sendo que uma delas, a camada de classificação era composta por quatro módulos de classificação, voltados ao suporte à decisão. A arquitetura implementada foi testada em uma clínica de fertilização de forma que os especialistas médicos podiam tanto indicar um tratamento como prever o sucesso do mesmo no processo de fertilização. A arquitetura foi publicada no Congresso Brasileiro de Informática em Saúde [94];
- Um trabalho inicial do aluno de mestrado Pedro Santoro Perez consistiu em desenvolver uma arquitetura de *software* composta por quatro camadas, sendo uma delas a camada de algoritmos de classificação. Um dos algoritmos nesta camada foi o algoritmo J48 (implementação da biblioteca Weka [60] do algoritmo C4.5) fornecendo modelos simbólicos aos especialistas do projeto INCT Adapta. Os resultados deste trabalho fo-

ram publicados no *International Conference on Computational Biosciences and Bio-Informatics* [95] e

- A pesquisa de iniciação científica do aluno Oscar Picchi Netto (bolsista PIBIC) envolveu mineração de dados textuais. O objetivo do trabalho consistiu em propor uma metodologia de estruturação para a base de laudos radiológicos do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto visando, construir de forma (semi-)automática um laudo estruturado que permita a extração de conhecimento para facilitar as atividades ligadas ao ensino e pesquisa deste hospital escola. A metodologia proposta foi avaliada em três bases contendo 5000 laudos cada uma e os resultados comparados com uma ontologia padrão na área médica. Os resultados desta pesquisa foram publicadas no *Workshop de Informática Médica do Congresso da SBC* [96].

Extração de Conhecimento é uma área ainda recente de pesquisa em Inteligência Artificial. EC é uma área interdisciplinar resultante da intersecção de pesquisas em várias outras — base de dados, estatística, Aprendizado de Máquina, lógica *fuzzy*, aquisição de conhecimento para sistemas baseados em conhecimento, visualização de dados, mineração de textos, recuperação de informação e computação de alto desempenho, a qual incorpora teorias, algoritmos e métodos de todas essas áreas.

Na área de EC começam a surgir ferramentas que suportem todo o processo, porém nem sempre são simples de se utilizar. Um dos tópicos que desafia pesquisadores é o uso de técnicas de Aprendizado de Máquina em grande volume de dados. A maioria das técnicas de AM aplicam-se a pequenos conjuntos de exemplos que podem caber na memória principal. Todavia, em aplicações de EC essas técnicas devem ser empregadas com grandes conjuntos de exemplos.

Várias abordagens têm sido utilizadas para ultrapassar essa limitação. Entre elas tem-se o uso de paralelismo, desenvolvimento de novas técnicas orientadas a bases de dados, redução da dimensão dos dados e amostragem dos dados, entre outras. Todavia, muito esforço será necessário, antes que EC possa ser aplicado com sucesso em grandes conjuntos de exemplos.

2.4 Considerações Finais

O trabalho descrito nesta seção, trouxe, a partir de 2003, contribuições que incluem:

- 2 publicações em periódicos internacionais [86, 67];

- 1 publicação em periódico nacional [90];
- 5 publicações em congressos internacionais [88, 87, 85, 95, 91];
- 8 publicações em congressos nacionais [45, 66, 89, 96, 93, 94, 22, 19];
- 2 resumos publicados em anais de congressos nacionais [18, 17];
- 4 mestrados concluídos (1 deles com recursos FAPESP e 1 com recursos CAPES) e 2 em andamento e
- 7 projetos de iniciação científica concluídos, com recursos FAPESP (1) e CNPq (2).

3 *Trabalhos Adicionais & em Colaboração*

If you want to show that one algorithm is *always* more accurate, then forget it: this simply cannot be proven.

—Steven L. Salzberg

Como docente e pesquisador na USP junto ao curso de Informática Biomédica a partir de 2003 e como consequência do programa de doutorado publiquei os seguintes trabalhos:

- No trabalho [14] detalhei o *framework* XRULER, já descrito na Seção 1 na página 1 e
- O material dos dois capítulos de livro [92, 97] é resultante da compilação e extensão do material da minha tese de doutorado [2] e de dois relatórios técnicos [74, 98];

Meu vínculo a ambos projetos INCT Adapta e NAP Aprendizado de Máquina em Análise de Dados (NAP AMAD) colocou-me em uma posição privilegiada para poder compartilhar o trabalho neles realizado. Assim, os estudos realizados em relação ao NAP AMAD puderam também ser repassados aos pesquisadores do projeto INCT Adapta. Minhas responsabilidades junto ao Projeto INCT Adapta, do qual era coordenador institucional do laboratório de mineração de dados em bioinformática, eram maiores e as contribuições de minha pesquisa são mais visíveis em seu contexto. A interação entre meus colegas do Departamento de Computação de Matemática, da Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto e da Faculdade de Medicina de Ribeirão Preto, permitiu-me contribuir com alguns trabalhos adicionais, além daqueles já citados na seção anterior:

- Na pesquisa de [99], para associar crianças com problemas de desenvolvimento e a literatura científica e alertar profissionais da saúde sobre os problemas de saúde foi desenvolvido o CISS (*Chronic Illness Surveillance System*). O CISS tem como objetivo prover aos profissionais de saúde informações a cerca de exposições a fatores ambientais no princípio da vida, os quais podem induzir a modificações no desenvolvimento humano,

podendo gerar impacto na saúde na vida adulta e causar risco de doença. Com o objetivo de divulgar informações preventivas, o serviço CISS associa fatores de risco genéticos e epigenéticos em termos de doenças crônicas apresentados em artigos científicos com os registros clínicos de pacientes. Ao estar ciente dessa abordagem, os profissionais da saúde podem então criar uma rotina clínica com as famílias buscando melhores condições de crescimento. O CISS utiliza termos ontológicos, em vez de palavras-chave para compor uma coleção mais precisa de artigos científicos em epigenética a serem relacionados com casos clínicos. Pode contribuir parcialmente na definição dos experimentos e análise dos resultados obtidos para validação do sistema desenvolvido;

