LEONARDO HENRIQUE PEREIRA

ANÁLISE DE TÉCNICAS DE CLASSIFICAÇÃO LOCAL HIERÁRQUICA USANDO SELEÇÃO DE ATRIBUTOS PARA PREVISÃO DA FUNÇÃO DE PROTEÍNAS

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

CURITIBA

2019

LEONARDO HENRIQUE PEREIRA

ANÁLISE DE TÉCNICAS DE CLASSIFICAÇÃO LOCAL HIERÁRQUICA USANDO SELEÇÃO DE ATRIBUTOS PARA PREVISÃO DA FUNÇÃO DE PROTEÍNAS

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Descoberta do Conhecimento e Aprendizagem de Máquina

Orientador: Prof. Dr. Júlio Cesar Nievola

CURITIBA

2019

Pereira, Leonardo Henrique
Análise de Técnicas de Classificação Hierárquica Usando Seleção de Atributos para a Previsão da Função de Proteínas. Curitiba, 2019. 70p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Bioinformática 2. Classificação Hierárquica de Proteínas 3. Predição de Funções Protéicas 4. Seleção de Atributos. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática II-t

Esta página deve ser reservada à ata de defesa e termo de aprovação que serão fornecidos pela secretaria após a defesa da dissertação e efetuadas as correções solicitadas.

Dedicatória

Aos meus pais por sempre me guiarem e
manterem a confiança em mim,
durante toda minha vida.

Agradecimentos

Aos meus pais por me darem o incentivo e a confiança de continuar em meus estudos, fornecendo amparo e se sacrificando para o meu crescimento.

Ao meu orientador, Dr. Júlio Cesar Nievola, pelo apoio e suporte para eu continuar o desenvolvimento desta dissertação.

Aos meus colegas de mestrado e todos que passaram pela minha vida e me ajudaram no engrandecimento deste trabalho.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Equações

Lista de Abreviaturas e Siglas

Acc	*Acurácia*
AM	*Aprendizado de Máquina*
CFS	*Correlation-based Feature Selection*
DAG	*Directed Acyclic Graph*
DNA	*Deoxyribonucleic Acid*
FN	*Falso Negativo*
FP	*Falso Positivo*
GPCR	*G-protein-coupled receptors*
IA	*Inteligência Artificial*
KDD	*Knowledge-discovery in Databases*
MLNP	*Mandatory leaf-node prediction*
mRNA	*RNA Mensageiro*
Pre	*Precision*
HP	*Hierarchical precision*
Rep	*Recall*
HR	*Hierarchical Recall*
RNA	*Ribonucleic acid*
rRNA	*RNA Ribossômico*
SVM	*Support Vector Machine*
Tach	*Taxa de Acerto Hierárquico*
Terr	*Taxa de Erro*
Terrh	*Taxa de Erro Hierárquico*
tRNA	*RNA Transportador*
VN	*Verdadeiro Negativo*
VP	*Verdadeiro Positivo*
WEKA	*Waikato Environment for Knowledge Analysis*

Resumo

Com o rápido avanço de pesquisas nas áreas genômica e proteômica, o crescimento de bases com dados biológicos foi inevitável, tornando a análise destes dados uma tarefa hercúlea para o ser humano. Assim, foi indispensável a intervenção da informática para suprir essa necessidade.

A Bioinformática é utilizada para fazer a análise da informação no campo da biologia utilizando técnicas da informática. Um dos problemas desta área é a previsão da função de proteínas, que não é tão comum pelo fato da análise ser muito trabalhosa e complexa de se tratar, principalmente quando existe classes com hierarquia, ou seja, suas classes organizadas em superclasses que herdam funções protéicas de subclasses, formando estruturas de arvores ou grafos acíclicos direcionados. A proposta aqui apresentada é a classificação hierárquica da função de proteínas utilizando algoritmos de aprendizagem de máquina para a classificação de classificadores, realizando assim a predição das funções protéicas. Este trabalho explora a aplicação das abordagens de classificação local hierárquica por nó, por nó pai e por camada em conjunto à seleção de atributos. Os resultados foram obtidos através da realização dos experimentos usando a média hierárquica e o desvio padrão, calculados através das taxas de acerto que os algoritmos de classificação hierárquica obtiveram. A partir dos resultados encontrados, foram feitas comparações entre os métodos de classificação hierárquicos com e sem a seleção de atributos, onde no cenário de predição da função de proteína aqui proposto, se tornam muito mais favoráveis com a abordagem de classificação hierárquica local por camada e não utilizando a seleção de atributos.

Palavras-Chave: Bioinformática, Classificação Hierárquica de Proteínas, Previsão da Função de Proteínas, Seleção de Atributos.

Abstract

With the rapid advance of research in the genomic and proteomic areas, growth of bases with biological data was inevitable, making the analysis of these data a herculean task for the human. Thus, it was indispensable the intervention of informatics to meet this need.

Bioinformatics is used to make the analysis of information in the field of biology using computer techniques. One of the problems of this area is the protein function prediction, which is not so common because the analysis is very laborious and complex to deal with, especially when there are classes with hierarchy, that is, their classes organized into superclasses that inherit protein functions of subclasses, forming tree structures or directed acyclic graphs. The proposal presented here is the hierarchical classification of the protein function using machine learning algorithms for the induction of classifiers, performing the protein functions prediction. This work explores the application of hierarchical local classification approaches by node, by parent node and by layer in conjunction with selection of attributes. The results were obtained by performing the experiments using the hierarchical average and the standard deviation, calculated by the hit rates that the hierarchical classification algorithms obtained. From the results found, comparisons were made between the hierarchical classification methods with and without attribute selection, where in the prediction of the protein function scenario proposed here, became much more favorable with the local hierarchical classification approach by layer and not using attribute selection.

Keywords: Bioinformatics, Hierarchical Protein Classification, Prediction of Protein Function, Attribute Selection.

Capítulo 1

Introdução

Com a grande quantidade de dados biológicos gerados e providos, há um aumento exponencial das bases de dados atualmente existentes [BAL01]. Isso se dá pela quantidade de exemplos e técnicas novas, que visam melhorar a eficiência na manipulação de sequência de genomas e proteoma. Para atender essa situação foi necessária a criação de um novo campo, o qual cobre a análise da informação na área de estudo da biologia, o qual denomina-se Bioinformática.

Em 1953 o pesquisador norte-americano James Watson e o britânico Francis Crick, fizeram a descoberta de uma estrutura molecular chamada ácido desoxirribonucleico, bem conhecida como DNA, valendo-lhes um prêmio Nobel. Desta forma surgiu uma enorme quantidade de sequencias que serão armazenadas, exigindo cada vez mais da eficiência de recursos computacionais [SOU03].

No final dos anos 80 em diante, o termo Bioinformática tem sido usado para referenciar métodos computacionais para a análise comparativa de dados do genoma, se difundindo com a grande quantidade de dados e estudos na área da biologia [HOG11]. Neste período, visto que era uma atividade hercúlea executar a manipulação destas informações de forma manual, ferramentas computacionais foram desenvolvidas para a praticidade e facilidade da analise destes dados [SOU03].

Com relações interdisciplinares, pois combina conhecimentos das áreas de química, física, biologia, ciência da computação, informática, matemática e estatística, a Bioinformática está ligada à utilização de diversas técnicas e ferramentas do mundo computacional que visam resolver os problemas da Biologia [BAL01]. Em suma, esse campo coleta e processa os dados de genomas para então manipular e estudar as funções protéicas. Nesse contexto, pode ser

utilizado como exemplo da utilização destas manipulações nas companhias farmacêuticas, ajudando-as a realizarem estudos mais profundos em estruturas de proteínas para desenvolverem novas drogas [COH04]. Também pode-se ter como exemplo a análise de expressão dos genes, pois no campo das Ciências Biológicas é extremamente importante acompanhar possíveis mudanças nos padrões dessas análises [ALB97].

Algumas das tarefas da Bioinformática incluem: classificar e deduzir as funções de uma proteína, buscar genes em um genoma determinado, encontrar possíveis áreas na estrutura da proteína, onde uma molécula pode ser anexada [COH04].

Sempre há requisições por análise de dados de uma forma eficiente mantendo um alto grau de confiabilidade. Assim, a técnica de microarranjo de DNA é muito bem vista nesse contexto, pois possibilita o estudo de expressões genicas, com um custo de tempo cada vez menor. Com a constante aceitação e referências nas pesquisas dessa área, esta se tornou padrão na maioria dos locais que pesquisam o genoma.

Com a frequente utilização de microarranjos de DNA em experimentos, gerando assim uma grande quantidade de dados, foi necessária a interação da área da inteligência artificial, para que posam ser explorados ao máximo o que essa tecnologia tem a oferecer. O ramo de IA está se envolvendo pela sua capacidade de aprendizagem automática desses dados gerados por microarranjos, assim é possível aplicar algoritmos de aprendizagem de máquina a fim de produzir hipóteses uteis a partir do resultado dos experimentos [BAL01].

Neste trabalho foi explorado algumas técnicas de Aprendizagem de máquina, envolvendo mineração de dados e classificação hierárquica, devido a presença da relação de herança em suas classes.

As bases de microarranjos, aqui trabalhadas, contêm um grande número de atributos e exemplos, que muitas vezes atrapalham e reduzem o desempenho dos algoritmos de AM, então existe o desafio de utilizar técnicas de redução de dimensionalidade para melhorar a predição dos classificadores.

1.1 Objetivos

Objetivo Geral

Neste trabalho serão analisadas técnicas de classificação local hierárquica de funções protéicas utilizando seleção de atributos. Serão empregadas as seguintes abordagens nas classificações hierárquicas: classificação por nó, por nó pai e por camadas.

Objetivos Específicos

Dentre os objetivos específicos, têm-se:

- Selecionar e comparar algoritmos de classificação local para a o treinamento da previsão da função de proteínas.
- Comparar abordagens hierárquicas que serão utilizadas com o uso da seleção de atributos.
- Implementar as abordagens de classificação local hierárquica local por nó, por nó pai e por camada, bem como um software que auxilie na execução algoritmos que serão comparados e analisados neste trabalho.

1.2 Estrutura do Trabalho

Este trabalho está organizado em 5 capítulos da seguinte maneira: O capítulo 2 apresenta a fundamentação teórica, o qual descreve conceitos relevantes e essenciais para o entendimento deste trabalho. No capítulo 3 é feita a descrição da metodologia com suas etapas, para o desenvolver desta pesquisa. O capítulo 4 expõe os resultados e análises relevantes do trabalho. Finaliza com o capítulo 5, o qual apresenta a conclusão trabalho, contando com as futuras pesquisas.

