UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MILTON SOARES DE OLIVEIRA JUNIOR

PREVISÃO DA QUANTIDADE DE CLASSES EM UM PROBLEMA DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2016

MILTON SOARES DE OLIVEIRA JUNIOR

PREVISÃO DA QUANTIDADE DE CLASSES EM UM PROBLEMA DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Geraldo Ranthum

PONTA GROSSA

2016

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Bacharelado em Ciência da Computação

TERMO DE APROVAÇÃO

PREVISÃO DA QUANTIDADE DE CLASSES EM UM PROBLEMA DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO

por

MILTON SOARES DE OLIVEIRA JUNIOR

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 24 de novembro de 2016 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Prof. MSc. GERALDO RANTHUM
Orientador

_____
Prof. Dr. GLEIFER VAZ ALVES
Membro titular

_____
Prof. Esp. MARCOS VINICIUS FIDÉLIS
Membro titular

_____
Prof. Dra. SIMONE DE ALMEIDA
Membro titular

_____
Prof. Dr. Augusto Foronda
Responsável pelo Trabalho de Conclusão de Curso

_____
Prof. Dr. Erikson Freitas de Morais
Coordenador do curso

À minha mãe que tornou possível e mais fácil chegar até aqui graças ao seu apoio, amor e dedicação.

AGRADECIMENTOS

Antes de citar as pessoas as quais agradeço pela realização deste trabalho, acredito ser necessário dizer que os bons momentos e as conquistas de nossas vidas só são possíveis quando não estamos sozinhos. Portanto, ter realizado este trabalho foi fruto de um conjunto de pessoas as quais sempre estiveram ao meu lado.

Primeiramente, agradeço à minha mãe que fez de tudo para me apoiar ao longo desta jornada que algumas vezes parecia não ter fim. Agradeço por ter depositado toda sua confiança em mim e por me confortar em momentos de desespero quando eu mesmo já não acreditava poder continuar.

Aos amigos que deixei temporariamente em São Paulo, especialmente Leandro e Patrick, que tornam minha vida muito mais interessante e são parte essencial na construção da minha personalidade. Agradeço por esses quase 10 anos de amizade e que continuemos como havíamos conversado em um dos nossos últimos encontros - não mudando nada.

Agradeço aos amigos de Ponta Grossa que fizeram parte dos melhores anos da minha vida. Aos vizinhos do condomínio que me tornaram uma pessoa melhor e que certamente pretendo levar para o resto da vida. Desejaria citar todos os nomes aqui, mas a quantidade de pessoas queridas é tanta que não caberia nesse pequeno espaço. Vocês estão entre as pessoas mais importantes que conheci.

Aos amigos da turma de 2012-2 que compartilharam das mesmas dificuldades e alegrias durante a graduação, inclusive os que seguiram outro caminho, mas ainda continuam fazendo parte da minha vida.

À professora e minha orientadora inicial Helyane Borges, que foi responsável por despertar meu interesse pela área de pesquisa deste trabalho e que mesmo tendo que sair de licença para o pós-doutorado, não mediu esforços para continuar me ajudando até onde podia. Agradeço por todo o referencial teórico e por estabelecer tão bem definidamente a prática deste trabalho durante o período em que me orientou.

Por fim, agradeço ao meu orientador atual Geraldo Ranthum, que aceitou assumir o lugar de última hora, mas mesmo assim não mediu esforços para me ajudar a desenvolver e terminar a pesquisa.

RESUMO

OLIVEIRA JR, Milton Soares. **Previsão da Quantidade de Classes em um Problema de Classificação Hierárquica Multirrótulo.** 2016. 67. Trabalho de Conclusão de Curso Bacharelado em Ciência da Computação - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2016.

Em problemas de classificação hierárquica multirrótulo, cada exemplo pode estar associado a uma ou mais classes simultaneamente pertencendo a um nível hierárquico de subclasse ou superclasse. Essa classificação é realizada através de técnicas de aprendizagem, que a partir de uma base de dados treinada, define o conjunto de classes que uma instância estará associada. Porém, há um problema na classificação multirrótulo que está associado aos classificadores das técnicas, que não preveem a quantidade de classes que será definida para novas instâncias apresentadas ao modelo de classificação. Na atual literatura de Aprendizagem de Máquina e Mineração de Dados, esses classificadores apenas definem na fase de teste o conjunto de classes de um exemplo de entrada com quantidade já pré-definida na fase de treinamento. Portanto, neste trabalho serão investigadas e adaptadas técnicas de classificação capazes de prever a quantidade de classes que será associada a uma nova instância, tal que ela esteja contida em um conjunto de entrada hierárquico e multirrótulo.

Palavras-chave: Classificação Hierárquica Multirrótulo. Aprendizagem de Máquina. Mineração de Dados.

ABSTRACT

OLIVEIRA JR, Milton Soares. **Forecast of Label numbers in a Hierarchical Multi-label Classification problem.** 2016. 67. Work of Conclusion Course Graduation in Ciência da Computação - Federal Technology University - Paraná. Ponta Grossa, 2016.

In hierarchical multi-label classification problems, each instance may be associated with one or more labels simultaneously belonging to a hierarchical level subclass or superclass. This classification is performed through learning techniques that from a trained database defines the set of labels that an instance is associated. However, an existing multi-label problem in rating refers to classifiers used in existing techniques which do not provide the number of labels to be defined for new instances presented to the classification model. In the current literature of Learning Machine and Data Mining, these classifiers only define in the training phase the set of labels of entry example with amount already preset in the training phase. Therefore, in this study will be investigated and adapted classification techniques able to predict the number of labels to be associated with a new instance, such that it is contained in a hierarchical and multi-label entries.

Keywords: Hierarchical Multi-label Classification. Learning Machine. Data Mining.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

AD	Árvores de Decisão
AM	Aprendizado de Máquina
ARFF	*Attribute-Relation File Format*
AUC	*Area Under the ROC Curve*
DAG	Grafo Acíclico Direcionado
DM	*Data Mining*
GO	Gene Ontológico
IA	Inteligência Artificial
KNN	*K-Nearst Neighbour*
MLP	*Multilayer Perceptron*
RN	Redes Neurais
RNA	Redes Neurais Artificiais
ROC	*Receiver Operating Characteristic curve*
SVMs	Máquinas de Vetores de Suporte

SUMÁRIO

1 INTRODUÇÃO

Classificação de dados é o processo de encontrar um modelo que descreva as diferentes classes presentes em um conjunto de dados, ou seja, extrair informações a partir de um conjunto de dados por meio de sua categorização (WITTEN; FRANK; HALL, 2011). Por exemplo, em uma aplicação bancária, clientes que possuam um cartão de crédito podem ser classificados como “risco baixo”, “risco normal” ou “risco alto”. De maneira geral, o processo de classificação consiste em atribuir rótulos aos dados, de tal maneira que representem de alguma forma os dados categorizados sob o mesmo rótulo (CERRI, 2010).

Para Mitchel (1997), a classificação faz parte de um tipo de aprendizagem de máquina (AM) chamado de aprendizagem supervisionada, em que são desenvolvidos e aplicados algoritmos que realizam induções de classificadores a partir de exemplos previamente classificados. Assim, a aprendizagem supervisionada é um modelo de classificação utilizando exemplos que contém a informação da sua saída esperada. Esse modelo é obtido por meio de um algoritmo de indução (indutor), que tem como objetivo fazer com que o classificador seja capaz de classificar corretamente novos exemplos (CERRI, 2010).

Em um problema de classificação, inicialmente, os dados pertencentes ao domínio sobre o qual será aplicado o algoritmo devem ser preparados para serem representados de forma adequada para processamento. Eles devem ser organizados em um conjunto de exemplos de maneira que cada exemplo seja representado por uma tupla (linha) de atributos. Os atributos de entrada representam as características de cada exemplo (variáveis independentes), e são utilizados para induzir o classificador. O atributo de saída representa as classes que são associadas aos exemplos (variável dependente) (CERRI, 2010).

Existem dois tipos principais de classificação: a tradicional, também conhecida como plana e a multirrótulo. Em problemas de classificação tradicional, um classificador é treinado a partir de um conjunto de instâncias (exemplos) onde cada uma está associada somente a uma classe. Por outro lado, em um problema de Classificação Multirrótulo, cada instância pode estar associada a mais de uma classe simultaneamente.

Por exemplo, em um problema de classificação plana de gêneros musicais, para classificar a música *Voodoo Child* do artista Jimi Hendrix, pode-se perguntar: “A

música *Voodoo Child* é Rock ou Pop?". Nessa classificação, existem duas possíveis classes: "Rock" e "Pop". Para o exemplo em questão, a resposta correta é a classe "Rock" (SILVA, 2014).

Outro exemplo seria na classificação multirrótulo (BORGES, 2012). Neste caso, uma instância terá mais de uma classe atribuída a ela. No exemplo anterior, de classificação de gêneros musicais, para classificar a música *Voodoo Child* do artista Jimi Hendrix, pode-se perguntar: "A música *Voodoo Child* é Rock, Blues ou Pop?". Nessa classificação, existem três classes possíveis: "Rock", "Blues" e "Pop". Numa classificação plana, se a saída do classificador fosse "Rock" a resposta estaria parcialmente correta, pois a resposta certa engloba duas classes simultaneamente, "Rock" e "Blues". Por esta razão, a saída esperada para um classificador multirrótulo será "Rock e Blues".

Problemas de classificação também podem ser divididos entre não hierárquicos (*Flat Classification*), em que cada exemplo é associado a uma classe pertencente a um conjunto finito de classes, todas em um mesmo nível, e também podem ser tipos de classificação hierárquicos, em que uma ou mais classes podem ser divididas em subclasses ou agrupadas em superclasses. Na classificação hierárquica há mais flexibilidade para especificar para cada nível da hierarquia uma classe que será definida para uma instância (FREITAS; CARVALHO, 2007).

Em sua tese, Borges (2012), explica que uma das principais aplicações de técnicas de classificação hierárquica é em problemas de classificação na Bioinformática, como por exemplo, a predição de funções de proteínas, que são macromoléculas formadas por longas sequências de aminoácidos e que executam quase todas as funções celulares nos seres vivos. Essas técnicas também são aplicadas em bases de dados da genômica funcional, que foram utilizadas neste trabalho, e são conhecidas como bases GO (Gene Ontológico).

Porém, um problema existente e ainda sem solução na classificação multirrótulo e classificação hierárquica multirrótulo, dada uma situação real, é prever a quantidade de classes quando uma nova instância é apresentada ao modelo de classificação. Por exemplo, existem três classes que uma instância pode assumir. No exemplo anterior a música assumiu duas classes ("Rock" e "Blues"), porque durante a indução do classificador este foi preparado para atribuir à instância exatamente este número. Caso uma nova música fosse apresentada ao modelo de

classificação, quantas classes deveriam ser atribuídas a este exemplo dadas suas características?

Portanto, a proposta deste trabalho é investigar e aplicar técnicas de classificação em bases de dados hierárquicas multirrótulo capazes de predizer a quantidade de classes para cada nova instância na fase de classificação.

1.1 DESCRIÇÃO DO PROBLEMA

A classificação hierárquica multirrótulo é uma área relativamente nova na literatura de aprendizagem de máquina, como disse Coelho (2011), e por ser abrangente faz com que pesquisas sejam desenvolvidas para aprimorar e criar novas técnicas de classificação.

Embora as técnicas existentes apresentem resultados satisfatórios e sejam capazes de formularem uma saída esperada de acordo com a entrada, existem, no entanto, problemas para os quais essas mesmas técnicas não apresentam soluções adequadas. Um desses problemas, por exemplo, é quando se tem uma nova instância apresentada ao modelo e precisa-se, antes de classificá-lo, definir a quantidade de classes que a ele será atribuída.

1.2 OBJETIVOS

Os objetivos deste trabalho são descritos a seguir. A seção 1.2.1 descreve o objetivo geral do trabalho. A seção 1.2.2 descreve os objetivos específicos do trabalho.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é investigar técnicas de classificação hierárquica multirrótulo que possam ser capazes de prever a quantidade de classes (rótulos) de uma nova instância de entrada.

1.2.2 Objetivos Específicos

- Estudar os principais conceitos de classificação hierárquica multirrótulo;
- Investigar as técnicas mais comuns da literatura sobre classificação;
- Aplicar técnicas estudadas a novas instâncias de entrada de uma base de dados hierárquica multirrótulo;
- Analisar os resultados através de avaliação de classificadores;
- Aplicar técnicas avaliadas para a previsão da quantidade de classes sobre novas instâncias de entrada.

1.3 JUSTIFICATIVA

O contexto em que o problema de classificação está inserido é abrangente, e por essa razão trabalhos que envolvam esse tema são modularizados em problemas menores. Um desses problemas trata-se da definição da quantidade de classes que uma instância deverá assumir levando em conta um modelo sobre uma base de dados treinada.

Na literatura atual, ainda não são exploradas técnicas de predição baseadas em AM para a previsão da quantidade de classes de uma nova instância em um problema hierárquico multirrótulo. As técnicas de classificação atuais são capazes apenas de predizer corretamente a classe à qual pertence um exemplo de entrada baseando-se em suas características, mas não preveem a quantidade de classes que a elas serão atribuídas.

Partindo de técnicas existentes de AM para classificação, será possível com a realização deste trabalho, prever a quantidade de classes em cada nível hierárquico de uma base de dados multirrótulo antes de predizer as classes de um conjunto de entrada.

1.4 TRABALHO RELACIONADO

O trabalho de Santos e Almeida (2014), intitulado de “Previsão da Quantidade de Classes em Classificação Hierárquica Multirrótulo", possui considerada relação a este. Em seu trabalho, as autoras adaptaram uma técnica de AM em bases de dados GO - duas delas utilizadas neste trabalho - para determinar o número de classes que pode ser atribuído a um exemplo real não pertencente à base de treinamento.

Em sua metodologia, as autoras dividiram a aplicação dos métodos em três fases: estudo dos métodos, aplicação dos métodos em diferentes bases de dados e avaliação e análise dos resultados. Todos esses métodos foram aplicados utilizando o *framework* Mulan (TSOUMAKAS et al, 2011), que implementa uma gama de métodos multirrótulo e disponibiliza diversas medidas de avaliação. Ao contrário do *framework* utilizado neste trabalho, que apenas processa bases planas e simples-rótulo.

Os resultados dos experimentos realizados para predição de classes foram obtidos a partir da comparação e análise das técnicas propostas utilizando o algoritmo ML-KNN (*Multi-label K-nearest neighbour*) – uma extensão do KNN para tratamento multirrótulo - que classifica uma instância da base comparando-a com todas as instâncias pertencentes ao conjunto de treinamento por meio do cálculo de distância (reta de Euler).

O objetivo geral, assim como a deste trabalho, de prever a quantidade de classes foi alcançado, seguindo a metodologia proposta pelas autoras, inicialmente pela adaptação do algoritmo hierárquico multirrótulo ML-KNN e sua aplicação nas bases de dados propostas para obtenção e avaliação dos resultados.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho é constituído de cinco capítulos. O Capítulo 2 aborda o referencial teórico necessário para o desenvolvimento do trabalho. Os temas abordados são mineração de dados e aprendizagem de máquina, técnicas de classificação, classificação multirrótulo e classificação hierárquica.

O capítulo 3 apresenta a metodologia utilizada no desenvolvimento deste trabalho. Os temas abordados são: bases de dados, ferramentas utilizadas e metodologia para realização dos experimentos.

O capítulo 4 contém a realização dos experimentos e análise de resultados contendo os seguintes temas: resultados dos métodos de classificação e análise comparativa dos métodos de classificação.

Por fim, no capítulo 5, é descrita a conclusão contendo as considerações finais do trabalho e trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo descreve os principais conceitos teóricos que fundamentam o desenvolvimento deste trabalho. A Seção 2.1 descreve conceitos de mineração de dados e aprendizagem de máquina. Da Seção 2.2 à seção 2.4 são conceituadas as técnicas de classificação, classificação multirrótulo e classificação hierárquica. E por fim, a Seção 2.6 contém as considerações finais do capítulo.

2.1 MINERAÇÃO DE DADOS E APRENDIZAGEM DE MÁQUINA

A quantidade de dados armazenados vem crescendo cada vez mais nos últimos anos. Essa grande quantidade de dados armazenados contém valiosos conhecimentos ocultos, que poderiam ser utilizados para análise, diagnóstico, simulação e/ou prognóstico do processo que gerou a base de dados (HAN; KAMBER; PEI, 2012).

A análise de grandes quantidades de dados pelo homem é inviável sem o auxílio de ferramentas computacionais apropriadas. Portanto, torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação (GOLDSCHMIDT; PASSOS, 2005).

Para entender este contexto, surge uma área denominada Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases* - KDD). O termo KDD foi formalizado em 1989 em referência ao amplo conceito de procurar conhecimento a partir de bases de dados. Uma das definições mais populares foi proposta por Fayyad *et al* (1996): “KDD é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados”.