- Contribuí no trabalho de [100], juntamente com o aluno de mestrado Pedro Santoro Perez, para extrair uma árvore de decisão — utilizando *Windowing* e demais as etapas do processo de EC — que, a partir de propriedades físico-químicas de peptídeos antimicrobianos, pudesse ajudar os especialistas no domínio a identificarem e desenvolverem novos compostos com atividade antimicrobiana;
- No artigo [101] pesquisei propriedades de três critérios de escolha do melhor atributo (*best-split*) para compor os nós em árvores de decisão, trazendo contribuições com relação ao número de classes e estabilidade das árvores em dados de alta dimensão, tipicamente encontrados em domínios biomédicos;
- Na pesquisa de [102] foi desenvolvida uma arquitetura de *software* para auxílio à tomada de decisão na classificação de prontuários eletrônicos de pacientes, que conta com três camadas, sendo uma delas a camada de classificação que utiliza algoritmos de diversos paradigmas de Aprendizado de Máquina, na qual se encontra minha contribuição neste trabalho;
- No trabalho de [103] foi desenvolvido o Automatic-SL, um sistema computacional para definição automática ou semi-automática em casos especiais, de graus de vigilância à saúde do paciente, a partir da manipulação e análise de informações provenientes de prontuários de atendimento de saúde. O Automatic-SL apresenta uma arquitetura organizada em três camadas, sendo uma delas a camada de classificação (na qual se encaixa minha contribuição), uma plataforma de software suportada por seis módulos (cinco módulos de classificação e um módulo de análise linguística responsável pelo pré-processamento de linguagem);
- No trabalho resumido de [104] pude contribuir com meus conhecimentos sobre o processo de Extração de Conhecimento no domínio da modelagem de crises epiléticas e

- No trabalho de [105] pude colaborar na aplicação de processo de busca de parâmetros de cinética enzimática. O *software* produzido neste trabalho tem aplicação tanto educacional como para pesquisa na área de cinética enzimática.

3.1 Considerações Finais

Os trabalhos mencionados nesta seção trouxeram contribuições que incluem:

- 6 publicações em periódicos internacionais [14, 99, 100, 101, 102, 105];
- 1 publicações em congressos internacionais [103];
- 2 capítulos de livro nacional [92, 97];
- 1 resumo publicado em anais de congresso internacional [104]
- 2 projetos temáticos de pesquisa; INCT Adapta, do qual fui coordenador local, captando recursos da ordem de R\$ 150mil para o DCM; NAP AMAD, no qual fui colaborador, captou US\$ 500mil;
- colaboração direta com 1 professora pesquisadora do DCM, 2 professores pesquisadores do ICMC-USP; 1 pesquisador da empresa Dow AgroSciences (Seeds, Traits & Oils) e 1 pesquisadora da Universidade Federal do Amazonas.

4 Conclusão

No, I'm not interested in developing a powerful brain. All I'm after is just a mediocre brain.

—Alan Turing

4.1 Impacto das Contribuições

A área em que tenho atuado, Extração de Conhecimento e Aprendizado de Máquina, é bastante recente — mesmo considerada dentro da também recente área de Ciência da Computação. O trabalho realizado desde meu doutorado teve impacto na consolidação da área de pesquisa tanto a nível local (no *campus* da USP de Ribeirão Preto) como em nível nacional. Minha produção científica desde 2003, ano em que fui contratado como docente junto à USP, totaliza 28 trabalhos, sendo que 23 (82%) foram trabalhos que desenvolvi ou orientei e 5 (18%) foram trabalhos em com os quais contribuí, a saber:

- 23 trabalhos que desenvolvi (como primeiro autor) ou orientei (como último autor):
 - 4 publicações em periódicos internacionais [86, 67, 101, 14];
 - 1 publicação em periódico nacional [90];
 - 5 publicações em congressos internacionais [88, 87, 85, 95, 91];
 - 8 publicações em congressos nacionais [45, 66, 89, 96, 93, 94, 22, 19];
 - 2 capítulos de livro nacional [92, 97];
 - 1 resumo publicado em anais de congresso internacional [104] e
 - 2 resumos publicados em anais de congressos nacionais [18, 17].
- 5 trabalhos com os quais contribuí em minha área de pesquisa (como autor intermediário, sem ser nem o primeiro, nem o último autor):
 - 4 publicações em periódicos internacionais [99, 100, 102, 105] e

- 1 publicações em congressos internacionais [103].

Como é possível notar, nos últimos anos meu foco tem sido a publicação em periódicos internacionais com sistema de arbitragem, com fator de impacto $JCR \geq 1$ e periódicos de índice restrito (estratos Qualis A1, A2 ou B1) em Ciência da Computação.

Em termos do Departamento de Computação e Matemática, destaca-se o seguinte:

- Dentre as colaborações, o maior destaque se dá à colaboração com a Professora Alessandra Alaniz Macedo pelo permanente e recíproco trabalho em conjunto dos grupos orientados pelos dois professores. A pesquisadora apoia fortemente meu grupo na modelagem e na preparação dos artefatos de *software*. Por outro lado, tenho apoiado o desenvolvimento e o uso de mecanismos de Inteligência Artificial e Aprendizado de Máquina, além dos métodos envolvidos na experimentação das pesquisas. Essa colaboração mútua têm gerado excelentes resultados publicados ao longo dos últimos anos [86, 67, 99, 102, 95, 103, 96, 22] e
- O Laboratório de Informática em Saúde (LIS) localizado na Sala 621 do Bloco Alan Turing do DCM foi criado a partir do apoio dos dois projetos FAPESP, coordenados pela pesquisadora e foi renovado por verba do INCT Adapta, coordenado pelo mim localmente e também contando com participação da pesquisadora. O laboratório LIS adquiriu um *cluster* de alta performance voltado para pesquisa exploratória e desenvolvimento de componentes de Recuperação de Informação. Além do cluster, o LIS conta com outros equipamentos como *notebooks*, computadores *desktops*, *tablets*, *scanner* e impressoras. Em torno do LIS que vem se consolidando um grupo de orientadores e alunos de graduação e mestrado.