Capítulo 2

Fundamentação Teórica

Nesta seção será apresentada, de uma forma breve, termos e conceitos importantes, que são essenciais à compreensão das análises e documentações aqui desenvolvidas. Neste sentido, as subsecções seguintes darão noções de entendimento dos respectivos assuntos: na sessão 2.1 é feito um resumo de entendimentos sobre a biologia molecular, dentre eles o DNA, RNA e Proteínas, constando informações sobre microarranjos de proteínas. Na sessão 2.2 é descrito o campo da inteligência artificial: Aprendizagem de máquina, o qual entram os assuntos de descoberta de conhecimento, suas etapas e tarefas, redução de dimensionalidade, classificação hierárquica e medidas de desempenho. Na classificação hierárquica, serão descritas as principais abordagens vistas na literatura, vistas na sessão 2.2.4.1. Ainda em aprendizagem de máquina serão citadas e explicadas técnicas de classificação de dados e exemplos destas, as quais serão trabalhadas neste trabalho. Finaliza na sessão 2.3 com os trabalhos relacionados na área.

2.1 Conceitos da Biologia Molecular

Com a utilização da tecnologia de microarranjos de DNA, em conjunto com algoritmos de classificação da área de aprendizagem de máquina, os quais extraem informações mais relevantes e padrões dessas informações, está sendo possível analisar grandes volumes de dados e identificar proteínas expressas pelo proteoma, auxiliando na predição de suas funções, o que é muito importante para a área da biologia molecular.

A biologia molecular é um ramo da ciência que abrange atividades biológicas a nível molecular, com um foco no estudo da estrutura e funções do material genético e seus produtos de expressão, as proteínas. Em suma, é uma área de estudo ampla, a qual enquadra áreas da

genética, química e bioquímica. A biologia molecular é compreendida como vários sistemas celulares interagindo, incluindo a inter-relação de DNA, RNA e síntese protéica [KEI07].

2.1.1 DNA, RNA e Proteínas

A molécula de DNA carrega a maior parte das instruções genéticas usadas no desenvolvimento, funcionamento e reprodução de todos os organismos vivos.

Para meios de estudos, o DNA é fragmentado e um desses fragmentos pode conter vários genes. Estes genes são uma sequência de nucleotídeos do DNA, os quais podem ser transcritos em uma versão de RNA, esses nucleotídeos entram no processo de expressão genética para a produção de proteínas. O processo é composto por duas fases, que podem ser observadas na Figura 1.

A primeira fase é a transcrição, quando um fragmento de DNA, um gene, tem suas duas cadeias separadas pela polimerase do RNA que se liga a uma região do DNA, denominada proteoma, iniciando assim a síntese de um RNA mensageiro (mRNA). A segunda fase denomina-se tradução, onde é feita a síntese da proteína, ocorre no RNA ribossômico (rRNA), é feito o pareamento do mRNA com o tRNA (RNA transportador). Assim faz com que o aminoácido ligado ao tRNA se desprenda fazendo uma ligação com os outros aminoácidos dos códons próximos ao mRNA, deste modo há a formação da proteína [WEA05].

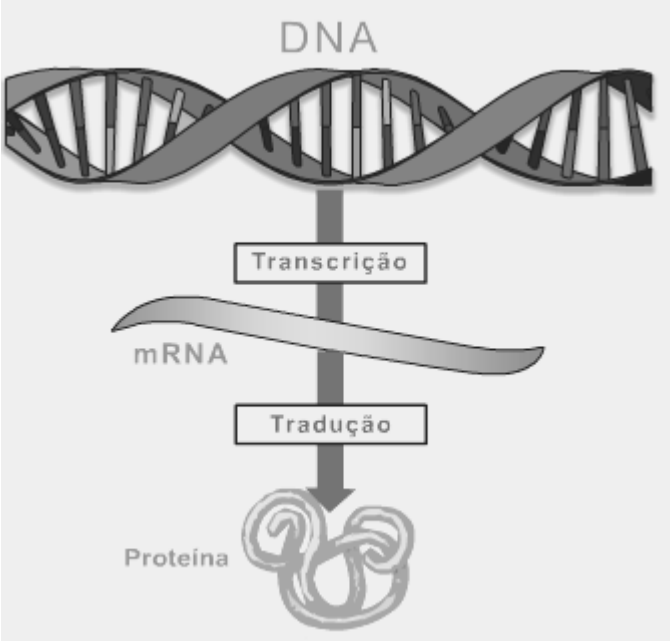

Figura 1: Síntese Protéica [MEI15].

Proteínas são macromoléculas que consistem de uma ou mais cadeias de resíduos de aminoácidos e estão presentes em todos os seres vivos, participando de quase todos os processos celulares.

2.1.2 Microarranjo de Proteínas

A técnica de microarranjo de proteína foi basicamente baseada em cima da tecnologia desenvolvida para microarranjos de DNA.

É uma técnica de larga escala utilizada para identificar as interações e atividades de proteínas e principalmente determinar sua função. Este método é eficaz pelo fato de ser possível buscar simultaneamente uma enorme quantidade de proteínas e consiste em conectar um conjunto de proteínas (microarranjo) a uma superfície de suporte, como membrana nitrocelulose ou uma lâmina de vidro [MEL04]. Um exemplo de análise de microarranjo pode ser visto na Figura 2.

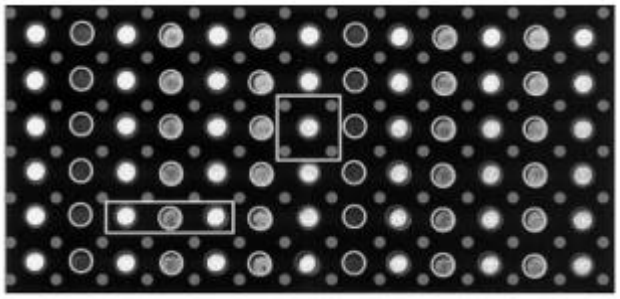

Figura 2: Microarranjos de proteínas [PAW02].

A superfície de suporte serve basicamente para as proteínas ficarem imobilizadas, então é injetado um pigmento que reage aos estímulos das proteínas imobilizadas, emitindo um sinal fluorescente que é registrado por um scanner a laser. Na Figura 3 pode-se observar um instrumento de leitura e análise de microarranjos (ZeptoChip), o qual proporciona flexibilidade e automatização nas análises.

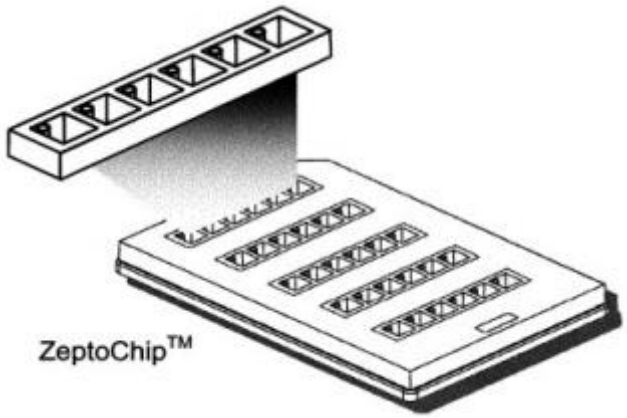

Figura 3: ZeptoChip. Analisador de microarranjos. [PAW02].

Hoje há cinco principais áreas onde microarranjos de proteína são aplicados: Diagnóstico, análise funcional de proteínas, proteômica, caracterização de anticorpos e desenvolvimento de tratamentos na área, como câncer e alergias.

2.1.3 Ontologia Gênica

A ontologia gênica (GO) é um termo inicialmente da bioinformática para unificar representações de genes e produtos genéticos em todas as espécies [TGO15] [TGO08]. A ontologia gênica visa manter e desenvolver um vocabulário de genes e atributos de produtos genéticos e dispõe de ferramentas para facilitar o acesso dos dados fornecidos, permitindo a interpretação funcional de dados experimentais usando o GO. A ontologia abrange três domínios disjuntos da biologia molecular:

- Componente celular: se refere ao local onde o produto gênico pode ser encontrado, as partes de uma célula ou o seu ambiente extracelular.
- Função molecular: as atividades elementares de um produto génico a nível molecular, tais como ligação ou catálise.
- Processos biológicos: operações ou conjuntos de eventos moleculares com um começo e fim definidos, pertinentes ao funcionamento de unidades de vida integradas: células, tecidos, órgãos e organismos.

2.2 Aprendizagem de Máquina

Também conhecido como aprendizagem automática, é um campo da inteligência artificial, o qual trabalha com o desenvolvimento de algoritmos que proporcionam

aprendizagem ao computador, ou seja, facilita a máquina realizar alguma tarefa de um modo aperfeiçoado. É fundamental ressaltar que trabalha com o método indutivo, o qual analisa conjuntos de dados e extrai regras e padrões. Será comentado sobre alguns métodos na seção 2.2.5.

2.2.1 Descoberta de Conhecimento

Para realizar a predição de hipóteses pode-se destacar o processo de extração de conhecimento, ou descoberta de conhecimento (KDD). O termo surgiu no final da década de 1980, pois teve uma explosão de volume de dados, devido a esse rápido crescimento criou a necessidade, bem como a grande oportunidade, de extrair conhecimento de uma forma automatizada, basicamente por reconhecimento de padrões de dados relacionados existentes nas bases. Este processo pode ser demonstrado na Figura 4.

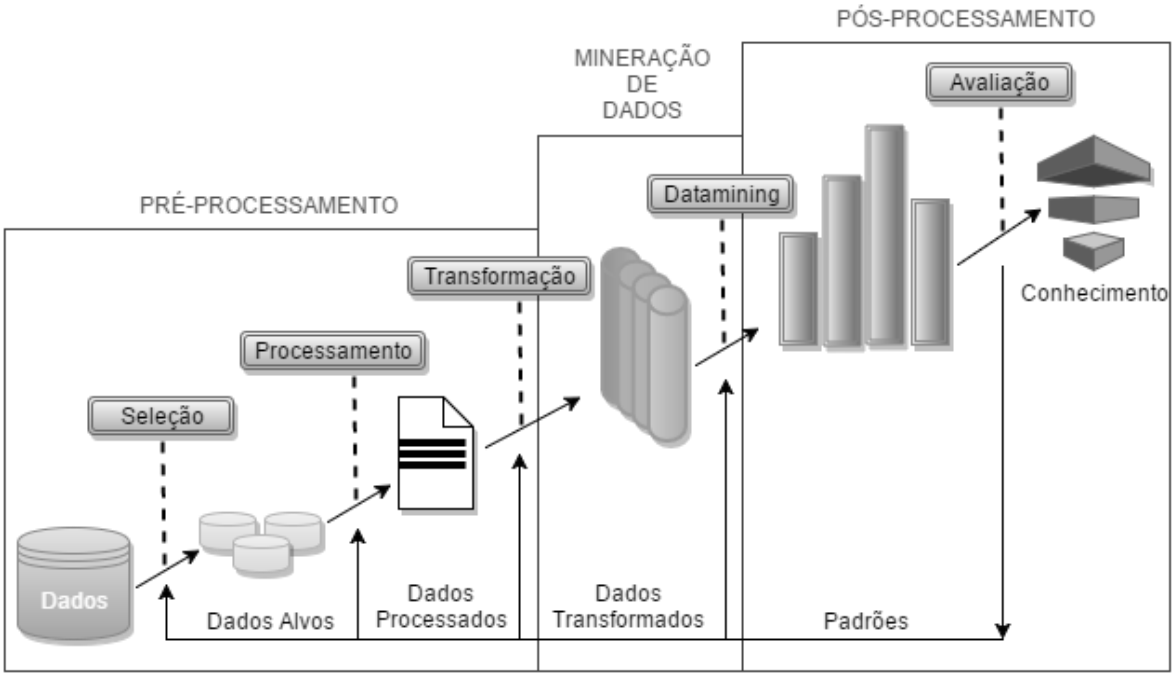

Figura 4: Diagrama de etapas para a descoberta do conhecimento.