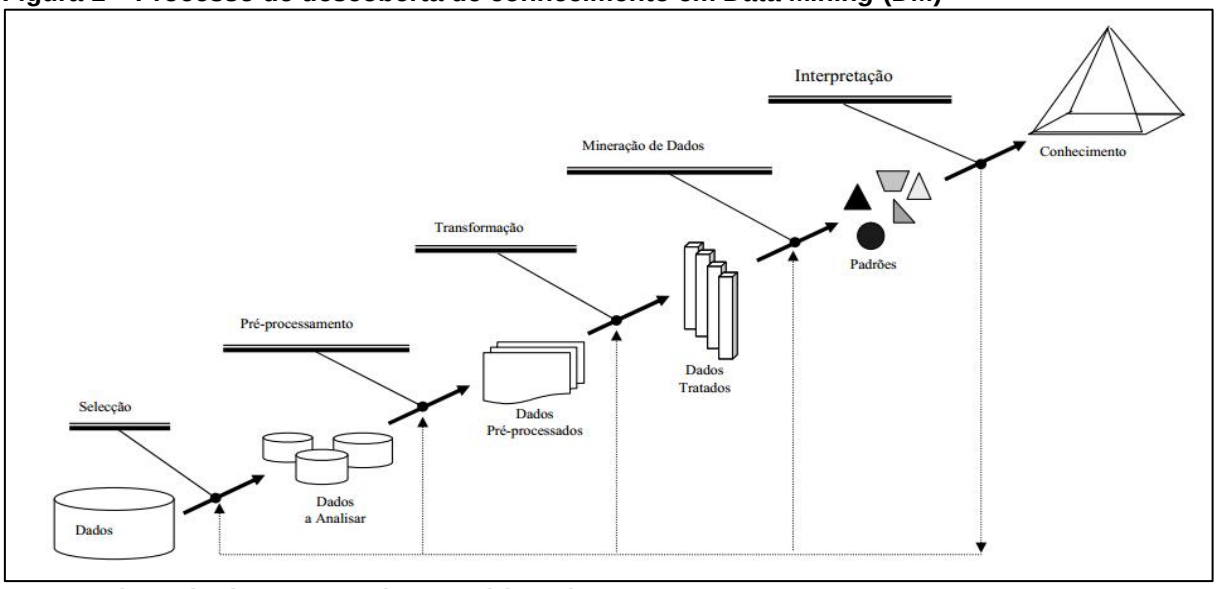
Mineração de dados ou *Data Mining* (DM) é uma das etapas de KDD e é definida como o processo de descoberta de padrões de dados. Este processo pode ser automático ou semiautomático. Os padrões descobertos devem ser significativos na medida em que apresentam alguma vantagem, normalmente econômica. Os

dados estão invariavelmente presentes em quantidades substanciais (WITTEN; FRANK; HALL, 2011). DM também pode ser descrita como extração de conhecimento de uma grande quantidade de dados. Sendo uma área multidisciplinar, que utiliza métodos de várias outras áreas, especialmente de Aprendizado de Máquina (AM) e Estatística para extrair conhecimentos a partir de conjuntos de dados (COELHO, 2011).

Em seu livro, Jain e Ghost (2005) descrevem DM como sendo um dos campos mais desenvolvidos na área de Inteligência Artificial (IA). Logo, a utilização direta e eficiente de um grande volume de dados é um grande desafio. Portanto, existe uma necessidade de se utilizar métodos semiautomáticos para extrair conhecimento destes dados.

Além da etapa de DM, que efetivamente extrai o conhecimento a partir de dados, o processo de mineração inclui outras etapas como pré-processamento (ou preparação de dados) e pós-processamento (ou refinamento de conhecimento). Como é mostrado na Figura 1, o objetivo do pré-processamento de dados é transformar os dados para facilitar a aplicação de uma ou várias técnicas de DM, enquanto que o objetivo dos métodos de refinamento do conhecimento é validar e aperfeiçoar o conhecimento descoberto (COELHO, 2011).

Figura 1 – Processo de descoberta de conhecimento em *Data Mining* (DM)



Fonte: Adaptado de Han, Kamber e Pei (2012)

O pré-processamento dos dados é necessário, na maioria das vezes, para permitir que os dados sejam utilizados adequadamente no processo de DM. Para a realização desta etapa alguns passos devem ser seguidos (COELHO, 2011):

- Limpeza dos dados;
- Integração dos dados;
- Transformação dos dados;
- Discretização dos dados (normalização dos dados);
- Redução dos dados;
- Seleção dos dados.

Segundo Mitchell (1997), a aprendizagem de máquina aborda a questão de como são construídos programas de computador que melhorem seu desempenho através da experiência. Algoritmos de aprendizagem tem grande valor prático em uma variedade de domínios de aplicação. Eles são especialmente úteis em:

- Problemas de extração de dados, onde bases de dados podem conter regularidades implícitas valiosas que podem ser descobertas automaticamente. Por exemplo, para analisar resultados de tratamentos médicos da base de dados de pacientes ou aprender regras gerais para merecimento de crédito de bases de dados financeiros;
- Domínios mal compreendidos onde seres humanos podem não ter o conhecimento necessário para desenvolver algoritmos eficientes. Por exemplo, reconhecimento de faces humanas a partir de uma imagem;
- Domínios onde o programa tem de se adaptar dinamicamente às mudanças das condições. Por exemplo, controle de fabricação sobre mudanças de estoque e fornecedores.

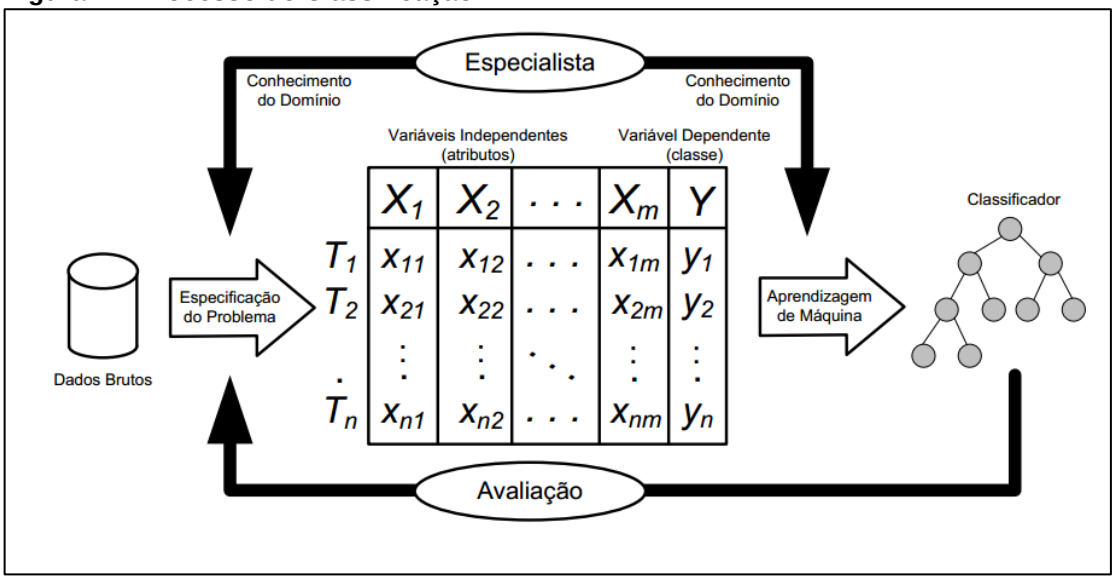
2.2 TÉCNICAS DE CLASSIFICAÇÃO

Classificação é o processo de encontrar um modelo ou função que descreve e distingue classes de dados ou conceitos, a fim de ser capaz de usar o modelo para prever a classe de um objeto cuja classe é ainda desconhecida. O modelo derivado é baseado na análise de um conjunto de dados de treinamento (COELHO, 2011).

A classificação faz parte de um tipo de aprendizado de máquina chamado de aprendizado supervisionado, em que são desenvolvidos algoritmos que realizam induções de classificadores a partir de exemplos previamente classificados. Assim, é obtido um classificador utilizando exemplos que contêm a informação da sua saída esperada. Esse classificador é obtido por meio de um algoritmo de indução (indutor), que tem como objetivo fazer com que o classificador seja capaz de classificar corretamente novos exemplos.

Inicialmente, os dados pertencentes ao domínio sobre o qual será aplicado o algoritmo de classificação devem ser preparados para serem representados de forma adequada para processamento. Eles devem ser organizados em um conjunto de exemplos de maneira que cada exemplo seja representado por uma tupla de atributos. Os atributos de entrada representam as características de cada exemplo (variáveis independentes), e são utilizados para induzir o classificador. O atributo de saída representa as classes que são associadas aos exemplos (variável dependente) (CERRI, 2010). A Figura 2 apresenta um esquema de um processo de classificação.

Figura 2 – Processo de Classificação



Fonte: Rezende (2005)

Após a obtenção do classificador, o mesmo é avaliado por meio da apresentação de novos exemplos, que não foram utilizados durante o treinamento. Essa etapa é conhecida como fase de teste ou validação.

Segundo Cerri (2011), podem ser encontrados muitos algoritmos de classificação. Esses algoritmos são divididos de acordo com o paradigma de classificação a que pertencem.

2.2.1 Técnicas e Paradigmas de Classificação

Nesta seção são abordados os principais métodos de classificação utilizados no desenvolvimento deste trabalho. São eles: *K-Vizinhos mais próximos* (*K-Nearst Neighbour* - KNN), Redes Neurais (RN) e Árvores de Decisão (AD).

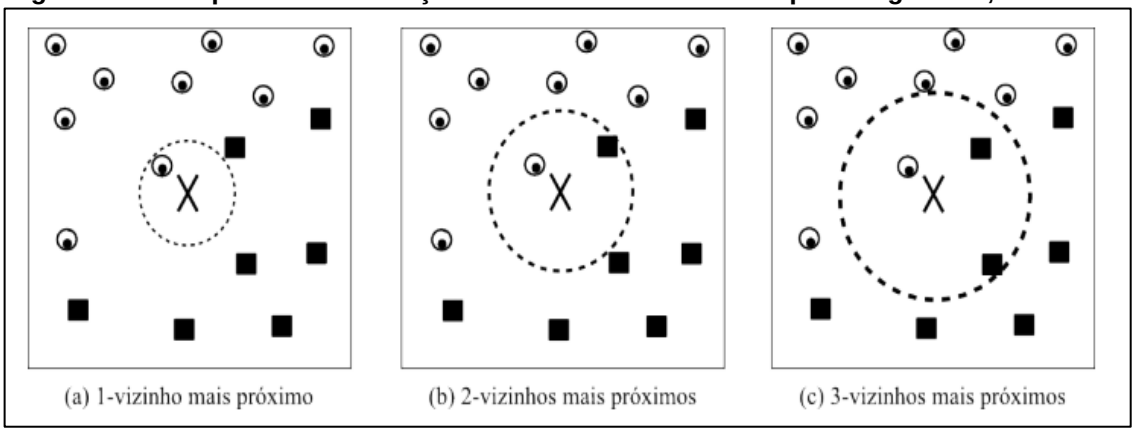
2.2.1.1 KNN

O método mais básico baseado em exemplos é o algoritmo KNN (MITCHEL, 1997). Este algoritmo assume que todas as instâncias correspondem a pontos no espaço *n*-dimensional. Os vizinhos mais próximos de um exemplo são definidos em termos da norma da distância euclidiana. Mais precisamente, seja uma instância arbitrária *x* descrita pela seguinte relação $a_1(x) + a_2(x) + \cdots + a_3(x)$ em que $a_r(x)$ é o valor do *r*-ésimo atributo da instância *x*. Logo, a distância entre duas instâncias x_i e x_j será definida por $d(x_i, x_j)$, como mostrada na Equação 1.

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r) - a_r(x_j))^2} \quad (1)$$

O número de exemplos a serem comparados com um novo exemplo é dado pelo parâmetro *K* do algoritmo. A Figura 3 ilustra a operação do algoritmo KNN para o caso onde as instâncias estão inseridas em um espaço bidimensional contendo exemplos de classificação utilizando valores de *K* iguais a 1, 2 e 3, figuras (a), (b) e (c) respectivamente. Um novo exemplo *X* é classificado baseado nas classes de seus *K* vizinhos mais próximos.

Figura 3 – Exemplo de classificação utilizando a técnica KNN para *K* igual a 1, 2 e 3



Fonte: Tan, Steinbache e Kumar (2005)

O algoritmo KNN faz suas predições baseado em informações locais. Devido a essa característica, o KNN torna-se uma das técnicas mais suscetíveis a ruído nos dados. Apesar disso, esse algoritmo produz fronteiras de decisão mais arbitrárias, fornecendo um modelo mais flexível que outros algoritmos (CERRI, 2010).

2.2.1.2 Redes neurais artificiais

As redes neurais artificiais (RNA) são inspiradas em redes biológicas neurais - que se baseiam no cérebro humano em sua organização de neurônios e processo de tomada de decisão que são úteis em áreas de aplicação como reconhecimento de padrões, classificação, etc. O processo de tomada de decisão do RNA é mais holística, com base no padrão do total de entrada, enquanto que um computador convencional tem de evoluir através do processamento de elementos de dados individuais para chegar a uma determinada conclusão (HAYKIN, 2000).

As redes neurais derivam seu poder devido a sua estrutura massiva e paralela e a habilidade de aprender por experiência. Essa experiência é transmitida por meio de exemplos obtidos do mundo real, definidos como um conjunto de características que levam a uma determinada situação. Se a situação gerada pela combinação de características for informada a rede, a aprendizagem é dita supervisionada e não supervisionada caso contrário (ACHARYA et al, 2003).

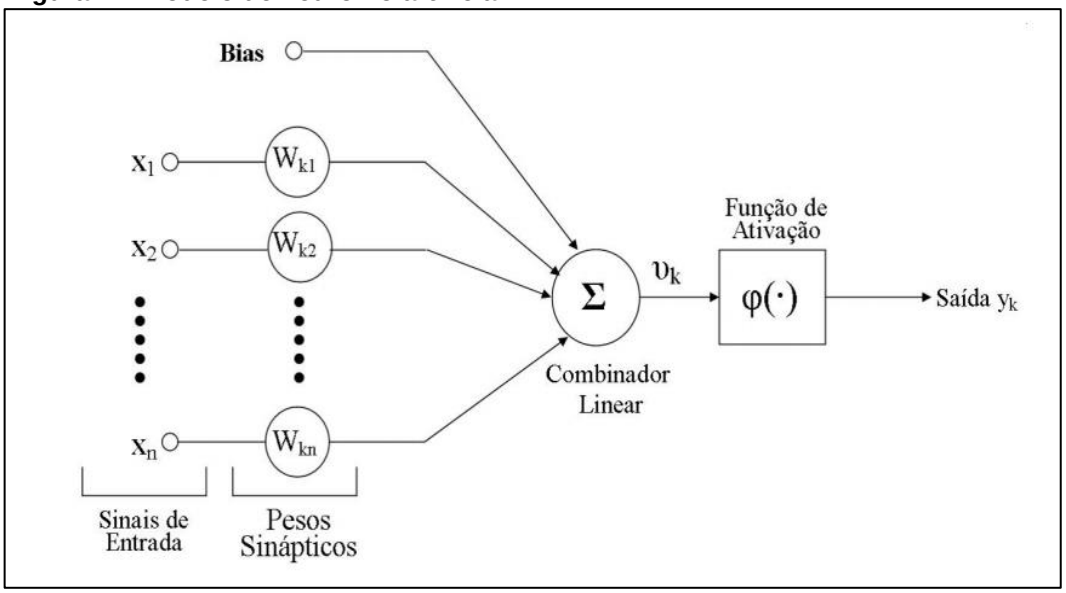
Uma rede neural tradicional é constituída por um conjunto de neurônios interligados, influenciando uns aos outros formando um sistema maior capaz de

armazenar conhecimento adquirido por meio de exemplos apresentados e assim realizando inferências sobre novos exemplos (novas situações) desconhecidos.

A partir da estrutura e funcionamento do neurônio biológico, pesquisadores tentaram simular este modelo em computador. O modelo mais próximo foi proposto por McCulloch e Pitts (1943), conhecido como *Perceptron*, o qual implementa de maneira simplificada os componentes e o funcionamento de um neurônio biológico.

O *Perceptron* é uma rede neural simples: constitui-se de uma camada de entrada e uma camada de saída. A cada entrada existe um peso relacionado, sendo que o valor de saída será a soma dos produtos de cada entrada pelo seu respectivo peso (CARDON; MÜLLER, 1994). A Figura 4 apresenta o modelo de neurônio artificial proposto por McCulloch e Pitts (1943).