Além disso, o trabalho no DCM em particular só tem sido possível dada a parceria também com o Professor Renato Tinós e, mais recentemente, com o Professor Zhao Liang e ao interesse dos alunos de graduação e pós-graduação. Em nível nacional, minha atuação nos Projetos INCT Adapta e NAP AMAD possibilitou a integração com um grande número de pesquisadores de instituições já estabelecidas e também de instituições emergentes.

A cooperação que obtive nestes dois projetos veio complementar minha atuação com benefícios que foram refletidos não apenas em minhas publicações, mas também infraestrutura ora disponibilizada no DCM. Em termos de formação de pessoal, essa colaboração levou à orientação de 4 mestrados concluídos, 2 em andamento e 7 iniciações científicas concluídas.

A criação de um grupo de pesquisadores no LIS e o início do programa de pós-graduação em Computação Aplicada do DCM bem como a consolidação da área no país foram fatores relevantes para minhas atividades de pesquisa em particular, uma vez que cada vez mais tem sido possível atrair bons alunos e colaborar com colegas orientadores e seus respectivos alunos, atividades essas catalisadoras de bons resultados de pesquisa.

4.2 Trabalhos Futuros

A curto e médio prazos, o trabalho a ser realizado se realizará no contexto de Extração de Conhecimento. Sucintamente, as atividades incluem:

- Com Alessandra Alaniz Macedo investigar mecanismos para a recuperação de informação textual para estruturação usando ontologias, hierarquias, redes semânticas, ou outras representações [96];
- Com Renato Tinós investigar formas de extração de conhecimento a partir de redes neurais artificiais ou envolvendo problemas de otimização [19];
- Com Zhao Liang investigar formas de representação simbólicas de sistemas complexos;
- Com Denise Gazotto Dezembro, que está iniciando o mestrado em Computação Aplicada, explorar a classificação de textos de pacientes que requerem atenção contínua utilizando ontologias;
- Com Lucas Fióro e Silva, que também está iniciando o mestrado em Computação Aplicada, explorar *Random Subspaces* em algoritmos de Aprendizado de Máquina;
- Com Sérgio Ricardo Nozawa, investigar problemas adicionais em bioquímica para aplicação de mineração de dados [100, 86];

4.3 Considerações Finais

A pesquisa realizada de modo cooperativo com alunos e outros pesquisadores é uma constante nas atividades que tenho desenvolvido desde meu doutorado. É com satisfação que observo o trabalho de pesquisa tem colaborado diretamente com minhas atividades de docência tanto na graduação como na pós-graduação, dentro do DCM e na colaboração em comitês científicos de simpósios nacionais. Meus planos para pesquisa a curto e a médio prazos incluem estender esse tipo de colaboração os quais, espero, tragam resultados similares.

APÊNDICE A – Mapeamento de Referências

Neste apêndice é disponibilizado o mapeamento entre as referências no Memorial e na Documentação com as referências deste documento, por meio da Tabela A.1.

Tabela A.1: Mapeamento entre referências

Referência no Memorial e Documentação	Referência neste documento
[1]	[86]
[2]	[67]
[3]	[99]
[4]	[100]
[5]	[101]
[6]	[90]
[7]	[102]
[8]	[105]
[9]	[14]
[10]	[106]
[11]	[107]
[12]	[88]
[13]	[87]
[14]	[85]
[15]	[95]
[16]	[91]
[17]	[103]
[18]	[9]
[19]	[10]
[20]	[11]
[21]	[8]
[22]	[5]
[23]	[3]
[24]	[45]
[25]	[66]
[26]	[89]
[27]	[96]
[28]	[93]
[29]	[94]
[30]	[22]
[31]	[19]
[32]	[108]
[33]	[4]
[34]	[109]
[35]	[92]
[36]	[97]

Referências

- [1] H. C. Bumpus, “Eleventh lecture. the elimination of the unfit as illustrated by the introduced sparrow, *passer domesticus*. (a fourth contribution to the study of variation.),” *Biological Lectures: Woods Hole Marine Biological Laboratory*, pp. 209–225, 1898. [Citado nas páginas xiii e xv].
- [2] J. A. Baranauskas, *Extração Automática de Conhecimento Utilizando Múltiplos Indutores*. PhD thesis, ICMC-USP, 2001. <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08102001-112806/>. [Citado nas páginas 1 e 29].
- [3] J. A. Baranauskas and M. C. Monard, “Experimental feature selection using the wrapper approach,” in *Proceedings of the International Conference on Data Mining*, (Rio de Janeiro, RJ), pp. 161–170, Sept. 1998. <http://dcm.ffclrp.usp.br/~augusto/publications/1998-icdm.pdf>. [Citado nas páginas 1 e 37].
- [4] J. A. Baranauskas, M. C. Monard, and P. S. Horst, “Evaluation of feature selection by wrapping around the CN2 inducer,” in *Encontro Nacional de Inteligência Artificial (ENIA/SBC)*, (Rio de Janeiro, RJ), pp. 315–326, jul 1999. <http://dcm.ffclrp.usp.br/~augusto/publications/1999-enia.pdf>. [Citado nas páginas 1 e 37].
- [5] J. A. Baranauskas, M. C. Monard, and P. S. Horst, “Evaluation of CN2 induced rules using feature selection,” in *Proceedings of the Argentine Symposium on Artificial Intelligence (ASAI/JAIIO/SADIO)*, (Buenos Aires, Argentine), pp. 141–154, sept 1999. <http://dcm.ffclrp.usp.br/~augusto/publications/1999-asai.pdf>. [Citado nas páginas 1 e 37].
- [6] J. A. Baranauskas and M. C. Monard, “The \mathcal{MLC}^+ wrapper for feature subset selection using decision tree, production rule, instance based and statistical inducers: Some experimental results,” Tech. Rep. 87, ICMC-USP, jul 1999. http://dcm.ffclrp.usp.br/~augusto/publications/rt_87.pdf. [Citado na página 1].
- [7] H. D. Lee, M. C. Monard, and J. A. Baranauskas, “Empirical comparison of wrapper and filter approaches for feature subset selection,” Tech. Rep. 94, ICMC-USP, oct 1999. http://dcm.ffclrp.usp.br/~augusto/publications/rt_94.pdf. [Citado na página 1].
- [8] H. D. Lee, M. C. Monard, and J. A. Baranauskas, “A practical approach for knowledge-driven constructive induction,” in *Proceedings of the Argentine Symposium on Artificial Intelligence (ASAI/JAIIO/SADIO)*, pp. 71–85, Sept 2000. <http://dcm.ffclrp.usp.br/~augusto/publications/2000-asai.pdf>. [Citado nas páginas 2 e 37].
- [9] J. A. Baranauskas, M. C. Monard, and G. E. A. P. A. Batista, “A computational environment for extracting rules from databases,” in *Proceedings of the Second International Conference on Data Mining* (N. Ebecken and C. A. Brebbia, eds.), (Cambridge,