Este diagrama pode ser definido, de forma iterativa, pelas seguintes de etapas:

1) Definição do problema

Esta é a etapa que inicia a descoberta de conhecimento e precede a iteratividade do diagrama, Figura 4. Aqui será feito o mapeamento e entendimento do problema, bem como

levantamento de requisitos e análise de atividades, definindo objetivos, assim possibilitará que os resultados realmente sejam uteis. Irá ser trabalhada, também, nesta etapa o tempo de vida da solução, que poderá se adaptar, identificar entidades e será definido o que será desenvolvido nas demais etapas.

2) Pré-Processamento

Nesta etapa são feitas seleções e segmentações dos dados adequados para a análise de acordo com os critérios selecionados, em suma é a preparação dos dados. Dentre os principais itens, destacam-se:

- Definição de atributos: Existe um armazenamento dos dados brutos e um conhecimento prévio do domínio, então é necessário definir quais atributos são de real importância para alcançar a meta do processo de descoberta de conhecimento. A definição de atributos em primeira instância é feita de forma manual, quando é selecionado um subconjunto do total de atributos disponível por um especialista na área, então em caso de qualquer desconhecimento, como exemplo: na dúvida de se a retirada do atributo irá ocorrer imprecisões no processo, é melhor mantê-lo. Pois algoritmos de aprendizados têm facilidade em trabalhar com atributos extras, mas para compor novos atributos que facilitem a predição, possuem dificuldades no processo.

- Extração e integração: Os dados podem se encontrar em vários locais de armazenamento, como base de dados, discos e arquivos, então é essencial realizar a extração e integrar os dados novos ao formato utilizado. Alguns casos de bases relacionais podem requerer junção de tabelas.

- Transformação de dados: Pode haver dados de locais externos que não tem a mesma formatação da base utilizada, deste modo é necessária a conversão destes para a fácil leitura dos algoritmos de aprendizagens de máquina. A transformação de dados pode se aplicar a campos como data, para um número inteiro ou períodos, caso haja necessidade, e até mesmo normalização de valores, ou seja, para obter um melhor resultado alguns algoritmos utilizam intervalos de valores específicos, por exemplo, valores inteiros entre 0 e 1. Algumas operações de transformação de dados são: redução de escala, extensão da escala, conversão de unidades, normalização de valores e adaptação de conjunto de dados.

- Limpeza: Este item pode ser dividido em 2 grupos: o primeiro é a limpeza do domínio, o qual retira inconsistências de atributos ou atributos que estejam duplicados indevidamente. O

segundo é a limpeza independente do domínio, o qual se encontra nos ruídos presentes nas amostras do conjunto de dados, que por algum motivo, são diferentes do resto dos dados. Geralmente são removidos no processo de limpeza, como mostra na Figura 5.

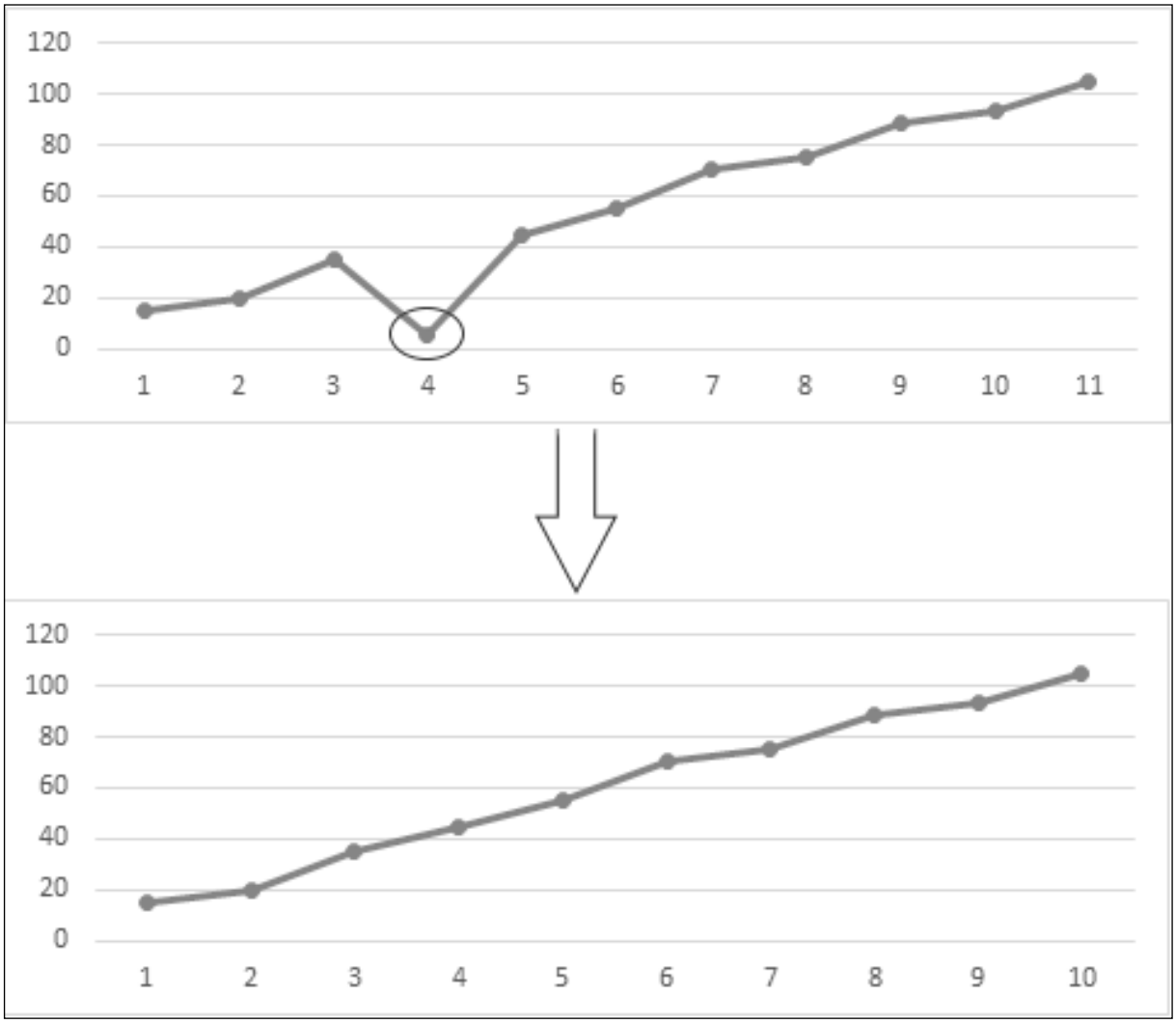

Figura 5: Remoção de um ruído da base.

- Composição: Este item é uma determinante muito importante para gerar resultados de qualidade, pois trabalha com a composição de atributos, ou seja, é o processo que constrói novos atributos, que se tornam diretamente relevantes, a partir de atributos primitivos, mostrado na Figura 6. Em alguns casos é feita a composição de atributos antes do método de seleção de atributos.

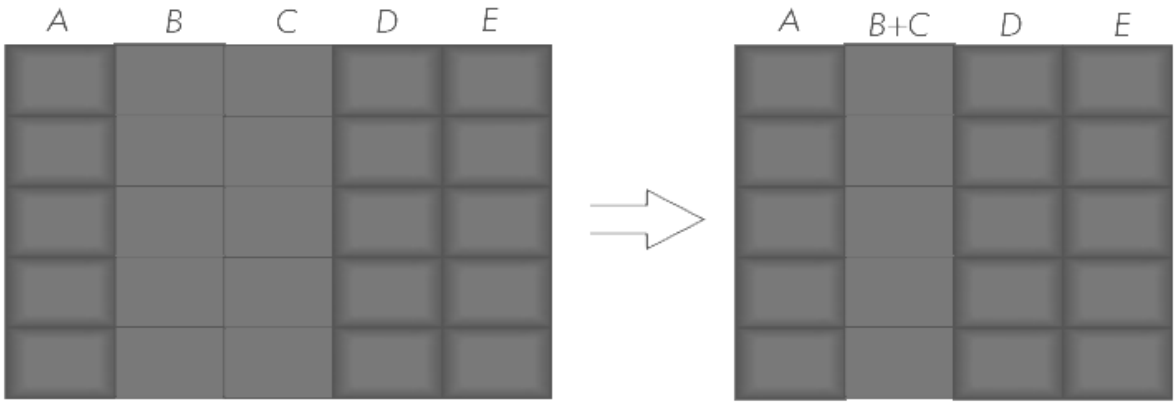

Figura 6: Composição: Junção dos atributos B e C.

- Redução: Considerando que já foi feito uma preparação de dados e teve como saída um conjunto de tamanho moderado, pode aplicar algoritmos de mineração de dados nos conjuntos exemplos gerados. No entanto, se forem conjuntos muito grandes de dados, seria interessante utilizar a redução antes da mineração. Essa redução pode ser feita através da retirada ou junção de um exemplo na base ou até mesmo um atributo, que pode acarretar em maior impacto.

3) Mineração de Dados

Mineração de dados é a etapa que explora grandes quantidades de dados na busca de padrões e tendências que existem nos dados. Nesta fase é utilizada o conjunto de técnicas e ferramentas que dispõem de algoritmos de aprendizado de máquina para explorar os conjuntos de dados, assim auxiliar na descoberta de conhecimento. As formas de apresentação são diversas, como arvores de decisões, esquema de regras e grafos.