Figura 4 – Modelo de neurônio artificial



Fonte: McCulloch e Pitts (1943)

Como Borges (2012) resumiu, esse modelo consiste de:

1. Um conjunto de sinapses, cada uma caracterizada por um peso próprio. Especificamente, um sinal de entrada x_i na entrada da sinapse *j* conectada ao neurônio *k* é multiplicado pelo peso sináptico wk_j;
2. Um combinador linear para somar os sinais de entrada, ponderados pela respectiva sinapse do neurônio;
3. Uma função de ativação para restringir a amplitude da saída de um neurônio. Essa função limita a faixa de amplitude permitida, a qual, normalmente, é limitada ao intervalo fechado de *[0, 1]* ou, alternativamente de *[-1, 1].*

2.2.1.2.1 Perceptron de múltiplas camadas

Uma importante classe de redes neurais é a rede de múltiplas camadas. Tipicamente, a rede consiste de um conjunto de unidades sensoriais (nós de fonte) que constituem a Camada de Entrada, uma ou mais Camadas Ocultas de nós computacionais e uma camada de saída de nós computacionais. O sinal de entrada se propaga para frente através da rede, camada por camada. Estas redes neurais são normalmente chamadas de *Perceptrons* de Múltiplas Camadas (*Multilayer Perceptron* - MLP*)*, as quais representam uma generalização do *perceptron* de camada única (HAYKIN, 2000).

Neste modelo todos os neurônios são ligados aos neurônios da camada subsequente, não havendo ligação com os neurônios laterais (da mesma camada) e também não ocorre realimentação.

Os MLPs têm sido aplicados com sucesso para resolver diversos problemas através do seu treinamento de forma supervisionada com um algoritmo popularmente conhecido como Algoritmo de Retropropagação de Erro (*error back-propagation*). Este algoritmo é baseado na Regra de Aprendizagem por Correção de Erro (HAYKIN, 2000).

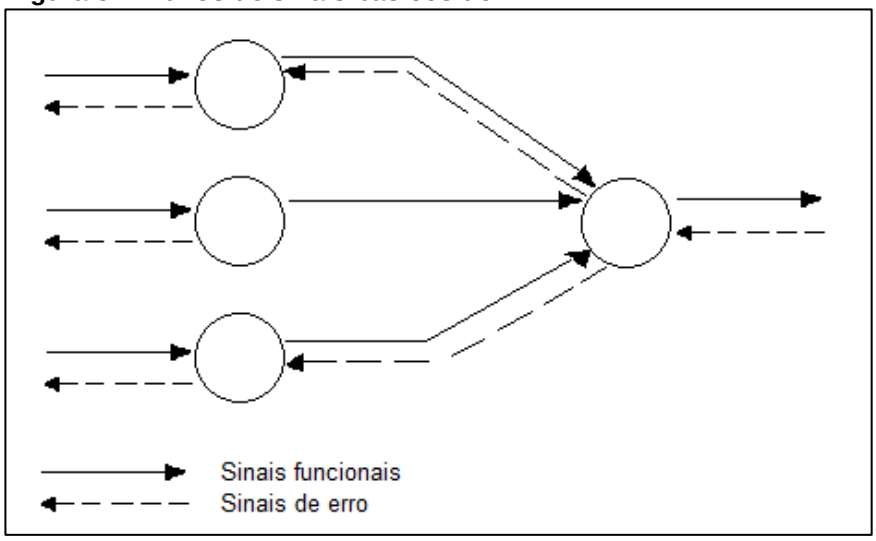
A aprendizagem por retropropagação de erro consiste de dois passos através das diferentes camadas da rede: um passo para frente, a propagação, e um passo para trás, a retropropagação.

No Passo para Frente, um padrão de atividade (vetor de entrada) é aplicado aos nós sensoriais da rede e seu efeito se propaga através da rede, camada por camada. Finalmente, um conjunto de saídas é produzido como a resposta real da rede. Durante o passo de propagação, os pesos sinápticos da rede são todos fixos.

Durante o Passo para Trás, por outro lado, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro. Especificamente, a resposta real da rede é subtraída de uma resposta desejada (alvo) para produzir um sinal de erro. Este sinal de erro é então propagado para trás através da rede, contra a direção das conexões sinápticas – vindo daí o nome de “retropropagação de erro” (*error back-propagation)*.

A Figura 5 ilustra as direções de dois fluxos de sinal básico em um MLP: a propagação para frente de sinais funcionais e a retropropagação de sinais de erro (HAYKIN, 2000).

Figura 5 – Fluxos de sinais básicos do MLP



Fonte: Parker (1987)

2.2.1.3 Árvores de decisão

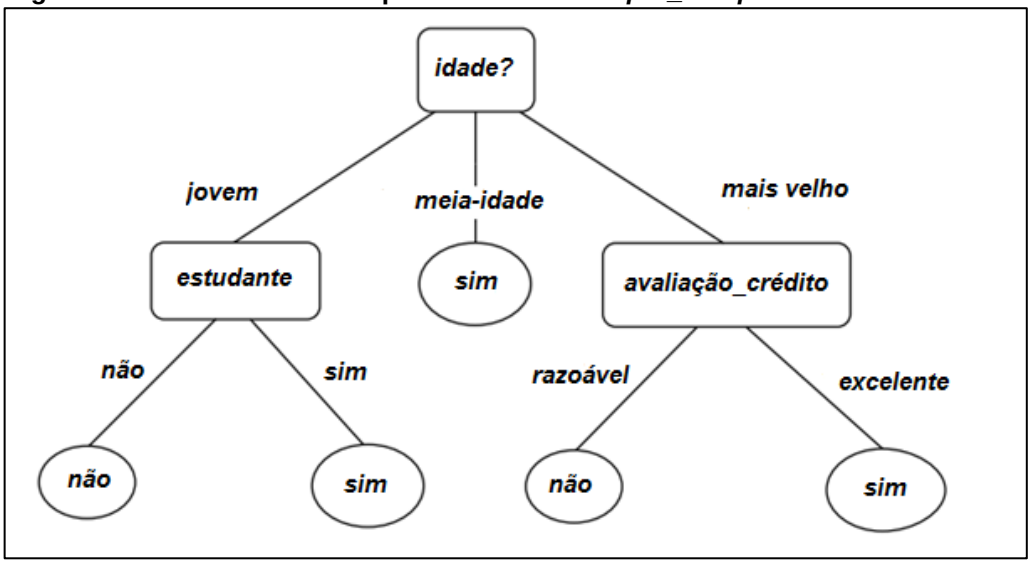
Aprendizagem por Árvores de Decisão (AD) é um método de aproximação de funções por valores discretos, em que a função aprendida é representada por uma estrutura de dados do tipo árvore. Este tipo de aprendizagem também pode ser reapresentado como um conjunto de regras se-então. Estes métodos estão entre os mais populares entre os algoritmos de inferência indutiva e têm sido aplicados com sucesso em uma ampla gama de tarefas de aprendizagem (MITCHEL, 1997).

Em uma AD cada nó interno (não-folha) indica um teste em um atributo, cada ramo representa um resultado do teste, e cada nó folha contém um rótulo de uma classe. O nó superior em uma árvore é o nó raiz (HAN; KAMBER; PEI, 2012).

Um exemplo de uma AD é mostrado na Figura 6 representando um cenário *compra_computador*, isto é, sua função é determinar se um cliente é suscetível a comprar um computador. Os nós internos são representados por retângulos e os nós folhas por formas ovais. Alguns algoritmos de AD produzem apenas árvores binárias

(onde cada ramo dos nós internos deriva exatamente outros dois nós), enquanto outros algoritmos podem produzir árvores não binárias.

Figura 6 – Árvore de Decisão para o cenário *compra_computador*



Fonte: Adaptado de Han, Kamber e Pei (2012)

A classificação em AD é realizada da seguinte maneira: dada uma tupla, *X*, para a qual o rótulo da classe associada é desconhecido, os valores dos atributos da tupla são testados contra a AD. Um caminho é traçado desde a raiz até um nó folha, que possui a classe predita para a tupla. As ADs podem ser facilmente convertidas em regras de classificação.

As etapas de aprendizagem e classificação das ADs são simples e rápidas. Em geral, os classificadores das ADs possuem bom desempenho por gerarem um modelo de classificação baseado em regras bem definidas. Algoritmos de indução de AD são utilizados para classificação em muitas áreas de aplicação, tais como a medicina, fabricação e produção, análise financeira, astronomia e biologia molecular. As ADs são à base de vários sistemas de indução de regras comerciais (HAN; KAMBER; PEI, 2012).

2.3 TÉCNICAS DE CLASSIFICAÇÃO MULTIRRÓTULO

Existe uma grande quantidade de problemas em que alguns exemplos dos dados podem pertencer a mais de uma classe (rótulo) simultaneamente. Esses problemas são conhecidos como classificação multirrótulo (TROHIDIS et al, 2008).

Em um problema de classificação multirrótulo, cada exemplo pode ser associado a duas ou mais classes ao mesmo tempo. Um classificador multirrótulo pode ser definido como uma função $H : x \rightarrow 2^L$, onde L são os rótulos (*label*), que mapeia um exemplo x em um conjunto de classes $C \in 2^L$, em que 2^L é o conjunto potência de L, ou seja, o conjunto formado por todos os subconjuntos de L (CERRI, 2010).

Segundo Silva (2010), um problema de classificação multirrótulo pode estar inserido nos domínios seguintes:

- Classificação semântica de cenas – o objetivo é rotular as imagens de acordo com o seu contexto semântico, por exemplo: “praia”, “nascer do sol” e “neve”. Este tipo de problema tem sido muito utilizado por ferramentas de busca que têm como objetivo retornar para o usuário uma imagem que apresente o conteúdo solicitado. Para realizar essa tarefa, a ferramenta necessita primeiramente rotular as imagens da maneira mais precisa possível, utilizando então um classificador multirrótulo capaz de identificar corretamente o contexto e os objetivos presentes na imagem (BOUTELL et al, 2004).
- Diagnóstico médico – neste contexto, um indivíduo pode possuir mais de uma doença ao mesmo tempo, por exemplo, um paciente pode ser diagnosticado como tendo “enfisema pulmonar” e “bronquite crônica”. Portanto, é de interesse do médico que ambas as doenças sejam corretamente associadas ao paciente, para que o tratamento adequado possa ser aplicado (HUAN et al, 2013).
- Classificação de documento – a classificação multirrótulo pode ser utilizada para identificar o assunto ou contexto de um documento, baseando-se em suas características. Por exemplo, um documento governamental ligado ao ministério da saúde pode pertencer aos rótulos “Governo”, “Saúde” e “SUS”. Em outro exemplo, um artigo científico pode ser rotulado de acordo com as áreas de pesquisa às quais está relacionado, por exemplo, um artigo de uma

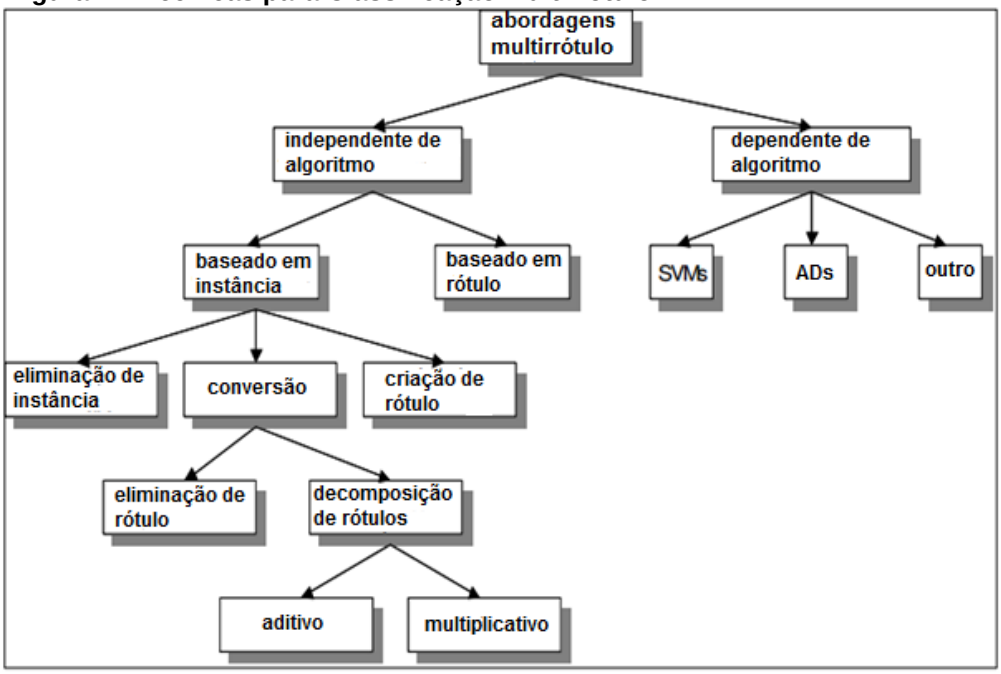
área interdisciplinar pode estar relacionado a várias áreas (rótulos), como “Ciência da Computação” e “Física” (SCHAPIRE; SINGER, 2000).

Diferentes técnicas têm sido propostas na literatura para tratar problemas de classificação multirrótulo. Em algumas dessas técnicas, classificadores simples-rótulo podem ser combinados para tratar problemas de classificação multirrótulo (CERRI, 2010).

A Figura 7 ilustra uma visão geral das diferentes técnicas propostas na literatura para tratar problemas de classificação multirrótulo. De acordo com a figura, essas técnicas pertencem a duas grandes abordagens: abordagem independente de algoritmo e abordagem dependente de algoritmo.

A abordagem independente de algoritmo utiliza algoritmos tradicionais de classificação para tratar problemas multirrótulo, transformando o problema multirrótulo original em um conjunto de problemas simples-rótulo. A abordagem dependente de algoritmo cria algoritmos específicos para tratar o problema multirrótulo. Esses algoritmos podem ser baseados em técnicas de classificação convencionais, como Máquinas de Vetores de Suporte (SVMs) e AD (CERRI, 2010).

Figura 7 – Técnicas para Classificação Multirrótulo



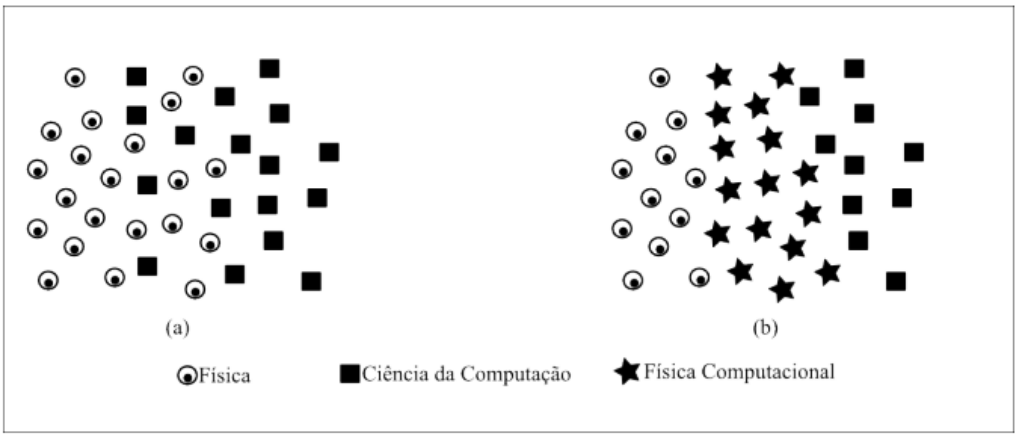
Fonte: Carvalho e Freitas (2009)

2.3.1 Transformação do Problema

Métodos de aprendizagem multirrótulo podem ser agrupados em duas categorias: transformação do problema e adaptação do algoritmo (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010).

O primeiro grupo de métodos são algoritmos independentes – abordagem independente de algoritmo. Eles transformam a tarefa de aprendizagem em um ou mais problemas de classificação simples (classificação plana), para o qual existe um grande grupo de algoritmos de aprendizagem. A Figura 8 mostra um exemplo dessa abordagem em que (a) é ilustrado um problema de classificação no qual um documento pode pertencer à classe “Física” ou à classe “Ciência da Computação”, mas nunca as duas ao mesmo tempo, enquanto (b) ilustra um exemplo de classificação multirrótulo, em que os documentos pertencentes simultaneamente às duas classes são classificados como “Física Computacional”.

Figura 8 – (a) Típico problema de classificação. (b) Problema de classificação multirrótulo



Fonte: Cerri (2010)

O segundo grupo de métodos estende algoritmos de aprendizagem específicos – abordagem dependente de algoritmo. Estes algoritmos utilizam aprendizado através de Árvores de Decisão, classificadores KNN, Redes Neurais Artificiais (RNA), Máquinas de Vetores de Suporte (SVMs), Algoritmos Genéticos (AG) e outros.

O Quadro 1 apresenta um conjunto de dados multirrótulo que consiste de quatro exemplos contendo uma ou mais classes: λ_1, λ_2, λ_3, λ_4. Como a

transformação afeta apenas o rótulo, para os quadros seguintes, contendo as etapas de transformação (Figura 9), serão desconsiderados os atributos por simplicidade de apresentação.