- UK), pp. 321–330, 2000. <http://dcm.ffclrp.usp.br/~augusto/publications/2000-icdm.pdf>. [Citado nas páginas 2 e 37].
- [10] J. A. Baranauskas and M. C. Monard, “An environment for rule extraction and evaluation from databases,” in Monard and Sichman [111], pp. 187–196. <http://dcm.ffclrp.usp.br/~augusto/publications/2000-sbia-iberamia.pdf>. [Citado nas páginas 2 e 37].
- [11] J. A. Baranauskas and M. C. Monard, “An approach for extracting symbolic ensembles from databases,” (Rio de Janeiro), LMC/EPUSP, 2000. <http://dcm.ffclrp.usp.br/~augusto/publications/2000-cilamce.pdf>. [Citado nas páginas 2 e 37].
- [12] R. C. Prati, J. A. Baranauskas, and M. C. Monard, “Uma proposta de unificação da linguagem de representação de conceitos de algoritmos de aprendizado de máquina simbólicos,” Tech. Rep. 137, ICMC-USP, 2001. http://dcm.ffclrp.usp.br/~augusto/publications/rt_137.pdf. [Citado na página 2].
- [13] R. C. Prati, J. A. Baranauskas, and M. C. Monard, “Extração de informações padronizadas para a avaliação de regras induzidas por algoritmos de aprendizado de máquina simbólico,” Tech. Rep. 145, ICMC-USP, 2001. http://dcm.ffclrp.usp.br/~augusto/publications/rt_145.pdf. [Citado na página 2].
- [14] J. A. Baranauskas and M. C. Monard, “Combining symbolic classifiers from multiple inducers,” *Knowledge-Based Systems*, vol. 16, no. 3, pp. 129–136, 2003. [http://dx.doi.org/10.1016/S0950-7051\(02\)00021-7](http://dx.doi.org/10.1016/S0950-7051(02)00021-7). [Citado nas páginas 3, 29, 31, 33, e 37].
- [15] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “The KDD process for extracting useful knowledge from volumes of data,” *Communications of the ACM*, vol. 39, pp. 27–34, Nov 1996. [Citado na página 5].
- [16] S. M. Weiss and N. Indurkha, *Predictive Data Mining: A Practical Guide*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. [Citado na página 5].
- [17] R. N. Lemos and J. A. Baranauskas, “Avaliação de arredondamento de valores de atributos contínuos na indução de Árvores de decisão,” in *Simpósio Internacional de Iniciação Científica da Universidade de São Paulo - SIICUSP*, (São Paulo, SP), 2004. <http://dcm.ffclrp.usp.br/~augusto/publications/2004-Lemos-siicusp.pdf>. [Citado nas páginas 8, 28, e 33].
- [18] R. N. Lemos and J. A. Baranauskas, “Uma avaliação de dados de expressão gênica em leucemias agudas utilizando Árvores de decisão,” in *X Congresso Brasileiro de Informática em Saúde*, (Florianópolis, SC), 2006. <http://dcm.ffclrp.usp.br/~augusto/publications/2006-Lemos-CBIS.pdf>. [Citado nas páginas 8, 28, e 33].
- [19] T. M. Oshiro, S. P. M. Carvalho, A. S. Peres, R. H. A. S. R. Tinós and, and J. A. Baranauskas, “Avaliação da identificação do sexo de pessoas não identificadas por meio de radiografias faciais com o uso de métodos de aprendizado de máquina,” in *XXX Congresso da Sociedade Brasileira de Computação, X Workshop de Informática Médica, ISSN 2175-2761*, (Belo Horizonte, MG), p. 10p., 2010. <http://dcm.ffclrp.usp.br/~augusto/publications/2010-Oshiro-WIM-SBC.pdf>. [Citado nas páginas 8, 28, 33, 35, e 37].