O ser humano sempre descobriu padrões através de formulação de hipóteses, testes e criação de regras. Porém se tornou um exercício muito desgastante e complexo identificar padrões em grandes bases de dados, então foi utilizado técnicas de mineração de dados no computador para descobrir informações novas e úteis [COR11].

4) Pós-Processamento

Esta é a etapa que é analisada a qualidade os dados obtidos na fase de mineração de dados. A partir disto será possível determinar se os dados obtidos são ou não uteis para o problema, caso negativo, pode-se ter a necessidade de retornar a alguma etapa anterior e refazer

alguns processos para minerar um conjunto mais válido de dados, pois há uma forte interatividade e iteratividade entre todas as etapas.

Vale ressaltar que nesta etapa também é feita a interpretação dos resultados, ou seja, serão avaliados os padrões descobertos, existirá uma visualização dos dados, remoção dos padrões que não importam ou são redundantes e é feita a tradução dos padrões selecionados. Deste modo observa-se que são desenvolvidos processos nesta fase que melhoram a compreensão do conhecimento descoberto.

2.2.2 Tarefas de Descoberta de Conhecimento

Existem diversas tarefas da mineração de dados para a descoberta de conhecimento, entre elas estão: classificação, agrupamento, regressão e associação.

- Classificação: A tarefa de classificação consiste em categorizar dados em classes previamente definidas, normalmente, de acordo com a existência de similaridades entre as características dos dados. A classificação é usada para prever a classe de um objeto baseado em seus atributos [BUE12].

Seu funcionamento baseia-se no mapeamento do conjunto de dados de entrada em um número finito de classes, buscando correlações entre os atributos e a classe. Deste modo, os atributos com correlação são utilizados para realizar a predição da classe de um novo registro, cuja classe é desconhecida. Seu principal objetivo é aumentar a quantidade de classificações corretas nos dados teste para ter um alto grau de predição em novos dados.

- Agrupamento ou Clusterização: Na tarefa de agrupamento existe a participação de dados heterogêneos que são particionados em vários grupos mais homogêneos. Desta forma, no agrupamento, não existe uma classe certa, os registros são agrupados de acordo com sua semelhança, o que diferencia da tarefa de classificação. Um algoritmo de agrupamento, obtém as características dos exemplos e agrupa os que tem um grande grau de semelhança e separa grupos com menos semelhanças. Geralmente essa tarefa é executada precedendo outras formas de mineração, como métodos de classificação, com intuito de criar regras.

- Regressão: A regressão não é tão utilizada como a tarefa de classificação, pois a maioria dos problemas não se encaixam nessa área. Porém, a tarefa de regressão, em termos conceituais, é muito similar a classificação, tem como a principal diferença a predição gerada, sendo esta continua e não discreta.

O principal objetivo é desenvolver um mapeamento entre a função dos atributos de entrada e um atributo-meta. Por exemplo, seja x um atributo de entrada em que $x = \{x1, x2, \dots, xn\}$, faz-se o mapeamento com y, o atributo-meta, da seguinte forma: $y = f\{x1, x2, x3 \dots, xn\}$.

- Associação: Na associação, o objetivo é encontrar regras, ou seja, uma regra de associação pode ser caracterizada quando um conjunto de itens implica no aparecimento de um outro conjunto de itens distinto, se propondo a encontrar todas as associações relevantes entre esses conjuntos.

Esta tarefa pode ser facilmente encontrada em planejamento de inventário, supermercados e planos de vendas, pois precisam ser determinados padrões de comportamentos. Como no exemplo do supermercado, pode se citar a situação de um perfil de consumidores que compram cerveja, também compravam fraldas nas sextas-feiras. Assim, com base nessa hipótese, seria viável este supermercado colocar as fraldas e as cervejas próximas nas sextas-feiras.

2.2.3 Redução de Dimensionalidade

A mineração de dados atual busca seu conhecimento em bases de dados relativamente grandes, desta forma, existe um número significativo de atributos também. Alguns classificadores, geralmente, tem uma perda de desempenho quando são executados e em suas bases existem muitos atributos irrelevantes, causando assim um problema para o processo de aprendizado. Um bom exemplo disto, são os atributos selecionados durante a geração da arvore de decisão, pois, à medida que esta arvore de decisão é montada, menos são os dados disponíveis para ajudar na escolha dos atributos, então, chega em um ponto que atributos aleatórios aparecem apenas por acaso e a chance de isto acontecer cresce bastante ao aumentar da profundidade da árvore.

Desta forma, é necessário utilizar algoritmos a fim de selecionar os atributos mais apropriados para a tomada de decisão. Neste sentido entra a seleção de atributos, que visa melhorar e acelerar o desempenho do aprendizado, produzindo assim uma versão mais condensada do conceito a ser aprendido. O objetivo é selecionar um subconjunto de atributos que realmente são importantes para fornecer ao classificador.

A redução de dimensionalidade não abrange apenas a redução de atributos, mas também se aplica em exemplos da base, mostrado no fluxo da Figura 7, pois dependendo dos atributos

selecionados, os exemplos podem ser redundantes ou insignificantes para o aprendizado. Como exemplo, pode-se citar bases que suas classes são divididas para uma análise por nó da estrutura, desta forma é necessário saber se o exemplo pertence ou não aquela classe, então são selecionados apenas os que constam para executar os classificadores.

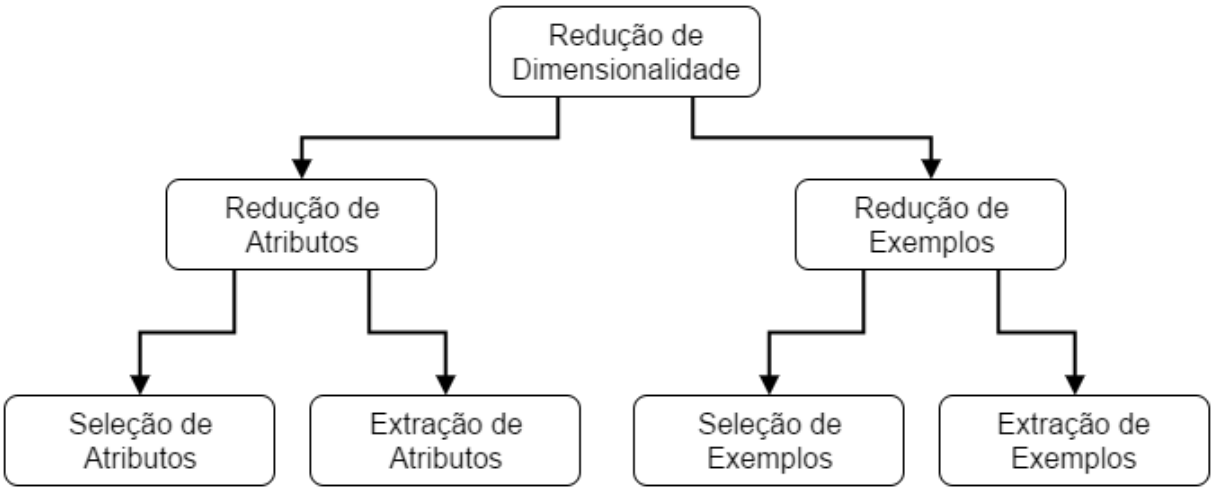

Figura 7: Redução de dimensionalidade de atributos e exemplos.

Os algoritmos que são utilizados na seleção de atributos podem ser divididos em dois passos principais: a busca do subconjunto de atributos e a avaliação destes subconjuntos de atributos. A busca pode ser feita de três formas: a primeira é conhecida como busca para frente, iniciando com um conjunto vazio, então são adicionadas características. A segunda é a busca para trás, sendo o contrário da busca para frente, iniciando com um conjunto cheio e sucessivamente são removidas as características. A última, a busca bidirecional, é uma mistura das duas primeiras abordagens, iniciando com o conjunto vazio ou cheio, para então adicionar e remover características simultaneamente.

Existem duas principais abordagens para a avaliação do subconjunto: *wrapper* e *filter*. A abordagem *wrapper* tem a necessidade de um algoritmo de classificação, utilizando ele como um critério de avaliação. Em suma, esta abordagem, seleciona o seu melhor conjunto de características, o qual irá melhorar o desempenho do classificador. Na abordagem *filter*, diferente da abordagem *wrapper*, não é necessária a utilização do algoritmo de classificação, pois a relevância dos atributos é estimada baseada nas características gerais dos dados.

Na Figura 8 pode-se observar um fluxo com interação entre as etapas da redução de dimensionalidade, de acordo com Dash e Liu [DAS97].

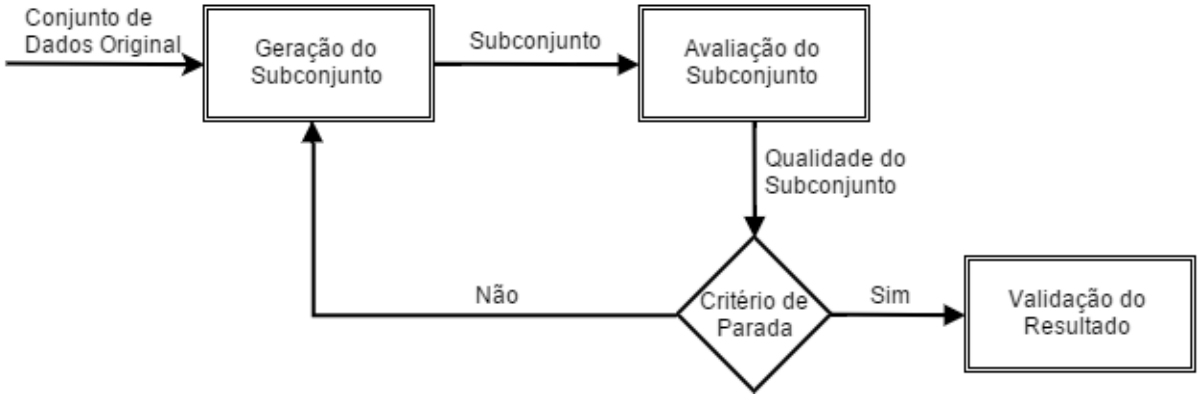

Figura 8: Etapas para o processo de seleção de atributos baseado em Dash e Liu [DAS97].

No processo de encontrar um subconjunto de atributos é selecionado um conjunto de variáveis mais relevantes com a ajuda de algoritmos de busca. Na avaliação dos subconjuntos de atributos é feita a medição de quão relevante é um determinado atributo dentro do conjunto selecionado.