Quadro 1 – Conjunto de dados multirrótulo

Exemplo	Atributos	Rótulo
1	$\boldsymbol{x_1}$	$\{\lambda_1, \lambda_4\}$
2	$\boldsymbol{x_2}$	$\{\lambda_3, \lambda_4\}$
3	$\boldsymbol{x_3}$	$\{\lambda_1\}$
4	$\boldsymbol{x_4}$	$\{\lambda_2, \lambda_3, \lambda_4\}$

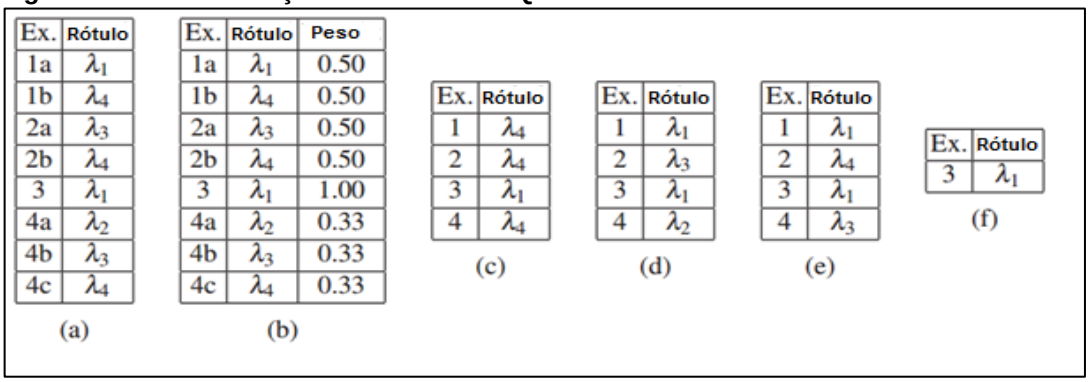
Fonte: Tsoumakas, Katakis e Vlahavas (2010)

Existem muitos métodos de transformações simples que podem ser usados para converter exemplos de entrada multirrótulo em simples-rótulo (classificação plana) (BOUTELL et al, 2004).

Um classificador simples-rótulo que apresenta o resultado através de distribuição de probabilidade sobre todas as classes pode ser utilizado na aprendizagem das posições das classes. A classe com a maior probabilidade terá a primeira posição, a classe com a segunda maior probabilidade terá a segunda posição, e assim por diante.

A transformação cópia (*copy*) substitui cada exemplo multirrótulo (x_i, Y_i) com $|Y_i|$ exemplos (x_i, λ_j), para cada $\lambda_j \in Y_i$. Uma variação dessa transformação, chamada cópia por peso (*copy-weight*), associa um peso de $\frac{1}{|Y_i|}$ para cada um dos exemplos produzidos. A etapa de seleção (*select*) da transformação substitui Y_i com um de seus membros. Esse rótulo pode ser mais (*select-max*) ou menos (*select-min*) frequente entre todos os exemplos. Também pode ser selecionado aleatoriamente (*select-random*). Finalmente, a transformação ignorada (*ignore*) simplesmente descarta exemplos de entrada multirrótulo (TSOUMAKAS, KATAKIS; VLAHAVAS, 2010). A Figura 9 mostra os dados transformados utilizando os métodos descritos – (a) cópia, (b) cópia por peso, (c) seleção máxima, (d) seleção mínima, (e) seleção aleatória e (f) ignorada.

Figura 9 – Transformação dos dados do Quadro 1



Ex.	Rótulo
1a	λ_1
1b	λ_4
2a	λ_3
2b	λ_4
3	λ_1
4a	λ_2
4b	λ_3
4c	λ_4

(a)

Ex.	Rótulo	Peso
1a	λ_1	0.50
1b	λ_4	0.50
2a	λ_3	0.50
2b	λ_4	0.50
3	λ_1	1.00
4a	λ_2	0.33
4b	λ_3	0.33
4c	λ_4	0.33

(b)

Ex.	Rótulo
1	λ_4
2	λ_4
3	λ_1
4	λ_4

(c)

Ex.	Rótulo
1	λ_1
2	λ_3
3	λ_1
4	λ_2

(d)

Ex.	Rótulo
1	λ_1
2	λ_4
3	λ_1
4	λ_3

(e)

Ex.	Rótulo
3	λ_1

(f)

Fonte: Tsoumakas, Katakis e Vlahavas (2010)

2.3.1.1 Abordagem independente de algoritmo

Os métodos de classificação multirrótulo independentes de algoritmos podem ser utilizados com qualquer algoritmo de classificação. Nesta abordagem, um problema de classificação multirrótulo é normalmente transformado em um conjunto de problemas simples-rótulo. Esta transformação pode ser tanto baseada nos rótulos de classes quanto nos próprios exemplos de treinamento (CARVALHO; FREITAS, 2009).

A Tabela 1 apresenta uma comparação dessas técnicas, em que *L* representa o número de classes, l_i representa o número de classes que rotulam o *i*-ésimo exemplo e *K* representa o número de classes do conjunto de dados que rotulam pelo menos um exemplo multirrótulo. Pode-se notar, pela tabela, que as técnicas diferem principalmente quanto à reversibilidade, ao número de classificadores utilizados, e ao tamanho do conjunto de dados após sua transformação (CERRI, 2010).

Tabela 1 – Comparação das técnicas independentes de algoritmo

Técnica de Transformação	Reversibilidade	N. Classificadores	N. Exemplos
Baseada nos rótulos	SIM	*L*	Se mantém
Eliminação de exemplos	NÃO	Se mantém	Reduz
Criação de rótulos	SIM	Se mantém	Se mantém
Simplificação de rótulos	Depende do critério	Se mantém	Se mantém
Decomp.: método aditivo	SIM	*K*	Aumenta
Decomp.: método multiplicativo	SIM	Πl_i	Aumenta

Fonte: Cerri (2010)

2.3.1.2 Abordagem dependente de algoritmo

Como o nome desta abordagem sugere, os métodos são baseados em algoritmos específicos. A vantagem desta abordagem é que, concentrando-se em apenas um algoritmo, o método pode apresentar melhor desempenho em problemas do mundo real do que abordagens independentes de algoritmos (CARVALHO; FREITAS, 2009).

As abordagens dependentes de algoritmos mais comuns para classificação multirrótulo relacionadas com este trabalho são: uma extensão alternativa de AD, chamada Árvore de Decisão Alternada, e uma técnica do KNN chamada *Multi-label* KNN (ML-KNN). As próximas seções serão dedicadas para conceituar brevemente as duas técnicas tidas como dependentes de algoritmos.

2.3.1.2.1 Árvores de decisão alternada

Uma extensão alternativa do algoritmo de AD para problemas de classificação multirrótulo foi proposto em Comité et al (2003), chamada de “Árvore de Decisão Alternada” (*Alternating Decision Tree* (ADT)). Essa técnica é uma generalização das ADs e seu princípio indutivo é baseado no método *boosting* de Freund e Schapire (1995). Esse algoritmo estende o ADT pela decomposição de problemas multiclasse usando a abordagem um-contra-todos (CARVALHO; FREITAS, 2009).

Outro trabalho que utiliza ADs foi proposto por Clare e King (2001). Neste, os autores modificaram o algoritmo C4.5 de Quinlan (1993) para a classificação de proteínas de acordo com suas funções. O algoritmo C4.5 define os nós da AD através de uma medida chamada entropia, que pode ser definida como sendo o grau de pureza desse conjunto de nós. Os autores modificaram a fórmula dessa medida, originalmente elaborada para problemas simples-rótulo, de maneira a permitir seu uso em problemas multirrótulo. Outra modificação feita pelos autores foi à utilização dos nós-folha da árvore para representar conjuntos de rótulos de classes. Quando um nó-folha, alcançado na classificação de um exemplo, contém um conjunto de

classes, uma regra separada é produzida para cada classe (CARVALHO; FREITAS, 2009).

2.3.1.2.2 ML-KNN

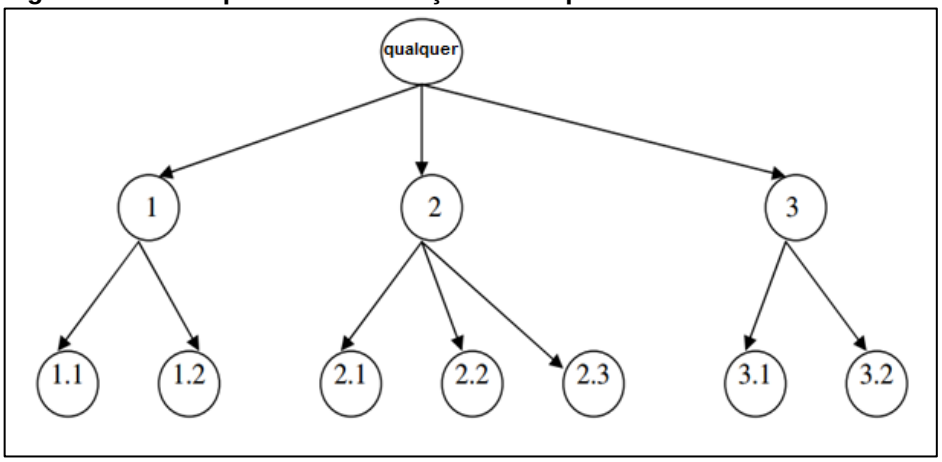
Em Zhang e Zhou (2005) é proposta uma nova técnica para classificação multirrótulo baseada no algoritmo KNN, chamada ML-KNN (*Multi-label* KNN). Nessa técnica, para cada exemplo, as classes associadas com os *K* exemplos vizinhos mais próximos são recuperadas, e é feita uma contagem dos vizinhos associados a cada classe. Então, o princípio *maximum a posteriori* Saridis (1983) é utilizado para definir o conjunto de classes de um novo exemplo (CERRI, 2010).

2.4 TÉCNICAS DE CLASSIFICAÇÃO HIERÁRQUICA

A grande maioria dos problemas de classificação contidos na literatura é referente à classificação plana, em que para cada exemplo é atribuída uma classe de um conjunto finito (e geralmente pequeno) de classes. Em contrapartida, em problemas de classificação hierárquica, as classes são dispostas em uma estrutura hierárquica, tal como uma árvore ou um Grafo Acíclico Direcionado (DAG) (CARVALHO; FREITAS, 2007).

Na estrutura em árvore, cada nó é rotulado com o número (id) da sua classe correspondente. Considera-se o nó raiz no nível zero, e o nível de qualquer outro nó é determinado pelo número de arestas que ligam o nó ao nó raiz. Nós no primeiro nível tem apenas um dígito, enquanto nós no segundo nível tem dois dígitos: o primeiro dígito identifica a classe pai (no primeiro nível) e o segundo dígito identifica a subclasse no segundo nível. Na Figura 10 é ilustrada uma estrutura do tipo árvore, onde cada nó representa uma classe – identificada pelo número dentro do nó – e as arestas entre os nós representam relações de superclasse ou subclasse.

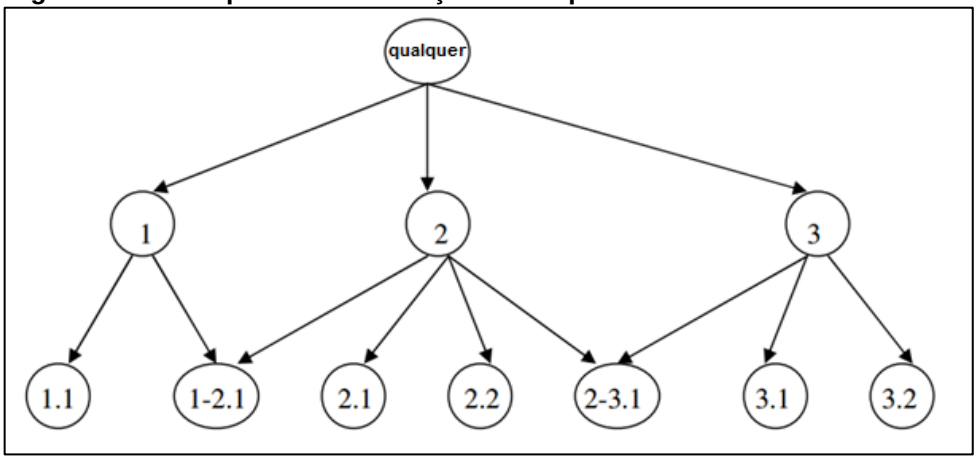
Figura 10 – Exemplo de classificação hierárquica utilizando estrutura árvore



Fonte: Carvalho e Freitas (2007)

Na estrutura DAG, um nó pode ter mais de um pai. Assim, modelos em níveis mais profundos podem ser induzidos com um número maior de exemplos de treinamento do que seus nós pais. Apesar disso, na prática, mesmo para DAGs, a precisão na predição decresce com o aumento da profundidade (CERRI, 2010). Na Figura 11, a representação das classes pai são especificadas por dois dígitos antes do delimitador do nível de classe ".", isto é, um dígito para cada classe pai.

Figura 11 – Exemplo de classificação hierárquica utilizando estrutura DAG



Fonte: Carvalho e Freitas (2007)

Nas técnicas hierárquicas de classificação, o algoritmo de aprendizado induz um classificador que captura os relacionamentos mais relevantes entre as classes funcionais no conjunto de dados de treinamento, considerando os relacionamentos hierárquicos entre as classes. Dessa maneira, pode ser utilizada uma grande variedade de algoritmos de AM no processo de classificação (CARVALHO; FREITAS, 2007).

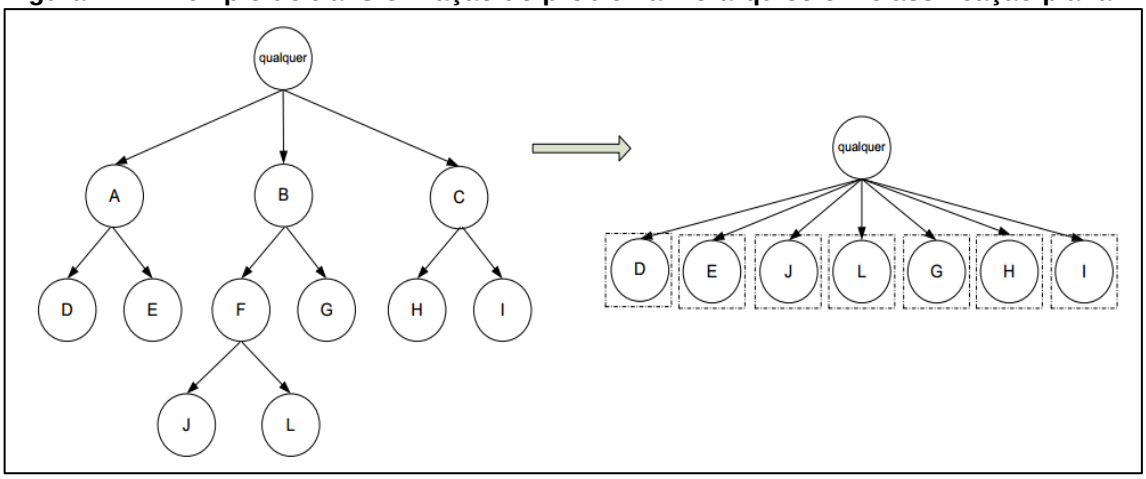
2.4.1 Abordagens para Tratar Problemas de Classificação Hierárquica

Nos últimos anos, algumas soluções têm sido propostas para a indução de modelos de classificação para problemas hierárquicos (COSTA, 2008). Segundo Carvalho e Freitas (2007), quatro abordagens podem ser utilizadas para tais soluções de classificação hierárquica: transformação do problema hierárquico em um problema de classificação plana, predição hierárquica utilizando algoritmos de classificação plana, classificação *Top-Down* e classificação *Big-Bang*.

Para o desenvolvimento deste trabalho serão utilizadas a transformação do problema hierárquico em problema de classificação plana e a classificação *Top-Down*, onde as classes de cada exemplo são representadas por um vetor binário, no qual cada posição corresponde a uma classe.

A classificação hierárquica plana tem o mesmo comportamento de um algoritmo de classificação convencional na fase de treinamento e teste. Esta abordagem considera que um problema de classificação hierárquica pode ser transformado em um problema de classificação plana, como ilustra a Figura 12, desconsiderando o conceito de ancestral e descendente, ou seja, ignora-se a hierarquia de classe, predizendo apenas os nós-folha. Esta técnica é parecida com a classificação plana convencional e pode ser aplicada em estruturas do tipo árvore e do tipo DAG (BORGES, 2012).

Figura 12 – Exemplo de transformação do problema hierárquico em classificação plana



Fonte: Borges (2012)

2.5 RESUMO DO CAPÍTULO

Foram apresentados neste capítulo os conceitos considerados importantes para o desenvolvimento deste trabalho. Primeiramente, foram introduzidos os principais conceitos contidos na literatura de mineração de dados e aprendizagem de máquina. Na segunda seção, foram explicadas as técnicas de classificação, contendo uma subseção para as técnicas e paradigmas de classificação. Em seguida, na seção técnicas de classificação multirrótulo, foram abordadas as técnicas para transformação do problema e as abordagens existentes na literatura. Foram descritos, na seção seguinte, as técnicas de classificação hierárquica, bem como as principais estruturas para representá-las. Na quinta seção, foi apresentado um trabalho relacionado com o tema deste trabalho que tratava do mesmo problema da previsão da quantidade de classes em classificação hierárquica multirrótulo.