- [20] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, pp. 245–271, Dec. 1997. [Citado nas páginas 9, 10, e 11].
- [21] P. Langley, *Elements of Machine Learning*. Morgan Kaufmann Publishers, Inc., 1996. [Citado na página 9].
- [22] O. P. Netto, S. R. Nozawa, R. A. R. Mitrowsky, A. A. Macedo, and J. A. Baranauskas, "Applying decision trees to gene expression data from dna microarrays: A leukemia case study," in *XXX Congresso da Sociedade Brasileira de Computação, X Workshop de Informática Médica, ISSN 2175-2761*, (Belo Horizonte, MG), p. 10p., 2010. <http://dcm.ffclrp.usp.br/~augusto/publications/2010-Netto-WIM-SBC.pdf>. [Citado nas páginas 9, 12, 28, 33, 34, e 37].
- [23] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Morgan Kaufmann, 2011. [Citado na página 9].
- [24] G. Ditzler, J. Morrison, Y. Lan, and G. Rosen, "Fizzy: feature subset selection for meta-genomics," *BMC Bioinformatics*, vol. 16, no. 1, p. 358, 2015. [Citado na página 10].
- [25] M. Mandal, A. Mukhopadhyay, and U. Maulik, "Prediction of protein subcellular localization by incorporating multiobjective pso-based feature subset selection into the general form of Chou's PseAAC," *Medical & Biological Engineering & Computing*, vol. 53, no. 4, pp. 331–344, 2015. [Citado na página 10].
- [26] P. Purkayastha, A. Rallapalli, N. Bhanu Murthy, A. Malapati, P. Yogeewari, and D. Sri-ram, "Effect of feature selection on kinase classification models," in *Computational Intelligence in Medical Informatics* (N. B. Muppalaneni and V. K. Gunjan, eds.), SpringerBriefs in Applied Sciences and Technology, pp. 81–86, Springer Singapore, 2015. [Citado na página 10].
- [27] S. Devaraj and S. Paulraj, "An efficient feature subset selection algorithm for classification of multidimensional dataset," *The Scientific World Journal*, no. Article ID 821798, p. 9 p., 2015. [Citado na página 10].
- [28] G. Govindan and A. S. Nair, "Sequence features and subset selection technique for the prediction of protein trafficking phenomenon in eukaryotic non membrane proteins," *International Journal of Biomedical Data Mining*, vol. 3, no. 2, pp. 1–9, 2014. [Citado na página 10].
- [29] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997. Relevance. [Citado nas páginas 10, 11, e 12].
- [30] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, pp. 389–422, Mar. 2002. [Citado na página 10].
- [31] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003. [Citado na página 10].
- [32] Özge Uncu and I. Türkşen, "A novel feature selection approach: Combining feature wrappers and filters," *Information Sciences*, vol. 177, no. 2, pp. 449–466, 2007. [Citado nas páginas 10 e 12].

- [33] D. W. Aha, *Feature Weighting for Lazy Learning Algorithms*, ch. 2. Kluwer Academic Publishers, 1998. [Citado na página 10].
- [34] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. San Francisco, CA. [Citado nas páginas 11, 19, e 22].
- [35] P. Clark and T. Niblett, “The CN₂ induction algorithm,” *Machine Learning*, vol. 3, no. 4, pp. 261–283, 1989. [Citado na página 11].
- [36] P. Clark and R. Boswell, “Rule induction with CN₂: Some recent improvements,” in *Proceedings of the 5th European Conference (EWSL 91)* (Y. Kodratoff, ed.), pp. 151–163, Springer-Verlag, 1991. [Citado na página 11].
- [37] G. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Proceedings of the Tenth International Conference on Machine Learning*, (San Francisco, CA), pp. 167–173, Morgan Kaufmann, 1994. [Citado na página 11].
- [38] T. M. Mitchell, *Machine Learning*. McGraw–Hill, 1998. [Citado na página 11].
- [39] D. W. Aha, “Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms,” *International Journal of Man-Machine Studies*, vol. 36, pp. 267–287, 1992. [Citado na página 11].
- [40] D. W. Aha, “Lazy learning,” *Artificial Intelligence Review*, vol. 11, pp. 7–10, 1997. [Citado na página 11].
- [41] R. Kohavi and D. Sommerfield, “Feature subset selection using the wrapper model: Overfitting and dynamic search space topology,” (Menlo Park, CA), pp. 192–197, American Association for Artificial Intelligence, American Association for Artificial Intelligence, Aug. 1995. [Citado na página 12].
- [42] H. Min and W. Fangfang, “Filter-wrapper hybrid method on feature selection,” in *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*, vol. 3, pp. 98–101, IEEE, 2010. [Citado na página 12].
- [43] Y. Lan, H. Ren, Y. Zhang, H. Yu, and X. Zhao, “A hybrid feature selection method using both filter and wrapper in mammography cad,” in *Image Analysis and Signal Processing (IASP), 2011 International Conference on*, pp. 378–382, IEEE, 2011. [Citado na página 12].
- [44] P. Estévez, M. Tesmer, C. Perez, and J. Zurada, “Normalized mutual information feature selection,” *Neural Networks, IEEE Transactions on*, vol. 20, no. 2, pp. 189–201, 2009. [Citado na página 12].
- [45] O. P. Netto and J. A. Baranauskas, “An iterative decision tree threshold filter,” in *Proceedings of the XXXII Congress of the Brazilian Computer Society, XII Workshop on Medical Informatics, ISSN 2175-2761*, (Curitiba, PR), p. 10p., 2012. <http://dcm.ffclrp.usp.br/~augusto/publications/2012-Netto-WIM-SBC.pdf>. [Citado nas páginas 12, 28, 33, e 37].
- [46] X. Shen, M. Boutell, J. Luo, and C. Brown, “Multi-label machine learning and its application to semantic scene classification,” 2004. [Citado nas páginas 13 e 17].