O método de avaliação C*FS* (*Correlation based Feature Selection*) [HAL99], o qual é um avaliador de valores no subconjunto de atributos do tipo *filter*, considera a capacidade individual de previsão de cada característica da base, em conjunto com o grau de redundância entre eles. Vale ressaltar que subconjuntos que possuem características com grande grau de correlação e que suas classes tem um pequeno grau de intercorrelação, são os mais relevantes para o método.

Em conjunto com o método CFS é utilizado um algoritmo de busca. Um exemplo é o algoritmo *Greedy Stepwise,* o qual seleciona variáveis por etapas. Seu funcionamento gira em torno da possibilidade de poder executar suas pesquisas movendo-se para frente ou para trás, através do espaço de subconjuntos de atributos. Pode começar com todos os atributos ou sem nenhum, de um ponto arbitrário no espaço. Sua parada acontece quando é adicionado ou excluído todos os atributos restantes, também pode produzir uma lista ordenada de atributos, atravessando o espaço de um lado para o outro, gravando assim a ordem que os atributos são selecionados.

2.2.4 Classificação de Dados

A classificação de dados permite que sejam extraídas informações a partir de um conjunto de dados brutos, os quais no processo são categorizados. Basicamente é determinada

a classe de novos exemplos a partir de exemplos treinados com classe rotulada. Diversos são esses classificadores e serão descritos alguns deles a seguir.

2.2.4.1 Classificação Baseada em Árvore de Decisão

Arvore de decisão é definida como uma estrutura que se assemelha a uma árvore, são produzidas por algoritmos que identificam formas de dividir um conjunto de dados em segmentos semelhantes a ramos, onde suas folhas correspondem a uma classe ou um nó de decisão, o qual carrega o teste sobre algum atributo, dependendo do resultado dessas decisões, irá ser provido uma aresta para uma subarvore. Pode-se observar um exemplo na Figura 9.

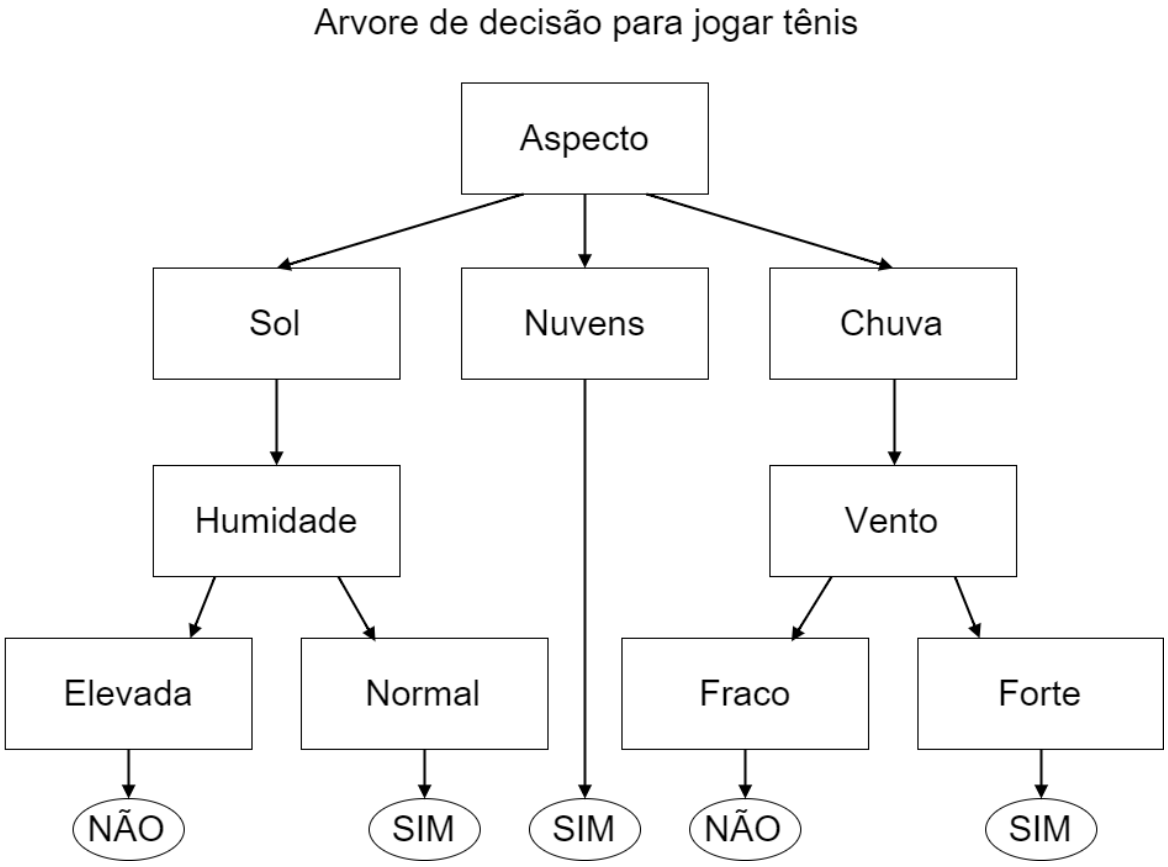

Figura 9: Arvore de decisão para jogar tênis.

A Figura 9 permite fazer conclusões sobre a possibilidade de jogar tênis ou não, baseado no tempo. Pode ser citado como exemplo a condição: se a perspectiva do tempo estiver com chuva e o vento estiver fraco, então não se deve jogar tênis.

A classificação a partir de arvore de decisão se dá a partir da raiz, percorrendo a estrutura para chegar em algum nó folha, o qual irá propor a classificação. Neste ponto ao fazer o teste

em um nó da arvore, caminha para o ramo inferior correspondente ao resultado do teste, então repete-se esse processo para o próximo nó.

O algoritmo C4.5 é utilizado para criar arvores de decisões a partir de um conjunto de dados de treinamento utilizando o conceito de entropia, sendo que os atributos selecionados são tidos como raízes das subarvores. O atributo que melhor faz a divisão dos subconjuntos de amostras é colocado no nó da arvore, ou seja, quanto maior o ganho de informação mais propício a ser escolhido para tomar a decisão. Esse algoritmo repete essa etapa nas partições menores, podendo haver podas (quando o algoritmo entende que um ramo da arvore é desnecessário).

2.2.4.2 Classificação Baseado em Regras

A finalidade da classificação por regras é extrair do conjunto de exemplos as regras neles contidas, geralmente utilizando a forma gulosa (conhecida como *greedy*) para obtê-las. O conjunto que contém as regras inicia vazio, à cada iteração é feita a análise de uma classe, registrando esta como positiva e todas as outras classes como negativas. Após a varredura das classes é selecionada a melhor regra, geralmente a que cubra o maior número de exemplos positivos, e adicionado ao conjunto de regras.

O algoritmo *Ripper* é um dos mais utilizados para a extração de regras. Esse método, quando trabalhado com multiclasses, ordena as classes de acordo com a frequência no conjunto de treinamento e seleciona a com maior frequência como classe *default*, para as demais classes há uma iteratividade no algoritmo gerando suas regras, iniciando pela menos frequente.

Esse algoritmo é ótimo para construção de modelos com conjuntos de dados desbalanceados entre número de exemplos das classes, sua forma de avaliação previne o ajuste excessivo da hipótese, trabalhando assim, muito bem com ruídos.

2.2.4.3 Classificador Baseado no Teorema de Bayes

Na classificação bayesiana é visado minimizar a probabilidade de erros de classificação, ou seja, trata a probabilidade de acontecimentos do evento com base nas avaliações feitas, então é ajustada conforme são inseridos novos dados.

O cálculo das probabilidades é feito através do Teorema de Bayes, ela calcula a probabilidade a posteriori expressada pela hipótese baseada na probabilidade a priori dessa hipótese, na probabilidade a priori da evidência e na plausividade entre a causa e o efeito.

Naive Bayes é um dos métodos de classificação bayesiana. Este classificador também é chamado de ingênuo, por assumir que os atributos de entrada são condicionalmente independentes, ou seja, informações de um evento não detém informações de nenhum outro evento. Por exemplo, um automóvel pode ser considerado como um caminhão, se tem mais de 6 rodas, transporta carga e maior que 6 metros de comprimento. Um classificador Naive Bayes considera estas características citados para, de forma independente, contribuir para a probabilidade de que este automóvel seja um caminhão, independentemente da presença ou ausência de outras características.

2.2.4.4 Redes Neurais

Em aprendizado de máquina, redes neurais são modelos computacionais que se inspiram na estrutura neural de organismos inteligentes, que podem adquirir conhecimento através da experiência. Geralmente são representadas por ligações entre neurônios, os quais são os nós da estrutura; esses neurônios têm pesos entre os nós e são ajustados conforme é feito o aprendizado da rede.

Existem diversas camadas de processamento em uma rede neural, as quais estão conectadas por canais de comunicação, estes canais são associados por um peso. Cada camada faz operações com seus dados locais, sendo que após o processamento de cada camada é feita uma interação estre elas, montando assim a rede neural artificial. Essas camadas recebem sinais que são apresentados à entrada, esses sinais são multiplicados por um pese, o qual indica sua influência, então é feita a soma ponderada de sinais que produziram um nível de atividade, de acordo com este nível é determinada a resposta da saída.

Multilayer perceptron (MLP) é um exemplo de rede neural com várias camadas de neurônios. Geralmente ela é composta por três ou mais camadas, sendo estas: uma camada de entrada, uma camada de saída e uma ou mais camadas escondidas, totalmente conectadas. No aprendizado é verificado o erro na camada de saída, comparando com o resultado esperado, então é feita a mudança de pesos.

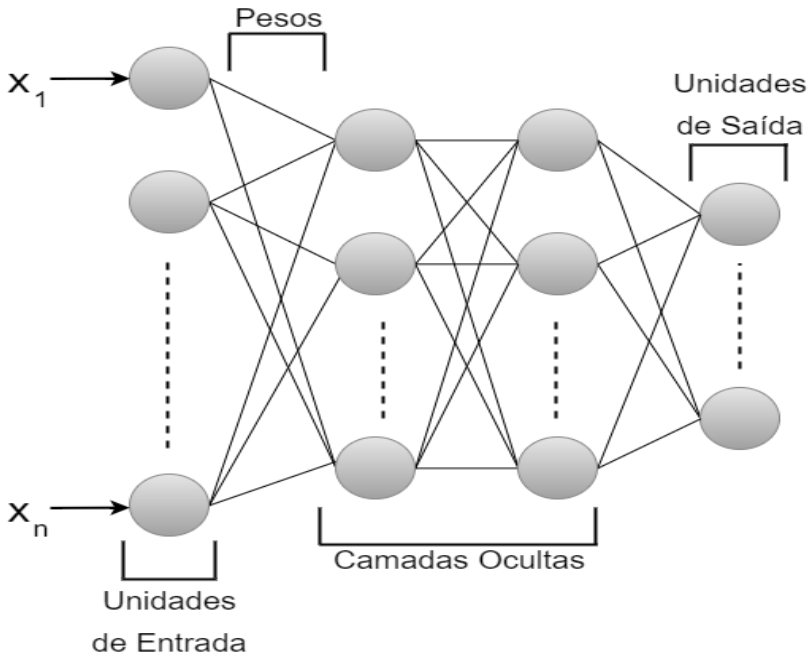

Figura 10: Grafo de rede neural.