3 METODOLOGIA

Este capítulo descreve a metodologia utilizada para o desenvolvimento deste trabalho. As etapas principais da metodologia são pré-processamento, classificação, análise comparativa e avaliação dos resultados. Na seção 3.1, serão descritas as características das bases de dados utilizadas nos experimentos. Na seção 3.2, será apresentada a ferramenta utilizada e os formatos das bases de dados por ela aceitos. Na seção 3.3, será descrita a metodologia para realização dos experimentos. E por fim, na seção 3.4, serão descritas as considerações finais do capítulo.

3.1 BASES DE DADOS

As bases de dados utilizadas nos experimentos deste trabalho são do campo da genômica funcional (*functional genomics*), conhecidas por *Gene Ontology Project*[1] (GO). Esse projeto fornece bases de dados estruturadas para classificação de diversos domínios da biologia molecular e celular (ASHBURNER, 2000).

Foram utilizadas duas bases de dados biológicos de expressão gênica para os experimentos: *Cellcycle* (SPELLMAN *et al*, 1998) e *Church* (ROTH *et al, 1998*).

Detalhes como, quantidade de amostras (instâncias), quantidade de atributos, quantidade de classes e tipos de dados são apresentados na Tabela 2.

Tabela 2 – Características gerais das Bases de Dados

Base de Dados	Quant. Amostras	Quant. Atributos	Quant. Classes	Tipos de Dados
Cellcycle	3751	77	4125	Atributos numéricos e alguns faltantes.
Church	3749	27	4125	Atributos numéricos, um categórico e alguns faltantes.

Fonte: Borges (2012)

[1] http://www.geneontology.org/

3.2 FERRAMENTAS UTILIZADAS

A ferramenta utilizada para pré-processamento, classificação, experimento e avaliação dos resultados foi o *framework* de mineração de dados Weka (*The Weka Data Mining Software in Java*) (HALL et al, 2009).

Weka é uma coleção de algoritmos de aprendizagem de máquina para realização de tarefas de DM. Os algoritmos podem ser diretamente aplicados a bases de dados ou serem instanciados de uma aplicação Java. O Weka possui ferramentas para pré-processamento, classificação, regressão, agrupamento, regras de associação e visualização. Este *framework* também possui o *Experimenter* que é um método da ferramenta capaz de comparar resultados de diferentes algoritmos de aprendizagem (BOUCKAERT, 2016).

3.2.1 Especificação das Configurações

Os experimentos foram realizados utilizando a versão 3.8 do Weka, que é a versão mais recente da ferramenta. Os tópicos seguintes especificam as configurações da máquina:

- Processador: Intel® Core™ i7-3770K CPU @ 3.50GHz 3.90GHz
- Memória RAM: 8,00 GB
- Sistema Operacional: Windows 7 Professional SP1 – 64 Bits

3.2.2 O Formato ARFF

Para que o Weka leia uma base de dados, a mesma deve estar em formato ARFF (*Attribute-Relation File Format*). Um arquivo ARFF é um arquivo de texto ASCII que descreve uma lista de instâncias que compartilham um conjunto de atributos.

Arquivos ARFF possuem duas diferentes sessões. A primeira sessão contém o cabeçalho de informações e a segunda contém as informações dos dados (instâncias). O cabeçalho de informações possui o nome da relação e uma lista de

atributos seguidos de seus tipos. Na segunda sessão é listado o conjunto de dados contendo todas as instâncias com seus respectivos atributos e classes.

As Figuras 13 (a) e (b) ilustram o cabeçalho de informações e os dados, respectivamente, de um exemplo da base de dados IRIS (FISHER, 1936). Esta base contém 50 exemplos de cada espécie da flor Iris (*Iris Setosa, Iris Virginica, Iris Versicolor*). Há quatro atributos que caracterizam cada exemplo: comprimento e largura das sépalas e das pétalas em centímetros.

Figura 13 – Exemplo de um arquivo ARFF - IRIS

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

(a) – Cabeçalho de um arquivo ARFF - IRIS

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

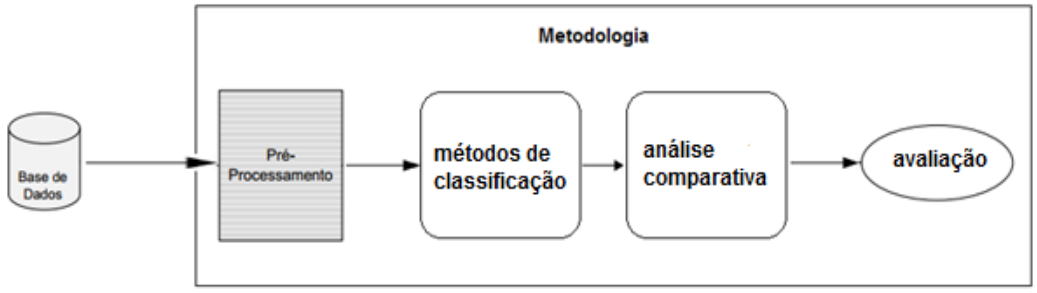
(b) – Dados de um arquivo ARFF - IRIS

Fonte: Bouckaert (2016)

3.3 METODOLOGIA PARA REALIZAÇÃO DOS EXPERIMENTOS

A metodologia utilizada para a realização dos experimentos é formada pelas seguintes etapas: pré-processamento, métodos de classificação, análise comparativa e avaliação dos resultados. A Figura 14 ilustra a sequência das etapas descritas.

Figura 14 – Metodologia para realização dos experimentos



Fonte: adaptado de Borges (2012)

3.3.1 Pré-processamento

As bases GO utilizadas possuem um padrão de classificação hierárquico e multirrótulo. As classes estão dispostas em uma árvore de decisão à qual possui um determinado nível medido entre sua raiz e o nó folha.

As instâncias podem pertencer a múltiplas classes simultaneamente e essas classes estarão organizadas em uma hierarquia. A hierarquia representa uma dependência entre as classes.

Quando os dados são formatados no formato ARFF, as classes atribuídas a um exemplo são colocadas no formato “1.2.4.6@3.7.9.10”, que indica que o exemplo pertence a duas classes: “1.2.4.6” e “3.7.9.10”. Ou seja, as classes são separadas por uma “@”. Os níveis hierárquicos são separados por um ponto. Assim, a classe “1” é uma superclasse de “2”, que por sua vez é superclasse de “4”, que é superclasse de “6” (CERRI, 2010).

Para que fosse possível fazer os experimentos com as bases GO no Weka, foi utilizada a abordagem *Top-Down*, onde as classes de cada exemplo são representadas por um vetor binário, no qual cada posição corresponde a uma classe. A *i*-ésima posição corresponde a *i*-ésima classe da hierarquia e recebe o valor 1 se o exemplo pertence à classe e 0 caso contrário (CERRI, 2010). Para tanto, foi desenvolvida uma aplicação em Java, que a partir da leitura das bases, gerasse novas bases seguindo o padrão descrito anteriormente. Essa aplicação será mais bem detalhada na subseção seguinte.

As figuras a seguir ilustram a transformação hierárquica. A Figura 15 (a) mostra uma instância da base de dados *Cellcycle* seguindo o padrão ARFF contendo seus atributos numéricos seguidos por sua hierarquia de classes. A Figura

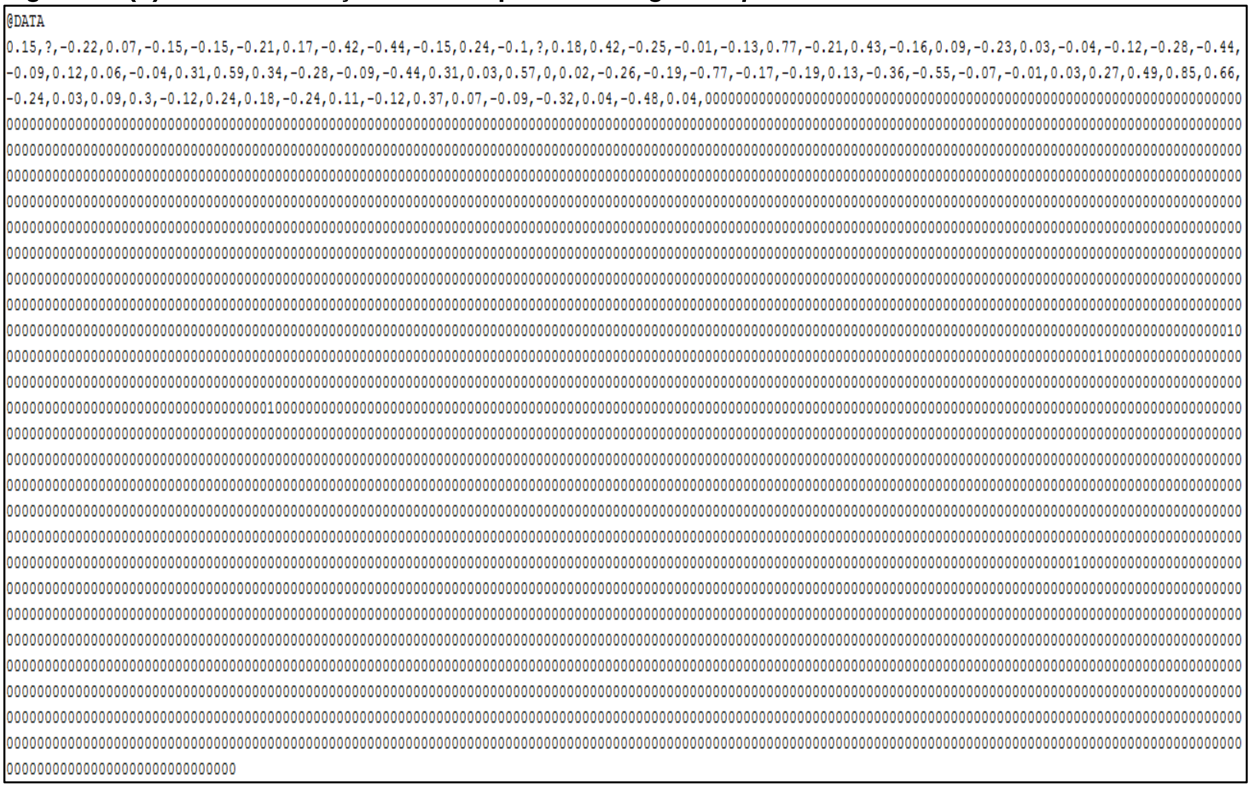
15 (b) mostra a transformação dessa mesma instância utilizando a abordagem *Top-Down*.

Figura 15 (a) – Instância da base de dados *Cellcycle* original

```
@DATA
0.15,?,-0.22,0.07,-0.15,-0.15,-0.21,0.17,-0.42,-0.44,-0.15,0.24,-0.1,?,0.18,0.42,-0.25,-0.01,-0.13,0.77,-0.21,0.43,-0.16,0.09,-0.23,0.03,-0.04,
-0.12,-0.28,-0.44,-0.09,0.12,0.06,-0.04,0.31,0.59,0.34,-0.28,-0.09,-0.44,0.31,0.03,0.57,0,0.02,-0.26,-0.19,-0.77,-0.17,-0.19,0.13,-0.36,-0.55,-
0.07,-0.01,0.03,0.27,0.49,0.85,0.66,-0.24,0.03,0.09,0.3,-0.12,0.24,0.18,-0.24,0.11,-0.12,0.37,0.07,-0.09,-0.32,0.04,-0.48,0.04,GO0003709@GO0000
127@GO0005739@GO0006384
```

Fonte: Autoria própria

Figura 15 (b) – Transformação da base pela abordagem *Top-Down*



Fonte: Autoria própria

3.3.1.1Aplicação para transformação das bases de dados

A transformação das bases de dados do formato hierárquico multirrótulo para o formato plano e simples-rótulo pelo método *Top-Down*, foi realizada através de uma aplicação desenvolvida em linguagem Java.

Para cada base de dados foram criadas duas classes. A primeira é uma classe aplicação contendo os métodos *scanner* para leitura da base no padrão ARFF e para leitura de um arquivo .txt contendo os 4125 diferentes rótulos sem

repetição. Nessa mesma classe, também são gerados dois arquivos de saída, um arquivo contendo todas as classes das instâncias das bases de dados transformadas para o padrão plano e simples-rótulo, e o segundo arquivo de saída que será a base de dados completa no formato ARFF transformada de acordo com a descrição do método *Top-Down*.

A segunda classe da aplicação faz uma varredura no *array list* onde estão armazenadas todas as classes para excluir qualquer uma que esteja repetida da estrutura. Após a varredura, é gerado um arquivo contendo todas as classes sem repetição.

3.3.2 Métodos de Classificação

Os métodos de classificação selecionados para realização dos experimentos nas bases GO, *Cellcycle* e *Church*, foram J48 (QUINLAN, 1993) como Árvore de Decisão, MLP como rede neural e o KNN como método de aprendizagem baseado em instâncias ou exemplos (*Instance-based learning algorithms* - IBK) (AHA, KIBLER, ALBERT, 1991). O Quadro 2 apresenta os métodos de classificação contendo seus tipos de classificadores e tipos de classes.

Quadro 2 – Métodos de classificação

Método	Tipo de classificador	Tipos de classe
J48	Baseado em árvore de decisão	Binária e nominal
MLP	Baseado em função	Numérica, binária e nominal
KNN	Baseado em exemplos (preguiçoso)	Numérica, binária e nominal

Fonte: Autoria própria

Os três métodos foram aplicados em bases de treinamento para gerarem o tipo de teste de validação utilizada, o modelo do classificador, a taxa de acerto e de erro de classificação das instâncias.

3.3.3 Análise Comparativa

A etapa de análise comparativa avalia os resultados dos três métodos utilizados para classificação nas bases de treinamento. Esta comparação será feita através da opção *Experimenter* do Weka, a qual possibilita analisar os resultados de diferentes métodos de classificação em uma única execução.

Os resultados obtidos serão: a porcentagem de precisão correta para cada método, o resultado estatístico do método que obteve maior e menor taxa de acerto de predição de classes e a margem de predição dos métodos.

3.3.4 Avaliação dos Métodos

Uma vez treinados, os classificadores precisam ser avaliados quanto à capacidade de predizer a classe de novos exemplos (CERRI, 2010). Esta etapa fará a avaliação de classificadores.

Os métodos de avaliação podem ser divididos em dois conjuntos: métodos de resubstituição, os quais utilizam os mesmo exemplos da fase de treinamento para o teste do classificador, e os métodos de reamostragem, que fazem distinção entre o conjunto de treinamento e o conjunto de teste do classificador.

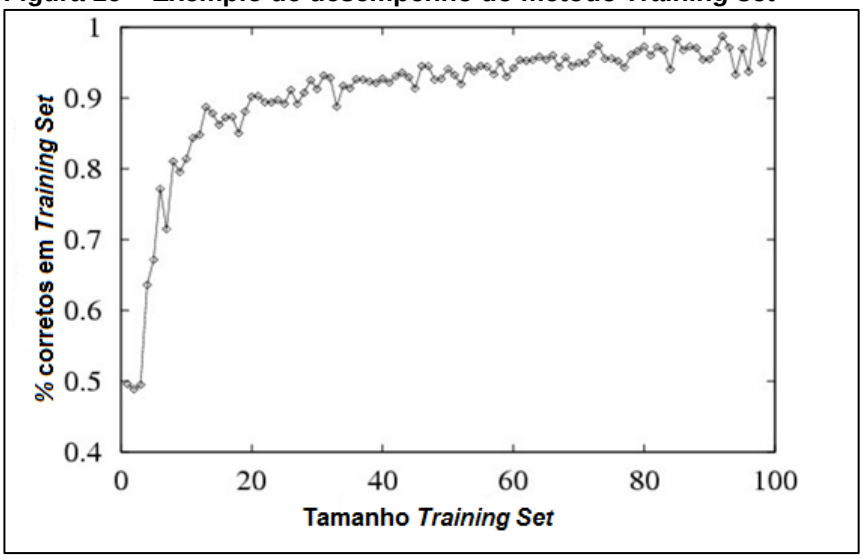
A ferramenta Weka oferece quatro opções para aplicação dos testes entre classe e classe predita. A seguir são descritas as funcionalidades de cada uma delas.

- *Training set*: o classificador é avaliado em o quão bem ele prevê a classe das instâncias em que foi treinado;
- *Supplied test set*: o classificador é avaliado em o quão bem ele prevê a classe de um conjunto de instâncias carregadas de um arquivo;
- *Cross-validation*: o classificador é avaliado por validação cruzada, que é um método estatístico de avaliar e comparar algoritmos de aprendizagem dividindo os dados em dois conjuntos mutualmente exclusivos: um usado para aprender ou treinar um modelo e outro usado para validar o modelo;
- *Percentage split*: o classificador é avaliado em o quão bem ele prevê uma determinada porcentagem dos dados utilizados para teste.