- [47] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” pp. 667–685, Springer, 2010. [Citado na página 15].
- [48] R. Cerri, R. da Silva, and A. de Carvalho, “Comparing methods for multilabel classification of proteins using machine learning techniques,” pp. 109–120, Springer, 2009. [Citado na página 15].
- [49] E. A. Cherman, J. Metz, and M. C. Monard, “Métodos multirrótulo independentes de algoritmo: um estudo de caso,” in *Anais da XXXVI Conferencia Latinoamericana de Informática (CLEI)*, (Asuncion, Paraguay), pp. 1–14, 2010. Publicado em CD-ROM. [Citado na página 15].
- [50] G. Tsoumakas and I. Vlahavas, “Random k-labelsets: An ensemble method for multilabel classification,” *Machine Learning: ECML 2007*, vol. 4701, pp. 406–417, 2007. [Citado na página 16].
- [51] A. Clare and R. D. King, “Knowledge discovery in multi-label phenotype data,” *Lecture Notes in Computer Science*, pp. 42–53, 2001. [Citado na página 18].
- [52] R. T. Alves, M. R. Delgado, and A. A. Freitas, “Multi-label hierarchical classification of protein functions with artificial immune systems,” *Advances in Bioinformatics and Computational Biology*, pp. 1–12, 2008. [Citado na página 19].
- [53] F. D. Comité, R. Gilleron, and M. Tommasi, “Learning Multi-label Alternating Decision Trees and Applications,” in *Proceedings of CAP’01 : Conférence en Apprentissage Automatique*, pp. 195–210, 2001. [Citado na página 19].
- [54] Y. Freund and L. Mason, “The alternating decision tree learning algorithm,” in *Proc. 16th International Conf. on Machine Learning*, pp. 124–133, Morgan Kaufmann, San Francisco, CA, 1999. [Citado na página 19].
- [55] R. E. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999. [Citado na página 19].
- [56] R. E. Schapire, “The strength of weak learnability,” *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990. [Citado na página 19].
- [57] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *Proceedings of the Second European Conference on Computational Learning Theory*, pp. 23–37, Springer-Verlag, 1995. [Citado na página 19].
- [58] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *Proceedings of the Thirteenth International Conference on Machine Learning*, (Lake Tahoe, California), pp. 123–140, Morgan Kaufmann, 1996. [Citado na página 19].
- [59] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, “Mulan: A java library for multi-label learning,” *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011. [Citado na página 19].
- [60] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd ed., 2011. [Citado nas páginas 19 e 26].

- [61] M. L. Zhang and Z. H. Zhou, “MI-knn: A lazy learning approach to multi-label learning,” *Pattern Recogn.*, vol. 40, no. 7, pp. 2038–2048, 2007. [Citado na página 19].
- [62] M. L. Zhang, “Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006. [Citado na página 19].
- [63] H. Blockeel, L. D. Raedt, and J. Ramon, “Top-down induction of clustering trees,” in *Proceedings of the 15th International Conference on Machine Learning, ICML '98*, pp. 55–63, 1998. [Citado na página 19].
- [64] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Pacific Grove, CA: Wadsworth & Books, 1984. [Citado na página 19].
- [65] H. Blockeel, L. Schietgat, J. Struyf, A. Clare, and S. Dzeroski, “Hierarchical multilabel classification trees for gene function prediction,” 2006. https://www.cs.helsinki.fi/bioinformatiikka/events/pmsb06/final/blockeel_et_al.pdf. [Citado na página 19].
- [66] E. A. Tanaka and J. A. Baranauskas, “An adaptation of binary relevance for multi-label classification applied to functional genomics,” in *Proceedings of the XXXII Congress of the Brazilian Computer Society, XII Workshop on Medical Informatics, ISSN 2175-2761*, (Curitiba, PR), p. 10p., 2012. <http://dcm.ffclrp.usp.br/~augusto/publications/2012-Tanaka-WIM-SBC.pdf>. [Citado nas páginas 20, 28, 33, e 37].
- [67] E. A. Tanaka, S. R. Nozawa, A. A. Macedo, and J. A. Baranauskas, “A multi-label approach using binary relevance and decision trees applied to functional genomics,” *Journal of Biomedical Informatics*, vol. 54, pp. 85–95, 2015. <http://dx.doi.org/10.1016/j.jbi.2014.12.011>. [Citado nas páginas 20, 27, 33, 34, e 37].
- [68] T. M. Mitchell, “Generalization as search,” *Artificial Intelligence*, vol. 18, pp. 203–226, 1982. Reprinted in Shavlik and Dietterich (eds.), 1990. *Readings in Machine Learning*, Morgan Kaufmann Publishers, Inc. [Citado na página 20].
- [69] T. G. Dietterich and E. B. Kong, “Machine learning bias, statistical bias, and statistical variance of decision tree algorithms,” 1997. [Citado na página 20].
- [70] L. Breiman, “Bias, variance and arcing classifiers,” Tech. Rep. 460, Statistics Department, University of California, april 1996. [Citado na página 20].
- [71] E. Bauer and R. Kohavi, “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants,” *Machine Learning*, vol. 36, pp. 105–139, 1999. [Citado na página 20].
- [72] T. G. Dietterich, “Machine learning research: Four current directions,” may 1997. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.5440>. [Citado nas páginas 20, 21, e 22].
- [73] L. Breiman, “The heuristics on instability in model selection,” tech. rep., Statistics Department, University of California, 1996. [Citado na página 20].

- [74] J. A. Baranauskas and M. C. Monard, “Reviewing some machine learning concepts and methods,” Tech. Rep. 102, ICMC-USP, jan 2000. http://dcm.ffclrp.usp.br/~augusto/publications/rt_102.pdf. [Citado nas páginas 22 e 29].
- [75] D. Opitz and R. Maclin, “Popular ensemble methods: An empirical study,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999. [Citado na página 22].
- [76] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, pp. 241–259, 1992. [Citado na página 23].
- [77] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Chapman & Hall, 1993. [Citado na página 23].
- [78] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Citado na página 24].
- [79] Y. Zhao and Y. Zhang, “Comparison of decision tree methods for finding active objects,” *Advances in Space Research*, vol. 41, pp. 1955–1959, 2008. [Citado na página 24].
- [80] P. Dubath, L. Rimoldini, M. Süveges, J. Blomme, M. López, L. M. Sarro, J. De Ridder, J. Cuypers, L. Guy, I. Lecoœur, K. Nienartowicz, A. Jan, M. Beck, N. Mowlavi, P. De Cat, T. Lebzelter, and L. Eyer, “Random forest automated supervised classification of hipparcos periodic variable stars,” *Monthly Notices of the Royal Astronomical Society*, vol. 414, no. 3, pp. 2602–2617, 2011. [Citado na página 24].
- [81] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. [Citado na página 24].
- [82] L. Breiman and A. Cutler, “Random forests: Classification/clustering.” <http://www.stat.berkeley.edu/users/breiman/RandomForests>, 2004. [Citado na página 24].
- [83] L. Breiman, “Wald lecture ii, looking inside the black box.” <http://www.stat.berkeley.edu/users/breiman>, 2004. [Citado na página 24].
- [84] Y. Ma, L. Guo, and B. Cukic, “Statistical framework for the prediction of fault-proneness,” in *Advances in machine learning applications in software engineering*, Idea Group, 2007. [Citado na página 24].
- [85] P. S. Perez and J. A. Baranauskas, “Analysis of decision tree pruning using windowing in medical datasets with different class distributions,” in *Proceedings of the ECML PKDD Workshop on Knowledge Discovery in Health Care and Medicine (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases)*, (Athens, Greece), pp. 28–39, 2011. <http://www.cs.gmu.edu/~hrangwal/kd-hcm/proc/papers/3-Perez-Baranauskas.pdf>. [Citado nas páginas 25, 28, 33, e 37].
- [86] P. S. Perez, S. R. Nozawa, A. A. Macedo, and J. A. Baranauskas, “Windowing improvements towards more comprehensible models,” *Knowledge-Based Systems*, vol. 92, pp. 9–22, 2016. <http://dx.doi.org/10.1016/j.knosys.2015.10.011>. [Citado nas páginas 25, 27, 33, 34, 35, e 37].