Os neurônios formam os nós de um grafo direcionado, como demonstra a Figura 10, sendo que cada camada está inteiramente conectada à uma camada mais próxima (exceto nós de entrada). MLP utiliza uma técnica de aprendizagem supervisionada denominada *backpropagation* para treinar sua rede [RUM86].

2.2.5 Classificação Hierárquica

Muitos dos problemas de classificação são tidos como problemas de classificação plana (do inglês *Flat Classification*). Nestes problemas temos cada um dos exemplos relacionados à uma classe, a qual pertence a um conjunto finito de classes [CER08].

No problema aqui trabalhado, as classes formam uma estrutura hierárquica, ou seja, uma ou mais classes podem ser divididas em subclasses ou agrupada em superclasses; deste modo, as classes formam um grafo acíclico direcionado (DAG) ou uma árvore, entende-se então que os nós dessas estruturas representam as classes do problema. Este tipo de problema é conhecido como problema de classificação hierárquica.

Vale ressaltar que na estrutura de árvore cada nó folha tem somente um nó pai, enquanto que no DAG pode ocorrer a existência de vários pais para apenas um nó folha. Este acontecimento pode ser observado na Figura 11, onde a primeira estrutura corresponde a um

grafo acíclico direcionado, diferindo da segunda estrutura, uma arvore, por seus nós 1.2.1 e 2.1.1 terem dois pais cada um.

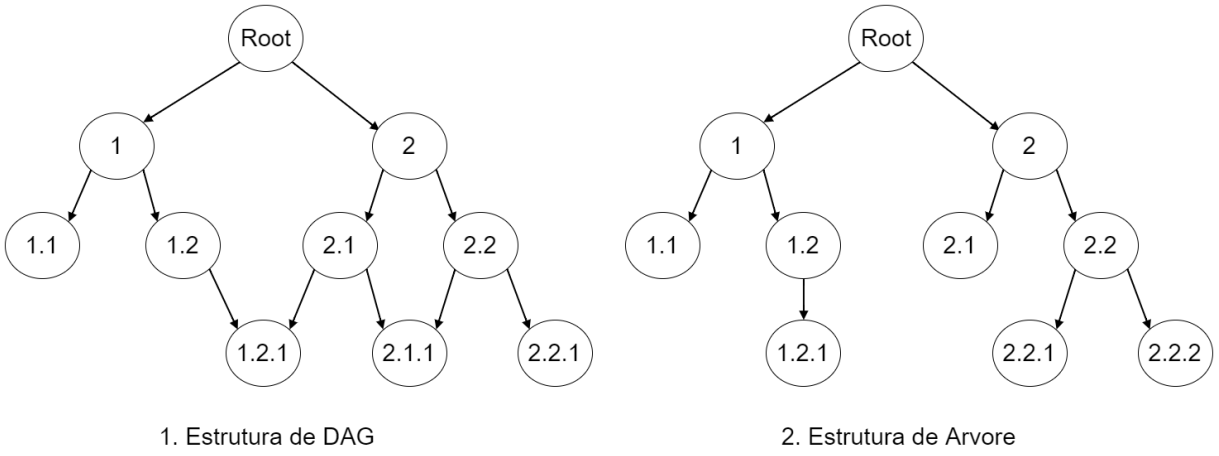

Figura 11: Estruturas de grafo acíclico direcionado e arvore, respectivamente.

O principal objetivo da classificação hierárquica é classificar cada novo padrão, dos dados de entrada, em um nó pai ou folha. Quanto mais profunda é essa classificação, mais específica e útil se torna o conhecimento, porém, muitas vezes a classificação feita em níveis mais profundos pode não ter um nível de confiabilidade aceitável, sendo assim, é mais garantido efetuar a classificação em níveis mais elevados.

2.2.5.1 Abordagens para Problemas Hierárquicos

Várias soluções foram propostas para a classificação de problemas que envolvem hierarquia, porém, três abordagens se destacam: transformação de problemas hierárquicos em problemas de classificação plana, classificação local (também conhecida como *Top-Down*) e classificação global (também conhecida como *Big-Bang*) [COS12]. Serão explicadas, abaixo, um resumo do funcionamento das abordagens citadas.

- **Transformação de problemas hierárquicos em problemas de classificação plana**: Mesmo sendo um problema de classificação hierárquica, esta abordagem reduz o problema originalmente hierárquico para um problema de classificação plana. Nesta abordagem se destacam três principais métodos. O primeiro método é o *Label Powerset*, que a partir de um problema multirrotulado, faz a transformação em um problema monorotulado, no qual cada combinação diferente de rótulos em um conjunto de dados é considerada como um único rótulo [TSO07]. O segundo método é o *Binary Relevance*, o qual divide um problema multirrotulado

em problemas de classificação binária, efetua o treinamento independente de um classificador binário para cada rótulo utilizando [TSO07]. O terceiro método é o *Stacking*, o qual é muito utilizado para combinar diversos modelos. Em suma, o método *Stacking* constrói meta-classificadores, os quais usufruem em suas entradas as saídas de outros classificadores base, para a sua predição [WOL92].

- **Classificação local**: Esta abordagem divide um problema hierárquico em um conjunto de problemas de classificação plana, ou seja, este método consiste em utilizar classificadores planos para gerar modelos de acordo com a estratégia utilizada [COS12]. Dentre as principais estratégias as mais utilizadas são: a classificação por nó, classificação por nó pai e classificação por camada.

Na classificação local por nó um ou mais classificadores são treinados gerando modelos para todos os nós da hierarquia, levando em consideração a informação local, após a classificação de toda a hierarquia é obtido um conjunto de modelos.

A fim de evitar exemplos irrelevantes, podem ser aplicadas podas no momento do treinamento, sendo assim utilizados apenas parte dos exemplos, ou seja, observando a Figura 12, caso o classificador esteja trabalhando no nó 2.2, usa-se seu nó pai como raiz (2), criando uma subarvore, e todos os exemplos que não estejam nessa arvore são retirados.

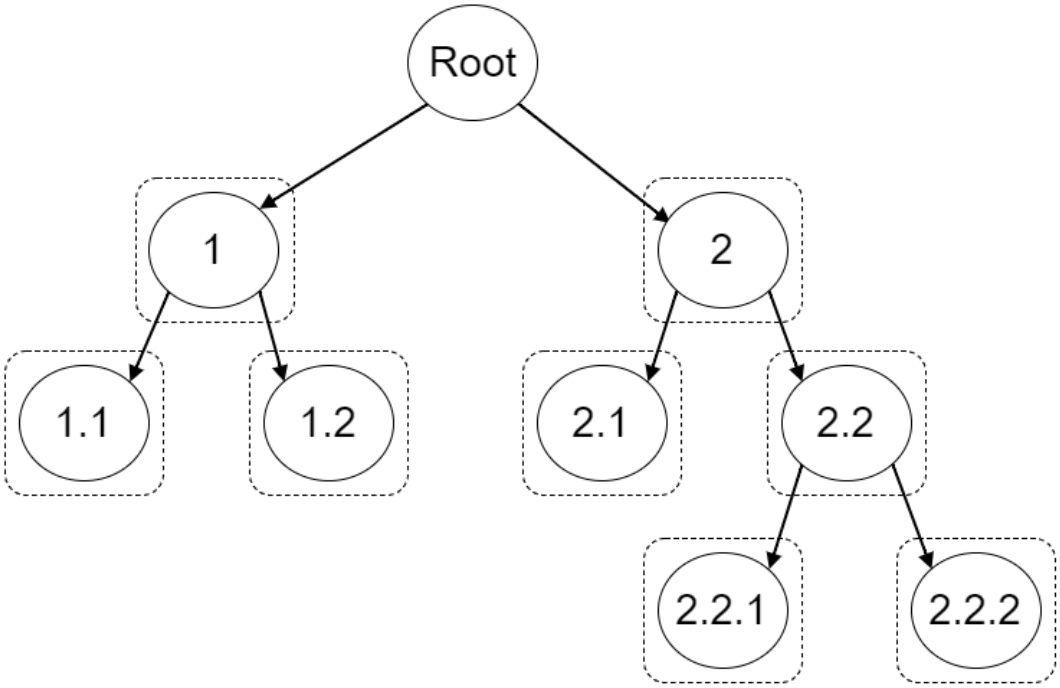

Figura 12: Abordagem de classificação local por nó.

A classificação local por nó pai, também conhecida como *top-down* na literatura, faz o treinamento de um ou mais classificadores planos que se associam a cada classe pai da hierarquia. Nesta estratégia cada classe pai tem um classificador sendo treinado que é baseado apenas em suas classes filhas.

Como demonstrado na Figura 13; o nó pai 2 terá como possíveis predição os nós 2.1, 2.2, 2.2, 2.3 e 2.4.

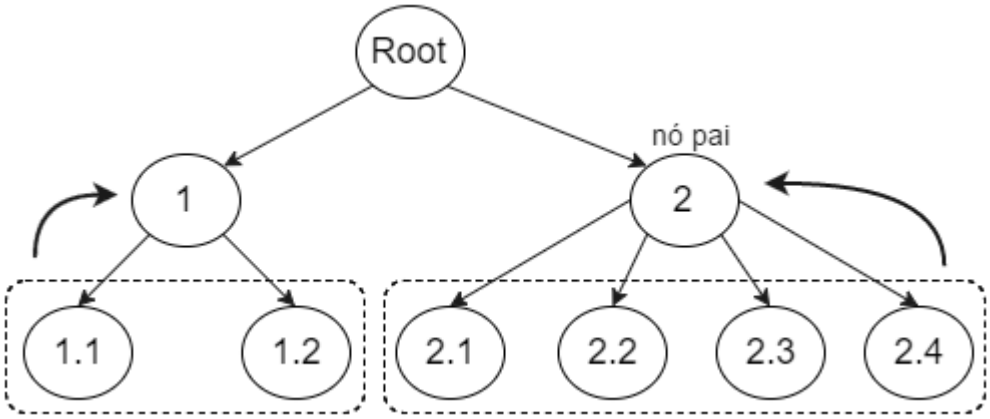

Figura 13: Abordagem de classificação local por nó pai.