Para a avaliação de classificadores deste trabalho será utilizado o método *training set*. Como as bases utilizadas para os experimentos serão as de treinamento, este método, se comparado aos outros, é o que apresenta melhor resultado da taxa de instâncias classificadas corretamente, portanto, torna-se mais preciso para avaliação dos resultados.

A Figura 16 apresenta um gráfico contendo a relação de tamanho do conjunto de treinamento por porcentagem de instâncias classificadas corretamente pelo método *training set*. O gráfico mostra que quanto maior o número de instâncias treinadas maior será a porcentagem de acerto na predição da classe correta, ou seja, existe uma relação de proporção entre elas.

Figura 16 – Exemplo de desempenho do método *Training set*



Fonte: Goldberg et al (2009)

3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentadas as etapas que compõem a metodologia para o desenvolvimento do presente trabalho. Essas etapas são interdependentes e devem seguir os passos descritos na Figura 14. Na primeira seção foram descritas as características das bases de dados utilizadas para os experimentos. Em seguida, foi descrita a ferramenta utilizada para aplicar as técnicas classificação e obtenção dos resultados. Nessa mesma seção, foram descritos os detalhes das extensões e as características que os arquivos devem seguir para serem lidos pela ferramenta.

Na terceira seção são explicadas cada etapa da metodologia, as quais são pré-processamento, métodos de classificação, análise comparativa e avaliação dos métodos.

No próximo capítulo serão descritos os resultados dos experimentos através da análise de resultados dos classificadores. Além dos resultados, haverá uma sessão contendo uma análise comparativa e a validação dos métodos através das taxas de acerto e erro na predição de classes.

4 REALIZAÇÃO DOS EXPERIMENTOS E ANÁLISE DE RESULTADOS

Neste capítulo serão apresentados os resultados dos experimentos realizados e as análises comparativas entre os métodos utilizados. Esses resultados, mostrados na primeira seção, foram obtidos através da ferramenta Weka utilizando duas bases de dados GO após serem pré-processadas. Esse pré-processamento teve como objetivo transformar as bases do formato hierárquico e multirrótulo para o formato não-hierárquico (plano) e simples-rótulo. Os experimentos realizados foram feitos utilizando os seguintes algoritmos: J48, KNN e MLP. Na segunda seção do capítulo, é realizada a análise comparativa dos métodos de classificação através das taxas de acerto e erro da margem de predição dos classificadores.

4.1 RESULTADOS DOS MÉTODOS DE CLASSIFICAÇÃO

Para se obter a taxa de acerto e/ou taxa de erro de um algoritmo é necessário medir a qualidade da classificação. Essas medidas são calculadas a partir dos exemplos que foram classificados correta e incorretamente, os quais são armazenados em uma matriz, denominada matriz de confusão, como mostrada no exemplo da Tabela 3 por (BORGES, 2012).

Quadro 3 – Matriz de confusão

	Classe Predita	
Classe Verdadeira	**Positiva**	**Negativa**
Positiva	VP	FN
Negativa	FP	VN

Fonte: Borges (2012)

Em um problema de classificação binária convencional quatro situações podem ocorrer:

1. Verdadeiro Positivo (VP): O exemplo é predito corretamente como pertencente à classe positiva;
2. Falso Positivo (FP): O exemplo é predito como pertencente à classe positiva, mas na realidade pertence à classe negativa;

3. Verdadeiro Negativo (VN): O exemplo é predito corretamente como pertencente à classe negativa;
4. Falso negativo (FN): O exemplo é predito como pertencente à classe negativa, mas pertence à classe positiva.

Os resultados obtidos a partir da execução das técnicas descritas anteriormente foram realizados considerando-se as médias das seguintes informações:

- Taxa de verdadeiros positivos (*TP Rate*): casos corretamente classificados como uma determinada classe;
- Taxa de falsos positivos (*FP Rate*): casos incorretamente classificados como uma determinada classe;
- Precisão (P): proporção de casos que são verdadeiramente de uma classe dividido pelo total de casos classificados como essa classe. A medida de precisão calcula a probabilidade de a predição positiva estar correta em relação a todas as amostras;

$$P = \frac{|VP|}{|VP| + |FP|} \tag{2}$$

- Revocação (R) (*Recall)*: proporção de casos classificados como uma determinada classe, dividido pelo total real nessa classe. Indica quantos exemplos positivos foram previstos do total de exemplos;

$$R = \frac{|VP|}{|VP| + |FN|} \tag{3}$$

- *F-Measure* (F_m): a medida entre a relação precisão e revocação calculada como

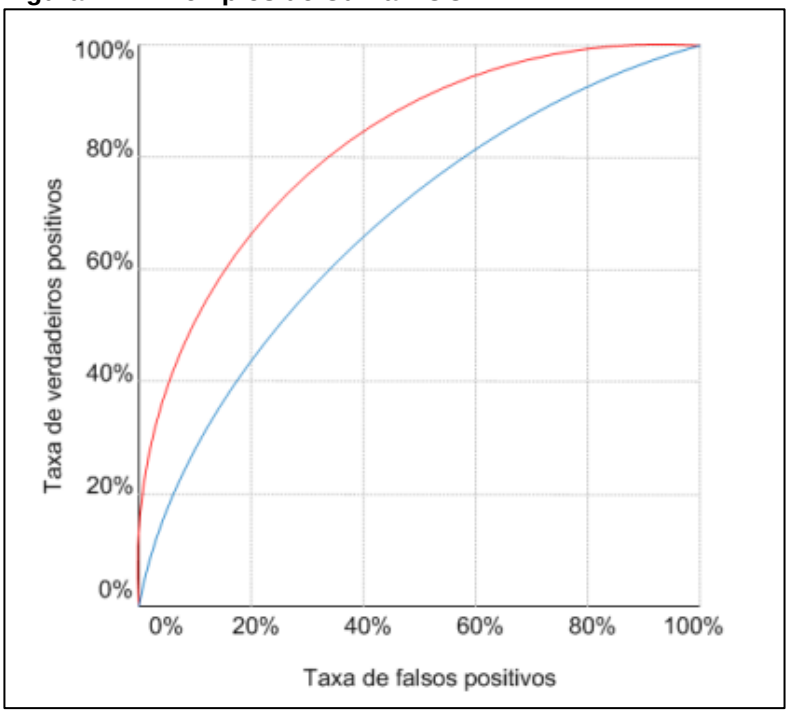
$$F_m = \frac{2 * P * R}{P + R} \tag{4}$$

sendo P o valor da proporção de Precisão e R o valor da proporção de revocação;

- Área sobre a curva ROC (*Receiver Operating Characteristic curve* - curva da Característica de Operação do Receptor) (*ROC Area)*: medição da área obtida através da fração de Verdadeiros Positivos (*TP Rate*) pelos Positivos Totais *versus* a fração de Falsos Positivos (*FP Rate*) pelos Negativos Totais.

A informação mais precisa sobre os resultados citados é a da área sobre a curva ROC, pois como Hand (2009) resumiu, esta é a principal medida de desempenho para os classificadores. Ela utiliza uma representação gráfica que ilustra o desempenho de um sistema de classificação e como seu limite de diferenciação é variado. Essa representação gráfica reproduz a taxa de verdadeiros positivos contra a taxa de falsos positivos para os diferentes pontos possíveis da fase de teste. A Figura 17 mostra um exemplo de Curva ROC.

Figura 17 – Exemplos de Curva ROC



Fonte: Prati et al (2008)

Na Figura 17, é apresentado um gráfico no qual a área abaixo da curva ROC é chamada AUC (*Area Under the ROC Curve*). Essa área corresponde a uma medida quantitativa usada na comparação entre dois classificadores. Quanto mais próximo de 1 o valor da AUC, melhor é o desempenho do classificador.

4.1.1 Resultados Obtidos pelo Método KNN

A aplicação do KNN foi realizada considerando valor de k igual a 1, ou seja, o modelo de classificação se deu baseado no exemplo do primeiro vizinho mais próximo de acordo com as características de cada instância. A Tabela 3 mostra a relação da média dos resultados obtidos entre a técnica de classificação KNN e as bases de dados.

Tabela 3 – Média dos resultados obtidos pelo KNN

Base de Dados	*TP Rate*	*FP Rate*	Precisão	*Recall*	*F-Measure*	*ROC Area*
Cellcycle	0.170	0.001	0.146	0.170	0.139	0.585
Church	0.312	0.004	0.281	0.312	0.266	0.886

Fonte: Autoria própria

4.1.2 Resultados Obtidos pelo Método de Árvore de Decisão - J48

O modelo da árvore de decisão, baseada no algoritmo C4.5, foi gerado sobre um valor de confiança de 0.25 considerando que cada nó da árvore de decisão pode ser podado. A poda consiste em remover a subárvore de dado nó, transformá-lo em folha e dar a ele a classificação mais comum dos exemplos nele contidos. Para a base GO *Church* o tamanho da árvore (altura) foi igual a 850, enquanto para a base GO *Cellcycle*, o tamanho da árvore foi igual a 1311. A Tabela 4 mostra a relação da média dos resultados obtidos entre a técnica de classificação J48 e as bases de dados.

Tabela 4 – Média dos resultados obtidos pelo J48

Base de Dados	*TP Rate*	*FP Rate*	Precisão	*Recall*	*F-Measure*	*ROC Area*
Cellcycle	0.452	0.000	0.249	0.452	0.307	0.999
Church	0.293	0.005	0.149	0.293	0.185	0.983

Fonte: Autoria própria

4.1.3 Resultados Obtidos pelo Método de Redes Neurais Artificiais - MLP

O método MLP já é configurado no Weka como um classificador que usa *Backpropagation* para classificar as instâncias e validação limiar (*Threshold)* que é utilizada para determinar um valor θ, tal que θ é quem define e envia para fora do neurônio o valor do estímulo a ser passado adiante, para os próximos neurônios da rede. A Tabela 5 mostra a relação da média dos resultados obtidos entre a técnica de classificação MLP e as bases de dados.

Tabela 5 – Média dos resultados obtidos pelo MLP

Base de Dados	*TP Rate*	*FP Rate*	Precisão	*Recall*	*F-Measure*	*ROC Area*
Cellcycle	0.036	0.004	0.029	0.036	0.024	0.885
Church	0.03	0.003	0.03	0.03	0.018	0.937

Fonte: Autoria própria

4.2 ANÁLISE COMPARATIVA DOS MÉTODOS DE CLASSIFICAÇÃO

A análise comparativa dos métodos de classificação levará em conta as taxas percentuais de instâncias classificadas correta e incorretamente. Os resultados obtidos serão: a porcentagem correta para cada método, o resultado estatístico do método que obteve maior e menor taxa de acerto de predição de classes e a taxa da margem de predição.

A classificação das bases GO trata-se de um problema real do campo da biologia molecular e celular, onde o conjunto de proteínas está organizado hierarquicamente. A transformação da base hierárquica para não-hierárquica pelo método *Top-Down* fez com que cada instância possuísse uma classe de 4125 caracteres. Isso tornou os classificadores menos eficientes quanto à taxa de acerto na predição de classes, especialmente o MLP que possui um número de iterações relativamente maior que os outros algoritmos por utilizar regra de correção de erro por retropropagação.

4.2.1 Taxas de Acerto e Erro dos Métodos de Classificação

A Tabela 6, 7 e 8 mostram os valores percentuais das taxas de acerto e erro na predição das classes das bases de dados GO *Cellcycle* e *Church* pelos classificadores KNN, J48 e MLP, respectivamente.

Tabela 6 – Taxas de acerto e erro na predição pelo método KNN

Base de Dados	Instâncias classificadas corretamente	Instâncias classificadas incorretamente
Cellcycle	17.0462 %	82.9538 %
Church	31.2231 %	68.7769 %

Fonte: Autoria própria

Tabela 7 – Taxas de acerto e erro na predição pelo método J48

Base de Dados	Instâncias classificadas corretamente	Instâncias classificadas incorretamente
Cellcycle	45.2308 %	54.7692 %
Church	29.2563 %	70.7437 %

Fonte: Autoria própria

Tabela 8 – Taxas de acerto e erro na predição pelo método MLP

Base de Dados	Instâncias classificadas corretamente	Instâncias classificadas incorretamente
Cellcycle	3.6308 %	96.3692 %
Church	2.9502 %	97.0498 %

Fonte: Autoria própria

4.2.2 Margem de Predição dos Métodos de Classificação

As predições dos métodos de classificação serão ilustradas através de um gráfico chamado curva de margem (*margin curve*). Esse gráfico ilustra a margem de previsão, que é definida como a diferença entre a probabilidade da classe predita real e a maior probabilidade predita para as outras classes, descrito pela Equação 5.

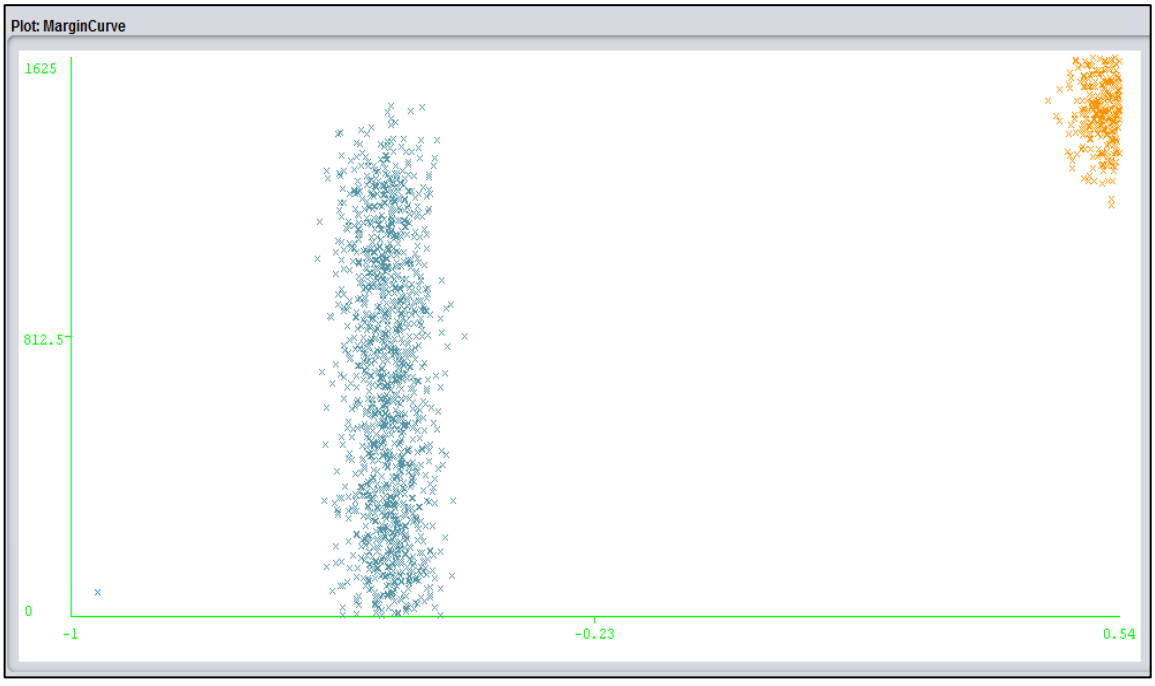
$$margin, M(X,Y) = P(Y_\theta = Y) - \max P(Y_\theta = Z) \quad (5)$$

Onde Y_θ é a classe predita da instância X de acordo com um conjunto de dados construído a partir da fase de teste. Quanto maior a margem, maior a probabilidade de o classificador predizer a classe de um exemplo X corretamente. O gráfico da curva de margem ajuda a avaliar o desempenho do classificador, o qual é medido probabilisticamente em termos de sua margem.

As Figuras 18 (a), 18 (b), 19 (a), 19 (b), 20 (a) e 20 (b) ilustram as curvas de margem dos algoritmos KNN, J48 e MLP sobre as bases de dados GO *Cellcycle* e *Church*. O eixo *X* representa o número da margem que é plotado com o número de casos no eixo *Y*. Nas Figuras a seguir, o conjunto de instâncias plotados entre -1 e 0.5 (azul) indica a margem de casos classificados errados, e o conjunto entre 0.5 e 1 (laranja) indica a margem dos casos corretamente classificados.