- [87] T. M. Oshiro and J. A. Baranauskas, “Root attribute behavior within a random forest,” in *Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science*, (Natal, RN), August 29-31 2012. http://dx.doi.org/10.1007/978-3-642-32639-4_87. [Citado nas páginas 25, 28, 33, e 37].
- [88] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, “How many trees in a random forest?,” in *Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM 2012, Lecture Notes in Computer Science, ISBN 978-3-642-31536-7*, vol. 7376, (Berlin, Germany), pp. 154–168, July 13-20 2012. http://dx.doi.org/10.1007/978-3-642-31537-4_13. [Citado nas páginas 25, 28, 33, e 37].
- [89] M. I. R. Caffé, P. S. Perez, and J. A. Baranauskas, “Avaliação do algoritmo de stacking em dados biomédicos,” in *XXXI Congresso da Sociedade Brasileira de Computação, XI Workshop de Informática Médica, ISSN 2175-2761, (Artigo premiado como melhor artigo completo do evento)*, (Natal, RN), p. 10p., 2011. <http://dcm.ffclrp.usp.br/~augusto/publications/2011-Caffe-WIM-SBC.pdf>. [Citado nas páginas 25, 28, 33, e 37].
- [90] M. I. R. Caffé, P. S. Perez, and J. A. Baranauskas, “Evaluation of stacking on biomedical data,” *Journal of Health Informatics, ISSN 2175-4411*, vol. 4, no. 3, pp. 67–72, 2012. <http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/181/119>. [Citado nas páginas 25, 28, 33, e 37].
- [91] P. M. Fugimoto, L. D. F. Sales, G. A. P. Júnior, A. D. C. Passos, D. Alves, and J. A. Baranauskas, “Análise comparativa entre Árvores de decisão e triss na predição de sobrevivência de pacientes traumatizados,” in *IV Congresso da Academia Trinacional de Ciências, ISSN 1982-2758*, (Foz do Iguaçu, PR), p. 10p., 2009. <http://dcm.ffclrp.usp.br/~augusto/publications/2009-Fugimoto-C3N.pdf>. [Citado nas páginas 26, 28, 33, e 37].
- [92] M. C. Monard and J. A. Baranauskas, *Conceitos sobre Aprendizado de Máquina*, ch. 4, pp. 89–114. In Rezende [110], 2003. http://books.google.com.br/books?id=UsJe_P1bnWc&lpg=PP1&pg=PA89#v=onepage&q&f=false. [Citado nas páginas 26, 29, 31, 33, e 37].
- [93] L. D. F. Sales, P. M. Fugimoto, A. D. C. Passos, G. A. P. Júnior, D. A. D., and J. A. Baranauskas, “Análise da metodologia triss em pacientes traumatizados do hospital das clínicas da faculdade de medicina de ribeirão preto,” in *XII Congresso Brasileiro de Informática em Saúde, ISSN 2178-2857*, (Porto de Galinhas, PE), p. 6p., 2010. <http://dcm.ffclrp.usp.br/~augusto/publications/2010-Sales-CBIS.pdf>. [Citado nas páginas 26, 28, 33, e 37].
- [94] E. A. Tanaka, C. M. Junta, L. Vagnini, J. A. Baranauskas, and S. Giuliatti, “Um sistema computacional integrando suporte à decisão na Área de reprodução humana,” in *XII Congresso Brasileiro de Informática em Saúde, ISSN 2178-2857*, (Porto de Galinhas, PE), p. 6p., 2010. <http://dcm.ffclrp.usp.br/~augusto/publications/2010-Tanaka-CBIS.pdf>. [Citado nas páginas 26, 28, 33, e 37].