A classificação por camada é a abordagem menos utilizada na literatura. Nesta classificação são treinados um ou mais classificadores planos para cada camada da hierarquia de classes.

Considerando a Figura 14, dois classificadores serão treinados, um classificador para cada camada da hierarquia de classes, ou seja, a segunda camada irá explorar todas as classes pertencentes a ela, sendo elas: 1.1, 1.2, 2.1 e 2.2, e o mesmo ocorrerá para as outras camadas presentes na estrutura.

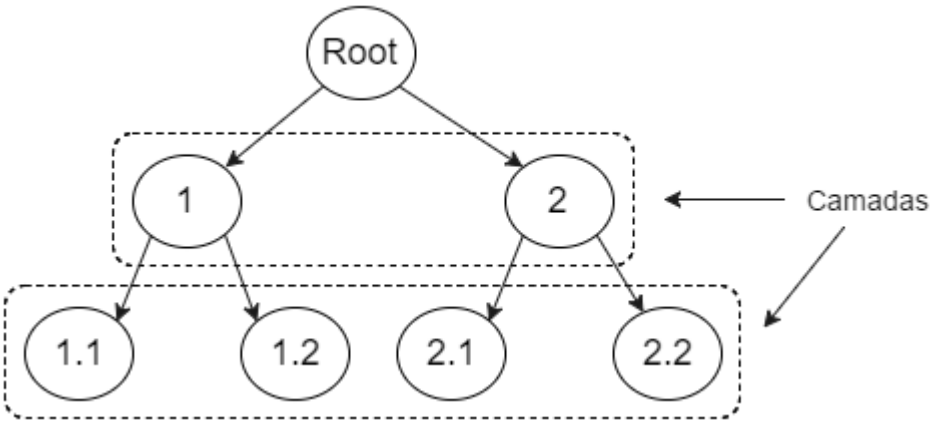

Figura 14: Abordagem de classificação local por camada.

- **Classificação global**: Nesta abordagem é criado apenas um modelo de classificação a partir dos dados de treinamento. Assim, após a fase de treinamento, a predição da classe de um exemplo é feita em um único passo. Neste sentido, pode-se citar uma desvantagem nesta técnica, que possibilita erros cometidos em níveis superiores da estrutura hierárquica, se propagarem aos níveis mais inferiores.

Observa-se na Figura 15, existe apenas um único classificador, o qual irá trabalhar com todas as classes da hierarquia. Se um exemplo for fornecido para a classificação, irá ser composto de apenas uma etapa a predição, sendo que, se a predição for feita pela classe 1.1 a classe pai (1) é predita automaticamente, pois neste modelo a predição é derivada.

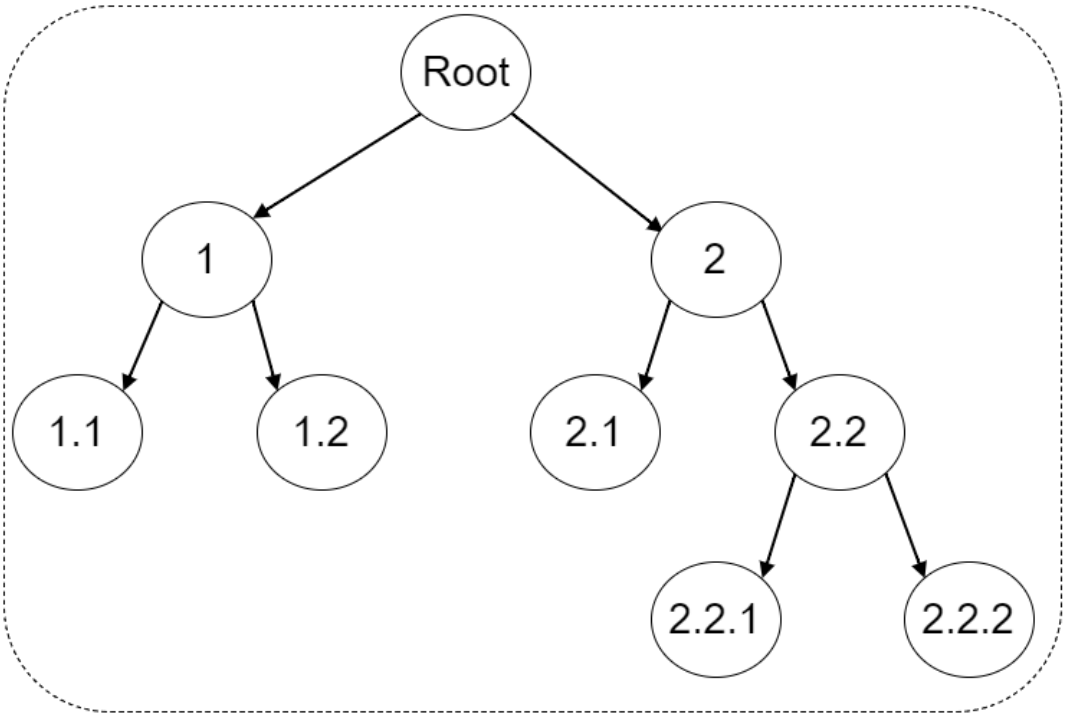

Figura 15: Abordagem de classificação global.

2.2.6 Medidas de Desempenho

Normalmente a forma de avaliação em problemas de classificação plana é feita no formato de uma matriz com os números de exemplos classificados como corretos e incorretos, esta matriz é chamada de matriz de confusão. Um exemplo pode ser visto na Tabela 1.

Tabela 1: Matriz de Confusão.

Matriz de Confusão		
	Classe Prevista	
Classe Verdadeira	Positiva	Negativa
Positiva	VP	FN
Negativa	FP	VN

Os significados dos acrônimos FP, FN, VP e VN são descritos como:

Falso positivo (FP): Exemplos previstos como positivos, os quais são de classes negativas.

Falso negativos (FN): Exemplos previstos como negativos, os quais as classes verdadeiras são positivas.

Verdadeiro positivo (VP): Exemplos previstos corretamente que pertencem as classes positivas.

Verdadeiro negativo (VN): Exemplos previstos corretamente que pertencem às classes negativas.

A medida de avaliação mais utilizada na prática é a acurácia (Acc), também conhecida como taxa de precisão [COS07]. Esta medida resulta na eficácia do classificador pela porcentagem de predições corretas. A equação 1 mostra como Acc é calculada.

$$Acc = \frac{|VN| + |VP|}{|\mathrm{FN}| + |FP| + |VN| + |VP|} \tag{1}$$

Assim, pode-se calcular a taxa de erro (Terr) a partir de Acc, obtendo seu complemento, equação 2, o qual representa a porcentagem de predições incorretas.

$$Terr = \frac{|FN| + |FP|}{|\mathrm{FN}| + |FP| + |VN| + |VP|} = 1 - Acc \tag{2}$$

Neste sentido, para medir a performance do classificador, pode-se citar o *Recall* (Rec) e a *Precision* (Pre). *Recall* é a representação da fração de exemplos corretamente classificados como relevantes em um conjunto de exemplos classificados como relevantes e verdadeiros, representado na equação 3. Por outro lado, *precision* representa a fração de exemplos relevantes

em um conjunto de exemplos relevantes, ou seja, com exemplos relevantes e falsos no conjunto, visto na equação 4.

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

$$Pre = \frac{|VN|}{|\mathrm{FP}| + |VN|} \tag{4}$$

Desta forma, com a combinação do Recall e da *Precision*, é possível obter a F-measure, *equação 5*, a qual representa a média harmônica ponderada destas duas medidas, onde β está contido na faixa [0,1].

$$F - measure = \frac{(\beta^2 + 1) * Pre * Rec}{\beta^2 * P + R} \tag{5}$$

2.2.7 Medidas de Desempenho Hierárquico

Existem várias alternativas para medir o desempenho da previsão de um algoritmo de classificação hierárquica. Geralmente elas são agrupadas em quatro tipos distintos: Baseados na hierarquia, baseados na semântica, dependentes da profundidade e baseados na distância [COS07]. Vale salientar que algumas medidas podem utilizar conceitos de mais de uma abordagem.

1) Medidas Baseadas na Distância

Classes muito próximas na hierarquia tendem a ser mais similares entre elas do que outras classes. Desta forma a abordagem baseada na distância leva em consideração a distância entre classes verdadeiras e classes previstas com intuito de obter o desempenho da classificação hierárquica [WAN99].

Esta abordagem foi utilizada de forma hierárquica por Sun & Lim em *Hierarchical Text Classification and Evaluation* [SUN01]. Estes pesquisadores estenderam as técnicas de medida em classificação plana: *recall*, *precision*, taxa de acerto e taxa de erro para problemas de classificação hierárquica. Vale enfatizar que uma das desvantagens da técnica é a

desconsideração da diferença entre as classes à medida que o nível de hierarquia aumenta, pois em níveis mais baixos se torna mais complicada a classificação que em níveis mais elevados.

O cálculo da medida baseada na distância é fundamentado nos resultados da contribuição de falsos positivos e de falsos negativos para cada classe do conjunto de dados.

$$Con(x, C_p) = 1 - \frac{\mathrm{Dis}(C_p, C_t)}{Dis_\theta} \tag{6}$$

$$RCon(x, C_p) = \min(1, \max(-1, Con(x, C_p))) \tag{7}$$

$$FpCon_i = \sum_{x \in FP_i} RCon(x, C_p) \tag{8}$$

Na equação 6, para cada classe é calculada a contribuição de cada falso positivo (Con). A variável x representa o atributo da classe e $\mathrm{Dis}(C_p, C_t)$ representa a distância entre C_p e C_t. Então é calculado a contribuição refinada $RCon$, equação 7, normalizando a contribuição de cada exemplo no intervalo [-1,1]. Assim é feito o somatório de $RCon$ para cada falso positivo, obtendo a contribuição falso positivo para cada classe, equação 8.