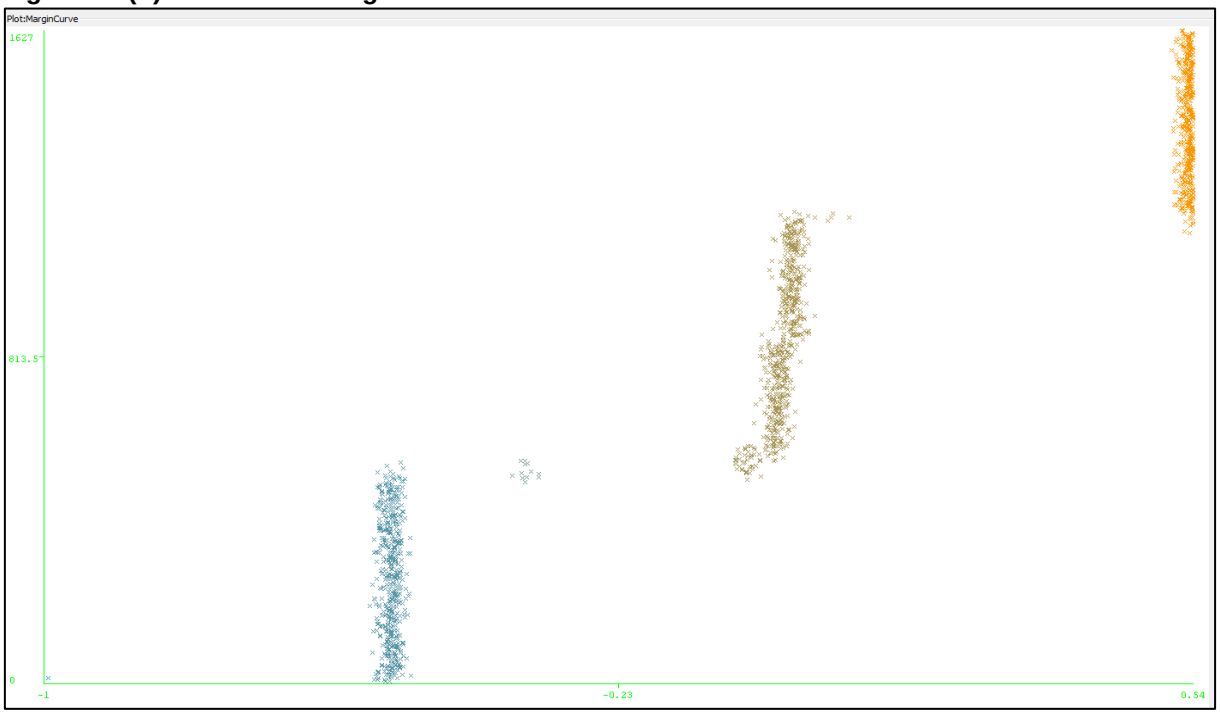
Nas Figuras 18 (a) e 18 (b) são ilustradas as curvas de margem do algoritmo KNN. Como o KNN é um algoritmo de aprendizagem baseado em exemplos, sua curva de margem possui limites bem definidos, ou seja, há pouca variação entre os pontos que são classificados correta e incorretamente.

Figura 18 (a) – Curva de margem do método KNN sobre a base *Cellcycle*



Fonte: Autoria própria

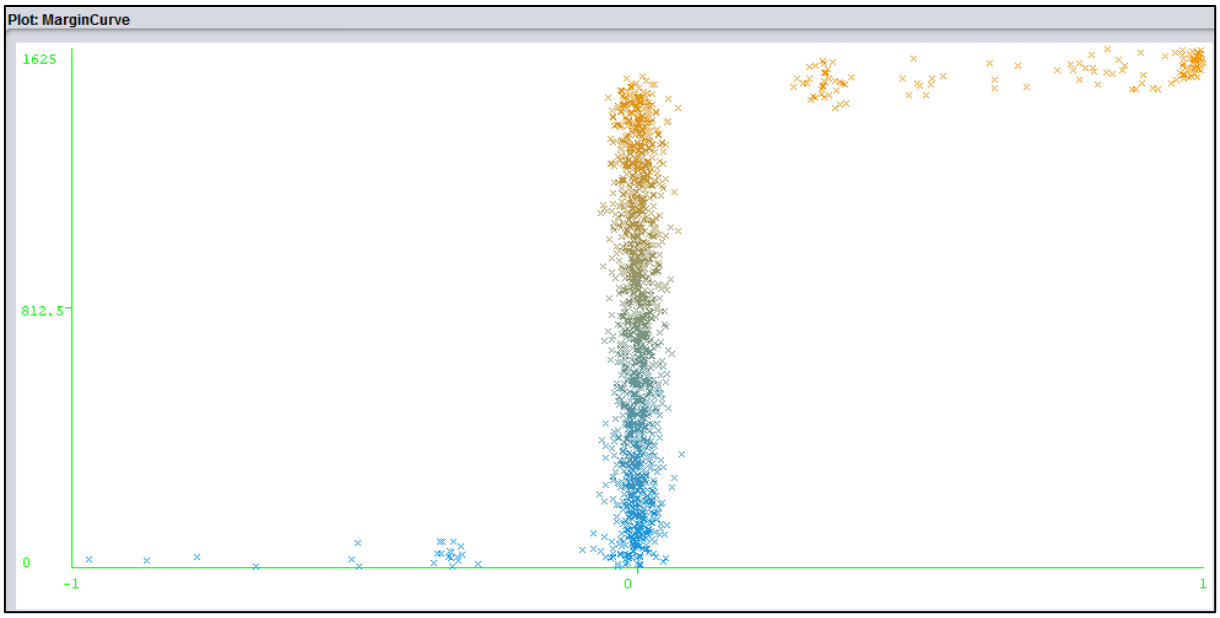
Figura 18 (b) – Curva de margem do método KNN sobre a base *Church*



Fonte: Autoria própria

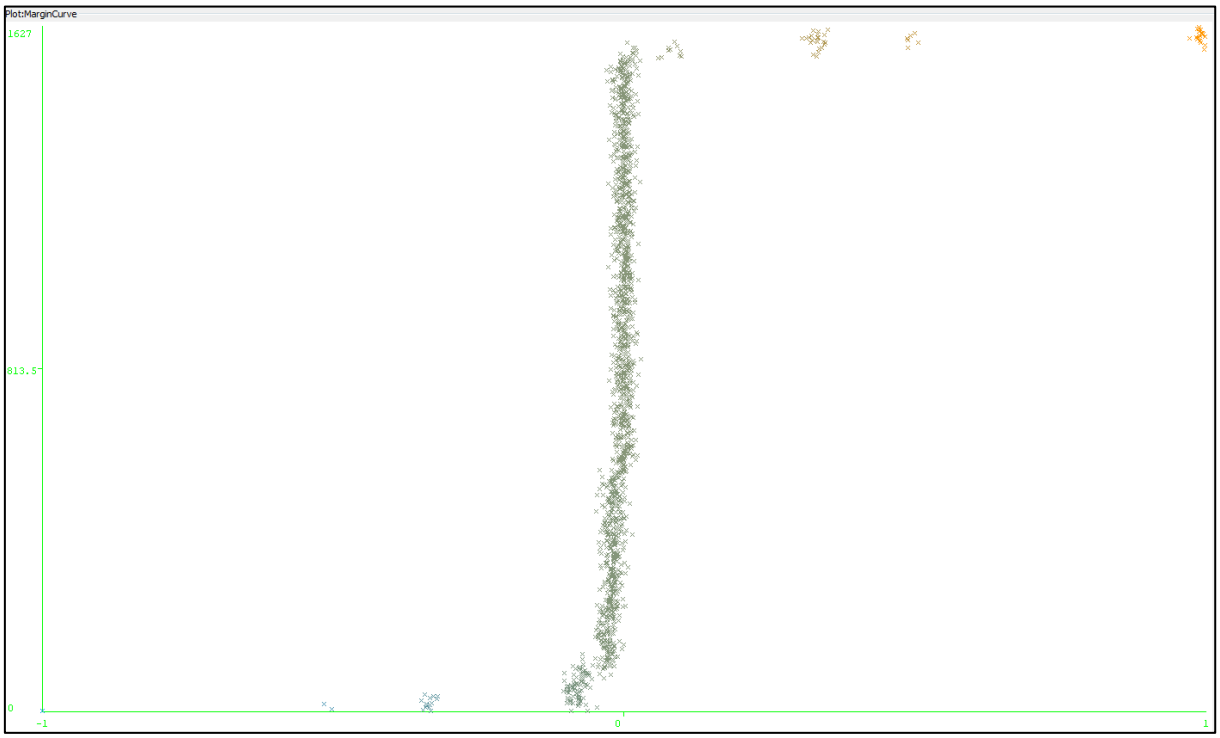
Nas Figuras 19 (a) e 19 (b) são ilustradas as curvas de margem do algoritmo J48. Nota-se que os pontos que representam as instâncias plotadas variam mais sobre o eixo da margem conforme se aumenta o número de instâncias de entrada para teste.

Figura 19 (a) – Curva de margem do método J48 sobre a base *Cellcycle*



Fonte: Autoria própria

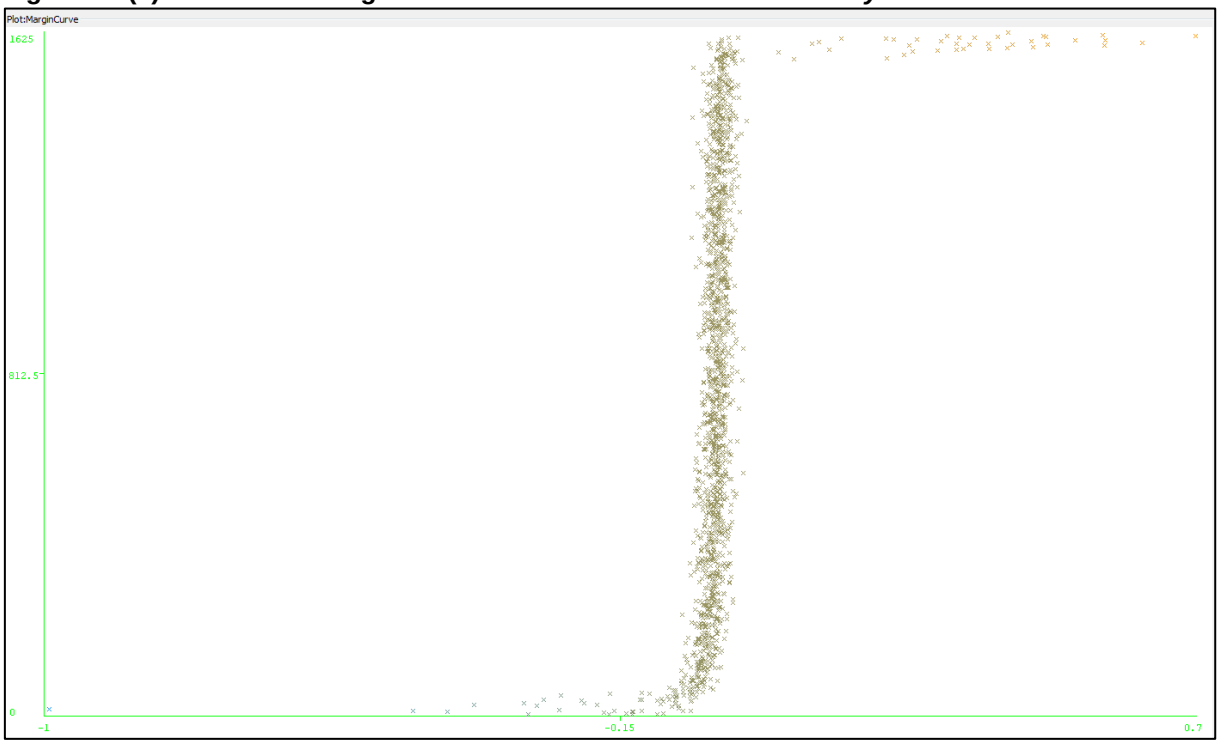
Figura 19 (b) – Curva de margem do método J48 sobre a base *Church*



Fonte: Autoria própria

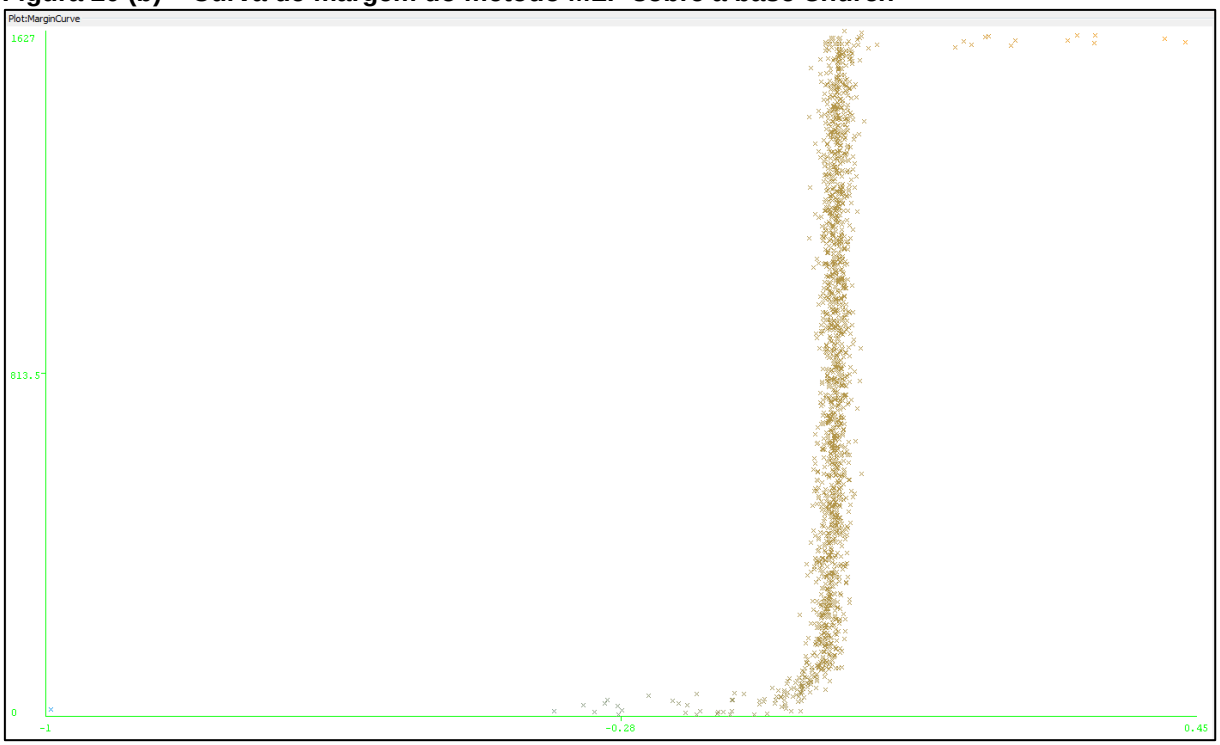
Nas Figuras 20 (a) e 20 (b) são ilustradas as curvas de margem do algoritmo MLP. Este é o método que apresenta a maior taxa de erro de predição de classes, portanto a curva de margem possui um padrão no qual a maior concentração de instâncias está plotadas entre -1 e 0.5 e com pouca variação.

Figura 20 (a) – Curva de margem do método MLP sobre a base *Cellcycle*



Fonte: Autoria própria

Figura 20 (b) – Curva de margem do método MLP sobre a base *Church*



Fonte: Autoria própria

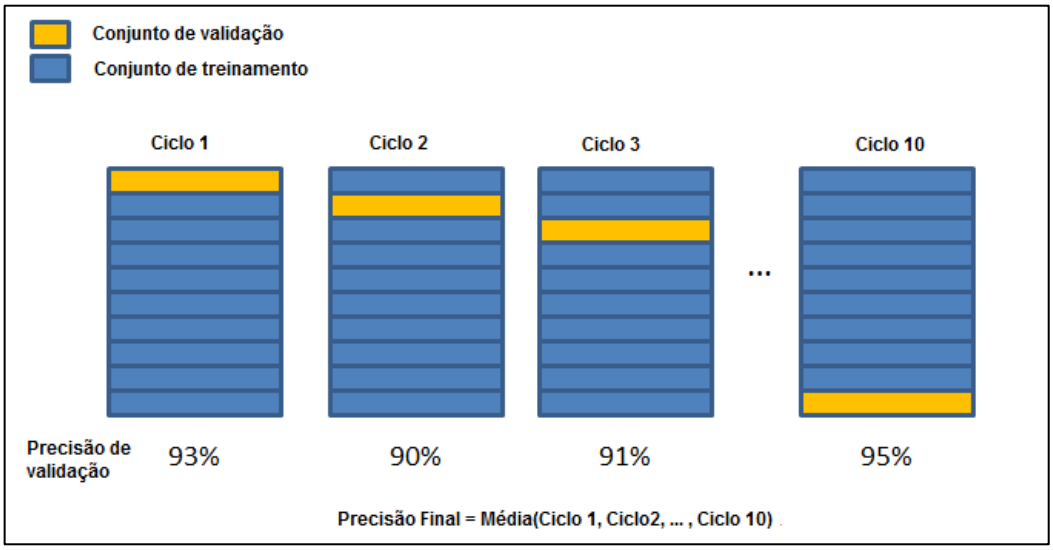
4.2.3 Validação dos Classificadores KNN e J48

A validação dos classificadores deste trabalho será realizada utilizando o método de *k-fold cross-validation* (validação cruzada com *k* dobras), que é um método estatístico de avaliar e comparar algoritmos de aprendizagem.

O método *k-fold* consiste em dividir o conjunto total de dados em *k* subconjuntos mutuamente exclusivos do mesmo tamanho e, a partir disto, um subconjunto é utilizado para teste e os *k-1* restantes são utilizados para estimação dos parâmetros para cálculo da precisão do modelo.