- [95] P. S. Perez, A. H. Bevilacqua, A. Ghelfi, A. A. Macedo, S. R. Nozawa, and J. A. Baranauskas, "A software tool for information management and data mining of biological data for studying adaptation of living organisms in amazonia," in *Proceedings of the International Conference on Computational Biosciences and Bio-Informatics (IC-CBBI 2011)*, (Bhubaneswar, India), 2011. <http://dcm.ffclrp.usp.br/~augusto/publications/2011-Perez-icbbi2011.pdf>. [Citado nas páginas 27, 28, 33, 34, e 37].
- [96] O. P. Netto, P. M. A. M. A. A. Macedo, and J. A. Baranauskas, "Uma metodologia para estruturação de laudos médicos usando ontologias," in *XXXI Congresso da Sociedade Brasileira de Computação, XI Workshop de Informática Médica, ISSN 2175-2761*, (Natal, RN), p. 10p., 2011. <http://dcm.ffclrp.usp.br/~augusto/publications/2011-Netto-WIM-SBC.pdf>. [Citado nas páginas 27, 28, 33, 34, 35, e 37].
- [97] M. C. Monard and J. A. Baranauskas, *Indução de Regras e Árvores de Decisão*, ch. 5, pp. 115–139. In Rezende [110], 2003. http://books.google.com.br/books?id=UsJe_P1bnWcC&lpg=PP1&pg=PA115#v=onepage&q&f=false. [Citado nas páginas 29, 31, 33, e 37].
- [98] J. A. Baranauskas and M. C. Monard, "An unified overview of six supervised symbolic machine learning inducers," Tech. Rep. 103, ICMC-USP, jan 2000. http://dcm.ffclrp.usp.br/~augusto/publications/rt_103.pdf. [Citado na página 29].
- [99] J. T. Pollettini, J. A. Baranauskas, E. S. Ruiz, M. da Graça Pimentel, and A. A. Macedo, "Surveillance for the prevention of chronic diseases through information association," *BMC Medical Genomics*, vol. 7, no. 1, pp. 1–11, 2014. <http://dx.doi.org/10.1186/1755-8794-7-7>. [Citado nas páginas 29, 31, 33, 34, e 37].
- [100] F. Lira, P. S. Perez, J. A. Baranauskas, and S. R. Nozawa, "Prediction of antimicrobial activity of synthetic peptides by a decision tree model," *Applied and Environmental Microbiology*, 2013. <http://dx.doi.org/10.1128/AEM.02804-12>. [Citado nas páginas 30, 31, 33, 35, e 37].
- [101] J. A. Baranauskas, "The number of classes as a source for instability of decision tree algorithms in high dimensional datasets," *Artificial Intelligence Review*, 2012. <http://dx.doi.org/10.1007/s10462-012-9374-7>. [Citado nas páginas 30, 31, 33, e 37].
- [102] J. T. Pollettini, S. R. G. Panico, J. C. Daneluzzi, R. Tinós, J. A. Baranauskas, and A. A. Macedo, "Using machine learning classifiers to assist healthcare-related decisions: Classification of electronic patient records," *Journal of Medical Systems*, vol. 36, pp. 3861–3874, 2012. <http://dx.doi.org/10.1007/s10916-012-9859-6>. [Citado nas páginas 30, 31, 33, 34, e 37].
- [103] J. T. Pollettini, F. P. Nicolas, S. R. G. Panico, J. C. Daneluzzi, R. Tinós, J. A. Baranauskas, and A. A. Macedo, "A software architecture-based framework supporting suggestion of medical surveillance level from classification of electronic patient records," in *The 12th IEEE International Conference on Computational Science and Engineering, International Conference on Computational Science and Engineering*, (Vancouver, Canadá), pp. 166–173, 2009. <http://dx.doi.org/10.1109/CSE.2009.231>. [Citado nas páginas 30, 31, 34, e 37].

- [104] N. G. Cairasco, F. Rossetti, G. M. Arisi, M. L. Foresti, P. Berti, O. W. Castro, A. Fernandes, D. C. Borragini, M. C. Rodrigues, R. N. R. Pereira, and J. A. Baranauskas, “Experimental and human epileptogenic circuits as sources of complexity and emergent properties suitable for neural network modeling,” in *62nd Annual Meeting of the American Epilepsy Society*, vol. 49, (Seattle, WA), pp. 342–343, 2008. https://www.aesnet.org/meetings_events/annual_meeting_abstracts/view/8969. [Citado nas páginas 30, 31, e 33].
- [105] F. A. Leone, J. A. Baranauskas, R. P. M. Furriel, and I. A. Borin, “Sigrafw: An easy-to-use program for fitting enzyme kinetic data,” *Biochemistry and Molecular Biology Education*, vol. 33, pp. 399–403, 2005. <http://dx.doi.org/10.1002/bmb.2005.49403306399>. [Citado nas páginas 31, 33, e 37].
- [106] F. A. Leone, P. Ciancaglini, and J. A. Baranauskas, “Enzyplot: A microcomputer assisted program for teaching enzyme kinetics,” *Biochemical Education*, vol. 23, pp. 35–37, 1995. [http://dx.doi.org/10.1016/0307-4412\(94\)00159-m](http://dx.doi.org/10.1016/0307-4412(94)00159-m). [Citado na página 37].
- [107] F. A. Leone, L. Degrève, and J. A. Baranauskas, “Sigraf: A versatile computer program for the fitting of enzyme kinetic data,” *Biochemical Education*, vol. 20, pp. 94–96, 1992. [http://dx.doi.org/10.1016/0307-4412\(92\)90109-y](http://dx.doi.org/10.1016/0307-4412(92)90109-y). [Citado na página 37].
- [108] M. C. Monard and J. A. Baranauskas, “Aplicações de inteligência artificial: Uma visão geral,” in *I Congresso de Lógica Aplicada à Tecnologia (LAPTEC)*, (Rio de Janeiro), pp. 339–348, out 2000. <http://dcm.ffclrp.usp.br/~augusto/publications/2000-laptec.pdf>. [Citado na página 37].
- [109] J. A. Baranauskas and M. C. Monard, “Metodologias para seleção de atributos,” in *Workshop de Teses e Dissertações do Simpósio Brasileiro de Inteligência Artificial (SBIA)*, 1998. <http://dcm.ffclrp.usp.br/~augusto/publications/1998-sbia.pdf>. [Citado na página 37].
- [110] S. O. Rezende, ed., *Sistemas Inteligentes - Fundamentos e Aplicações*. Manole, 2003. [Citado nas páginas 46 e 47].
- [111] M. C. Monard and J. S. Sichman, eds., *Proceedings of the International Joint Conference 7th Ibero-American Conference on Artificial Intelligence and Brazilian Artificial Intelligence Symposium (IBERAMIA/SBIA)*, (Atibaia, SP), 2000. [Citado na página 40].