Este processo é realizado *k* vezes alternando de forma circular (ciclos) o subconjunto de teste (KOHAVI et al, 1995). A Figura 21 mostra um exemplo com *k* = 10 iterações. Em cada ciclo é obtido uma porcentagem de precisão da validação e ao final é calculada a média para obter a precisão final.

Figura 21 – Exemplo de esquema de particionamento e execução do método *k-fold* com *k* = 10



Fonte: AKYILDIZ et al (2014)

Ao final das *k* iterações calcula-se a precisão sobre os erros encontrados obtidos por:

$$Ac_f = \frac{1}{v}\sum_{i=1}^{v} \epsilon_{y_i,Y_i} = \frac{1}{v}\sum_{i=1}^{v}(y_i - Y_i) \qquad (6)$$

onde: v é o número de dados de validação e $\in_{y_i,Y_i}$ é a diferença entre o valor real da saída i e o valor da classe predita. Com isso, obtêm-se uma medida mais confiável sobre a capacidade do modelo em representar o processo gerador dos dados (KOHAVI *et al*, 1995).

Nesse experimento, os classificadores KNN e J48 são executados dez vezes com 10-*fold cross-validation*. O experimento será executado sobre as bases de dados GO *Cellcycle* e *Church* com 66% das instâncias utilizadas para treinamento e 34% para teste.

Para o classificador MLP não foi possível executar a validação cruzada em virtude de sua complexidade em tempo de execução ser inestimada, não tendo previsão para o seu término e apresentação dos resultados.

As porcentagens de precisão corretas para cada um dos dois classificadores sobre as duas bases de dados são mostradas nas Tabelas 9 e 10: 0.89% para o KNN (*lazy.IB*), com um desvio padrão de 0.31, e 0.50% para o J48 (*tree*), com um desvio padrão de 0.09, sobre a base de dados *Cellcycle*; 0.71% para o KNN, com um desvio padrão de 0.25, e 1.14% para o J48, com um desvio padrão de 0.31, sobre a base de dados *Church*.

Tabela 9 – Resultados sobre a base de dados *Cellcycle*

Base de Dados	(1) *lazy.IB*	(2) *tree*
cellcycle0.names	(100) 0.89 (0.31)	0.50 (0.09)
	(v/ / *)	(0/ 1/ 0)

Fonte: Autoria própria

Tabela 10 – Resultados sobre a base de dados *Church*

Base de Dados	(1) *lazy.IB*	(2) *tree*
church0.names	(100) 0.71 (0.25)	1.14 (0.31) v
	(v/ / *)	(1/ 0/ 0)

Fonte: Autoria própria

As anotações v ou * indicam que um resultado específico é estatisticamente melhor (v) ou pior (*) (nesse caso, KNN) a um nível de significância especificado (0.05). Os resultados do J48 são estatisticamente melhores que os resultados obtidos por KNN. Na parte inferior de cada coluna após a primeira existe uma

contagem (xx/ yy/ zz) do número de vezes em que um classificador foi melhor (xx), igual (yy), ou pior (zz) que os outros classificadores do experimento.

No experimento com a base de dados Cellcycle, houve uma vez em que J48 teve desempenho igual que o KNN e nunca melhor ou pior (0/ 1/ 0); no experimento com a base de dados Church, houve uma vez em que o classificador J48 foi melhor que o KNN e nunca equivalente ou pior.

O valor (100) no início da linha onde estão descritas as bases, representa o número de estimativas que são usadas para calcular o desvio padrão (número de execuções nesse caso).

4.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentados os resultados dos experimentos dos métodos de classificação aplicados sobre as bases de dados GO estudadas. Além dos resultados, foi realizada uma análise comparativa entre os métodos, listando suas taxas de acerto na predição através da quantidade de instâncias classificadas corretamente e pela análise das curvas de margem ilustradas na seção 4.2.2.

Por fim, foi feita a validação dos métodos utilizados a fim de se obter um índice de confiança dos classificadores e comparar seus desempenhos sobre cada execução. Esta validação foi realizada pelo método de validação cruzada que avalia e compara estatisticamente diferentes algoritmos de aprendizagem.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo serão apresentadas as considerações finais sobre o estudo, comparando e avaliando os objetivos gerais e específicos propostos no trabalho com os resultados obtidos após a realização dos experimentos. Também serão mostradas as limitações ocorridas durante o desenvolvimento, bem como as contribuições e os possíveis trabalhos futuros.

5.1 CONCLUSÕES

O objetivo geral de investigar técnicas de classificação hierárquica multirrótulo que pudessem ser capazes de prever a quantidade de classes de uma nova instância de entrada foi alcançado, baseando-se na metodologia descrita no capítulo 3.

Os objetivos específicos de estudar os principais conceitos de classificação hierárquica multirrótulo, aplicar as técnicas de classificação estudadas a uma base de dados hierárquica e multirrótulo, analisar os resultados através de avaliação de classificadores e aplicar essas técnicas avaliadas para previsão da quantidade de classes de novas instâncias, foram alcançados como se pode analisar pelo referencial teórico e pelos resultados dos experimentos deste trabalho.

Apesar de o objetivo de aplicar as técnicas avaliadas para previsão da quantidade de classes de novas instâncias ter sido alcançado, uma das técnicas se mostrou pouco eficiente quanto aos resultados obtidos. Esta técnica foi a MLP, que é uma técnica de RNA baseada em função. A hipótese levantada para o seu baixo desempenho foi a de que para uma base de dados hierárquica multirrótulo pré-processada para uma base plana e simples-rótulo, torna a função de ativação pouco precisa para a predição da classe correta de uma instância.

Além de seu baixo desempenho na predição, outro problema identificado quanto à técnica MLP foi o tempo de execução. Enquanto as outras técnicas, KNN e J48, demoraram em torno de 5 minutos para apresentarem os resultados, o MLP demorou aproximadamente 22 horas. O tempo de execução do MLP está

relacionado à retropropagação, que ajusta os pesos da função de ativação de acordo com uma regra de correção de erro na predição de classes.

O baixo desempenho e o problema no tempo de execução do MLP impossibilitou a comparação entre os três métodos estudados, sendo apresentados apenas os resultados da validação entre os métodos KNN e J48, que também se mostraram mais eficientes.

A partir dos resultados obtidos e das comparações entre os métodos estudados, conclui-se que o J48, método de classificação baseado em árvore de decisão, é o que possui melhor desempenho para previsão da quantidade de classes de novas instâncias.

Por possuir uma maior taxa de acerto na classificação correta de novos exemplos, o J48 é também o que possui maior precisão no valor de confiança sobre as classes. Com base no modelo obtido e em suas características, as novas instâncias possuirão as classes que tiverem o maior valor de confiança, podendo assim se obter a quantidade de classes que será determinada para novas instâncias apresentadas ao modelo de classificação.

5.2 TRABALHOS FUTUROS

A partir da pesquisa e dos resultados obtidos, algumas sugestões de trabalhos futuros podem ser listadas:

- Utilização das mesmas técnicas estudadas, mas em bases de dados de diferentes domínios com diferentes padrões;
- Aplicação do classificador ZeroR que prediz os rótulos das instâncias pela categoria majoritária e desempenha uma linha de base como referência para outros métodos de classificação;
- Utilização de outras ferramentas e *frameworks* de AM, como o Mulan (TSOUMAKAS et al, 2011) e o Meka (READ et al, 2016), que é uma extensão do Weka para bases multirrótulo, para realização dos mesmos experimentos;
- Aplicação de diferentes técnicas de classificação, mas mantendo o padrão hierárquico multirrótulo das bases de dados.

REFERÊNCIAS

ACHARYA, U. R.; BHAT, P. S.; IYENGAR, S. S.; RAO, A.; DUA, S. *Classification of heart rate data using artificial neural network and fuzzy equivalence relation.* ***Pattern Recognition,*** v. 1, n. 36, p. 61-68, jan. 2003.

AHA, D. W.; KIBLER, D.; ALBERT, M. K. *Instance-based learning algorithms.* ***Machine learning***, v. 6, n. 1, p. 37-66, 1991.

AKYILDIZ, B. *et al*. *An Introduction to Supervised Learning via Scikit Learn*. ***Machine Learning Newsletter***, New York, 2014. Disponível em: < http://bugra.github.io/work/notes/2014-11-22/an-introduction-to-supervised-learning-scikit-learn/>. Acesso em: 28 set. 2016.

ASHBURNER, M. et al. *Gene Ontology: tool for the unification of biology*. ***Nature genetics***, v. 25, n. 1, p. 25-29, 2000.

BARBARA, C. S.; THISSIANY, B. A. **Previsão da Quantidade de Classes em Classificação Hierárquica Multirrótulo**. 2014. 75f. Trabalho de Conclusão de Curso, Universidade Tecnológica Federal do Paraná, UTFPR. Ponta Grossa, 2014.

BORGES, H. B. **Classificador Hierárquico Multirrótulo Usando uma Rede Neural Competitiva**. 2012. 188f. Tese (Doutorado), Pontifícia Universidade Católica do Paraná, PUC-PR. Curitiba, 2012.

BOUCKAERT, R. R. *et al*. ***WEKA Manual for Version 3-8-0***. Hamilton, New Zealand, 2016.

BOUTELL, M. R.; LUO, J.; SHEN, X.; BROWN, C. M. *Learning Multi-label Scene Classification.* ***Pattern Recognition,*** v. 9, n. 37, p. 1757-1771, 2004.

CARDON, A.; MÜLLER, D. N.; **Introdução às Redes Neurais Artificiais**. 1994. 32f. Curso de Pós-Graduação em Ciência da Computação, Instituto de Informática, UFRGS. Porto Alegre, 1994.

CARVALHO, A. C.; FREITAS, A. A. *A Tutorial on Hierarchical Classification with Applications in Bioinformatics.* ***Idea Group. Research and Trends in Data Mining Technologies and Applications,*** v. 1, n. 7, p. 175-208, 2007.

CARVALHO, A. C.; FREITAS, A. A. *A Tutorial on Multi-label Classification Techniques.* ***Foundations of Computational Intelligence****,* v. 5, n. 205, p. 177-195, 2009.

CERRI, R. **Técnicas de Classificação Hierárquica Multirrótulo**. 2010. 241f. Dissertação (Mestrado), Instituto de Ciências Matemáticas e de Computação, USP. São Carlos, 2010.

CLARE, A.; KING, R. D. *Knowledge Discovery in Multi-label Phenotype Data.* ***5th European Conference on Principles of Data Mining and Knowledge Discovery****,* v. 2168, p. 42-53, 2001.

COELHO, T. A. **Uma Estratégia Híbrida para o Problema de Classificação Multirrótulo**. 2011. 75f. Dissertação (Mestrado), Universidade Federal Minas Gerais, Niterói, UFMG. Belo Horizonte, 2011.

COMITÉ, F.; GILLERON, R.; TOMMASI, M. *Learning Multi-label Alternating Decision Trees from Texts and Data.* ***Proceedings of the MLDM 2003****,* n. 2734, p. 251-274, 2003.

COSTA, E. P. **Investigação de Técnicas de Classificação Hierárquica para Problemas de Bioinformática**. 2008. 184f. Dissertação (Mestrado), Instituto de Ciências Matemáticas e de Computação, ICMC-USP. São Carlos, 2008.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. *From Data Mining to Knowledge Discovery in Databases*. ***AI magazine***, v. 17, n. 3, p. 37, 1996.

FISHER, R. A. *The use of multiple measurements in taxonomic problems*. ***Annals of eugenics***, v. 7, n. 2, p. 179-188, 1936.

FREUND, Y.; SCHAPIRE, R. E. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* ***Journal of Computer and System Sciences****,* n. 971504, p. 23-37, 1995.

GOLDBERG, A. B. *et al*. *Toward text-to-picture synthesis*. In: ***NIPS 2009 mini-Symposia on Assistive Machine Learning for People with Disabilities***. 2009.

GOLDSCHMIDT, R.; PASSOS, E. ***Data Mining*****:** Um Guia Prático-Conceitos, Técnicas, Ferramentas, Orientações e Aplicações. Rio de Janeiro: Campus, 2005.

HALL, M.; FRANK, E.; GEOFFREY, H.; BERNHARD, P.; REUTEMANN, P.; WITTEN, I. H. *The WEKA Data Mining Software: An Update*. ***SIGKDD Exploration***, v. 11, 2009.

HAN, J.; MICHELINE, K.; PEI, J. ***Data Mining:*** *Concepts and Techniques*. 3. ed. Waikato: Elsevier, 2012.

HAND, D. J. *Measuring classifier performance: a coherent alternative to the area under the ROC curve*. ***Machine learning***, v. 77, n. 1, p. 103-123, 2009.

HAYKIN, S. **Redes Neurais:** Princípios e Prática. 2. ed. Porto Alegre - RS, 2000.

HUAN, S.; GUOZHENG, L.; GUOPING, L.; YIQIN, W. *Symptom Selection for Multi-label Data of Inquiry Diagnoses in Traditional Chinese Medicine.* ***Science China Information Sciences,*** v. 5, n. 56, p. 1-5, 2013.

JAIN, L. C.; GHOST, A. ***Evolutionary Computation in Data Mining*** *(Studies in Fuzziness and Soft Computing).* 4. ed. Springer-Verlag New York, Inc., NJ, USA, 2005.

KOHAVI, R. *et al*. *A study of cross-validation and bootstrap for accuracy estimation and model selection*. **In: Ijcai,** p. 1137-1145, 1995.

McCULLOCH, W. S.; PITTIS, W. *A Logical Calculus of the Ideas Immanent in Nervous Activity.* ***Bulletin of Mathematical Biophysics,*** v. 5, p. 115-133, 1943.

MITCHELL, T. M. ***Machine Learning***. Redmond: McGraw-Hill, 1997.

MONARD, M. C.; BARANAUSKAS, J. A. **Conceitos sobre Aprendizado de Máquina**. (Ed.) Sistemas Inteligentes: Fundamentos e Aplicações. São Carlos: Manole, 2003.

PARKER, D. B. *Optimal algorithms for adaptive networks: second order back propagation, second order direct propagation, and second order.* ***Proceedings of the IEEE International Conference on Neural Network****,* p. 593-600, 1987.

PRATI, R. C.; BATISTA, G.; MONARD, M. C. Curvas ROC para avaliação de classificadores. **Revista IEEE América Latina**, v. 6, n. 2, p. 215-222, 2008.

QUINLAN, J. R. ***C4.5:*** *programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

READ, J. et al. *Meka: a multi-label/multi-target extension to weka.* ***Journal of Machine Learning Research***, v. 17, n. 21, p. 1-5, 2016.

REZENDE, S. O. **Sistemas inteligentes:** Fundamentos e Aplicações. Editora Manole Ltda, 2003.

ROTH, F. P.; HUGHES, J. D.; ESTEP, P. W.; CHURCH, G. M. *Finding DNA Regulatory Motifs within Unaligned Noncoding Sequences Clustered by Whole-genome mRNA Quantitation*. ***Nature Biotechnology***. V. 16, p. 939-949, 1998.

SARIDIS, G. *Parameter Estimation: Principles and Problems.* ***IEEE Transactions on****,* n. 28, p. 634-635, 1983.

SCHAPIRE, R. E.; SINGER, Y. *A Boosting-based System for Text Categorization.* ***Machine Learning,*** v. 2-3, n. 39, p. 135-168, 2000.

SILVA, P. N. **Classificação Multirrótulo em Cadeia:** Novas Abordagens. 2014. 81f. Dissertação (Mestrado), Universidade Federal Fluminense, Niterói, 2014.

SPELLMAN, P. *et al*. *Comprehensive Identification of Cell Cycle-regulated genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization*. ***Molecular Biology of the Cell***. V.9, p. 3273-3297, 1998.

TAN, Pang-N.; STEINBACH, M.; KUMAR, V. ***Introduction to Data Mining***. *Pearson Education* India, 2006.

TROHIDIS, K.; TSOUMAKAS, G.; KALLIRIS, G.; VLAHAVAS, I. *Multilabel Classification of Music into Emotions.* ***9th International Conference on Music Information Retrieval (ISMIR 2008),*** v. 3, 2008.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. *Mining Multi-label Data.* ***Data Mining and Knowledge Discovery Handbook,*** v. 2, n. 34, p. 667-685, 2010.

TSOUMAKAS, G.; XIOUFIS, E. S.; VILCEK, J.; VLAHAVAS, I. P. *Mulan: A Java Library for Multi-label Learning*. ***Journal of Machine Learning Research***, 7 jul. 2011.

VENS, C. *et al*. *Decision trees for hierarchical multi-label classification*. ***Machine Learning***, v. 73, n. 2, p. 185-214, 2008.

WITTEN, I. H.; FRANK, E.; HALL, M. A. ***Data Mining:*** *Practical Machine Learning Tools and Techniques*. Waikato: Elsevier, 2011.

ZHANG, Min-L.; ZHOU, Zhi-U. *A k-nearst Neighbor Based Algorithm for Multi-label Classification.* ***IEEE Computational Intelligence Society****,* v. 2, n. 2, p. 718-721, 2005.