Na equação 9 pode-se observar o cálculo da contribuição falso negativo $FnCon$ para cada classe C_i, onde esta equação é similar à $FpCon$, porém no cálculo de $RCon$, equação 7, e Con, equação 6, C_p e C_t foram substituídos por C_t e C_p respectivamente.

$$FnCon_i = \sum_{x \in Fn_i} RCon(x, C_t) \tag{9}$$

Desta forma, com os valores de $FpCon_i$ e $FnCon_i$ podem-se calcular as medidas hierárquicas: *precision*, *recall*, taxa de acerto e taxa de erro, as quais podem ser vistos nas equações 10, 11, 12 e 13 respectivamente.

$$HP = \frac{\max(0, |VP_i| + FpCon_i + FNCon_i)}{|VP_i| + |FP_i| + FnCon_i} \tag{10}$$

$$HR = \frac{\max(0, |VP_i| + FpCon_i + FNCon_i)}{|VP_i| + |FN_i| + FpCon_i} \quad (11)$$

$$Tach = \frac{|\text{VN}| + |\text{VP}| + FpCon_i + FnCon_i}{|FN| + |FP| + |VN| + |VP|} \quad (12)$$

$$Terrh = \frac{|\text{FP}| + |\text{FN}| + FpCon_i + FnCon_i}{|FN| + |FP| + |VN| + |VP|} \quad (13)$$

2) Medidas baseadas na profundidade

A abordagem baseada na profundidade, em suma, pondera os níveis da hierarquia, dando pesos maiores para classificações erradas nos níveis mais elevados [BLO02]. Neste método, a distância entre duas classes leva em consideração dois fatores: O primeiro fator é o número de arestas entre as classes previstas e as classes verdadeiras na estrutura hierárquica. O segundo fator é a profundidade das classes verdadeiras e previstas na estrutura hierárquica.

Neste sentido para calcular o erro de classificação associado à diferença entre as classes verdadeira e predita, é realizada a soma dos pesos dos vértices do caminho entre as duas classes. Para garantir que o custo dos erros de classificação nos níveis superiores sejam maiores que em níveis inferiores, é necessário que os pesos dos vértices dos níveis inferiores tendam a ser menores do que em níveis superiores. Esta medida é necessária pois em níveis superiores a classificação é mais confiável.

3) Medidas Baseadas na Hierarquia

Esta abordagem baseada na hierarquia usa conceitos de ancestralidade e descendência para formar novas medidas de avaliação. Esta medida pode ser exemplificada por um grafo com várias classes, as quais possuem nós filhos que são subarvores herdeiras destas classes. A intersecção destas arvores são utilizadas para calcular as medidas *precision* e *recall*. Para calcular a *precision* é feita a contagem do número de classes pertencentes à intersecção e então dividido pelo número de classes pertencentes à subarvore da classe prevista, equação 14.

$$hP = \frac{|Descendente(C_p) \cap Descendente(C_t)|}{|Descendente(C_p)|} \quad (14)$$

Foi definida a variável $Descendente(C)$, a qual representa o conjunto da subarvore do a qual sua raiz é C, a qual essa subarvore representa as classes descendentes de C e a própria classe C. Sendo C_p a classe prevista e C_t a classe verdadeira.

Para calcular a medida *recall*, conta-se o número de classes pertencentes a intersecção e divide-se pelo número de classes pertencentes à subarvore das classes verdadeiras, equação 15.

$$hR = \frac{|Descendente(C_p) \cap Descendente(C_t)|}{|Descendente(C_t)|} \quad (15)$$

Vale salientar que a mesma formula pode ser usada para nós ancestrais, utilizando o conjunto $Ancestral(C)$, porém não é utilizado o nó raiz, pois todos os exemplos são derivados da raiz.

4) Medidas Baseadas na Semântica

A abordagem baseada na semântica utiliza o conceito de similaridade das classes para calcular o desempenho da predição dos modelos de classificação e pode ser usada para definir a *precision*, *recall*, taxa de acerto e erro [SUN01]. Para o cálculo desta abordagem cada classe é representada por um vetor de característica. Por exemplo, sendo C_n representado pelo vetor $V = \{x_1, x_2, x_3, \ldots x_n\}$, onde n é o número de características. Assim a similaridade entre as classes é calculada utilizando estes vetores.

Na equação 16 pode-se observar a similaridade de categorias (SC), onde tem-se os pares de classes Ci e Cj.

$$SC(C_i, C_j) = \frac{\sum_{k=1}^{H} w_k \times v_k}{\sqrt{\sum_{k=1}^{H} w_k^2 \times \sum_{k=1}^{H} v_k^2}} \quad (16)$$

Então SC pode ser utilizado para calcular a média de similaridade de categorias (MSC), equação 17.

$$MSC = \frac{2 \times \sum_{i=1}^{M} \sum_{j=i+1}^{M} CS(C_i, C_j)}{\mathrm{M} \times (\mathrm{M} - 1)} \quad (17)$$

Após o cálculo de SC e MSC de cada categoria, é possível calcular a contribuição (Con) dos falsos positivos e falsos negativos. A equação 18 representa o cálculo dos falsos positivos,

todavia pode ser utilizada para os falsos negativos apenas substituindo C_p e C_t por C_t e C_p, respectivamente. Sendo C_p a classe prevista e C_t a classe verdadeira.

$$Con(x, C_p) = \frac{SC(C_p, C_t) - MSC}{1 - \text{MSC}} \tag{18}$$

Assim, com o cálculo de Con, as etapas para calcular as medidas de avaliação de desempenho são os mesmos descritos no tópico 1: Medidas Baseadas na Distância.

2.3 Trabalhos Relacionados

A utilização do aprendizado de máquina em bases biológicas pode ser muito bem esquematizado e ramificado em *Machine learning in bioinformatics* [LAR06]. Foi feita uma análise de métodos de aprendizado de máquina para a bioinformática, apresentando métodos de modelagem, como: classificação supervisionada, técnicas de agrupamento e modelos gráficos probabilísticos para a descoberta de conhecimento. Vale ressaltar a menção de um quadro de aplicações da utilização de AM nos problemas da biologia, demonstrado na Figura 16.

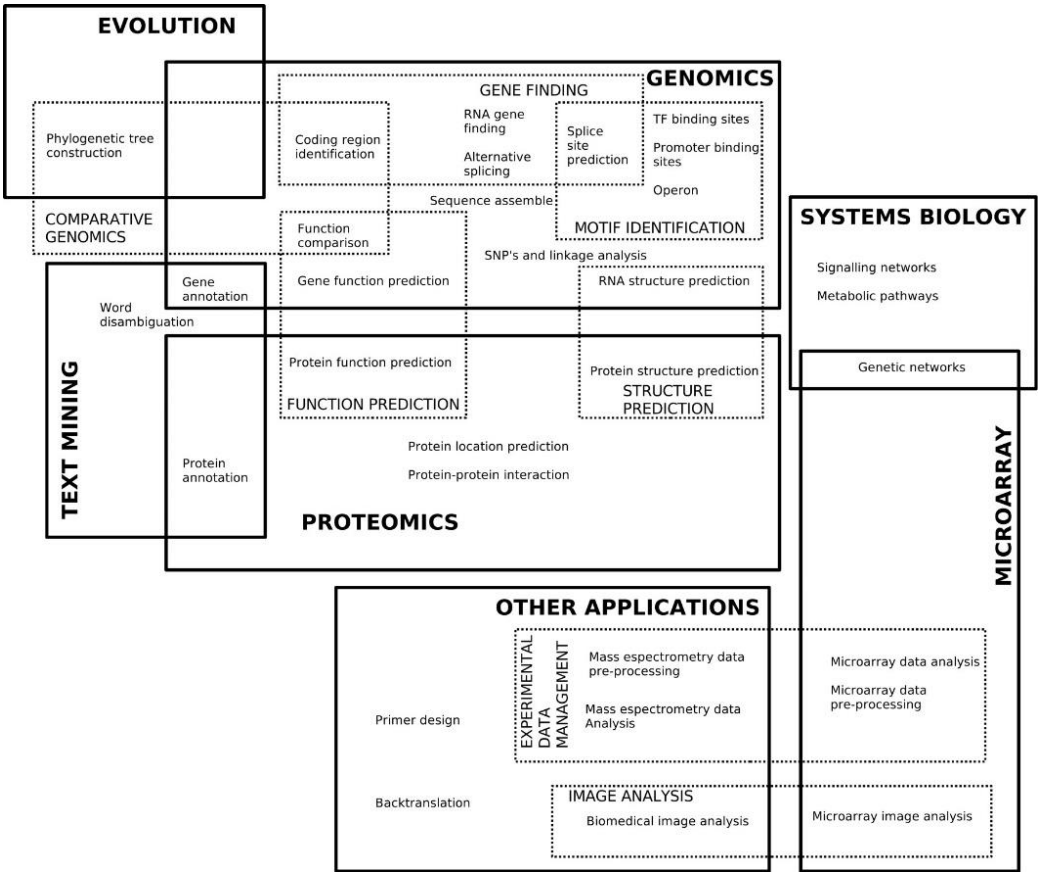

Figura 16: Quadro de classificação dos tópicos onde os métodos de aprendizado de máquina são aplicados [LAR06].

Na Figura 16 os problemas de AM são classificados em seis áreas: genômica, proteômica, microarranjos, sistemas biológicos, evolução e mineração de texto. Há também a categoria de outras aplicações, em que são agrupados os problemas restantes.

Este artigo é muito relevante pois, apresenta as técnicas mais utilizadas para a extração do conhecimento de grandes bases, como classificadores bayesianos, regressão logística, análise discriminante, árvores de classificação, agrupamento hierárquico, redes neurais e outros muito utilizados. Também é concluído em *Machine learning in bioinformatics*, que o artigo pode servir como uma porta de entrada para algumas obras mais representativas desta área e como uma categorização e classificação dos métodos de aprendizado de máquina em bioinformática [LAR06].

Com uma extrema relevância na literatura, é feito um tutorial na classificação hierárquica com aplicações na bioinformática, destacando técnicas existentes, em *A Tutorial on Hierarchical Classification with Applications in Bioinformatics* [FRE08]. Se discute neste

trabalho como as técnicas de classificação hierárquica têm sido aplicadas à área de bioinformática, ressaltando a previsão de funções de proteínas, onde os problemas de classificação hierárquica são frequentemente encontrados.

De acordo com o artigo, existem três critérios de um método de classificação hierárquica: Sua estrutura, podendo ser arvore ou DAG, o segundo critério é de acordo com a profundidade da estrutura, visando o nó que será feita a classificação, onde o classificador pode atuar em novos exemplos como nós folhas (MLNP) ou classificar novos exemplos em qualquer nó da hierarquia (não MLNP). O último critério é sobre a exploração da estrutura hierárquica, a qual pode ser dividida em três tipos: classificadores do tipo plano (*flat*), locais e globais (*big-bang*).

Em classificação hierárquica ainda, pode-se citar mais dois grandes trabalhos: *A Survey of Hierarchical Classification Across Different Application Domains* [SIL11] e *Classificação Hierárquica de Proteínas Utilizando Abordagens Top-Down e Big-Bang* [CER08]. O primeiro trabalho se trata de uma pesquisa sobre classificação hierárquica entre diferentes domínios de aplicação. Freitas e Silla Junior discutem na pesquisa que o campo está muito fragmentado, ou seja, domínios muitas vezes próximos desconhecem o desenvolver de outros domínios. Eles apresentam novas perspectivas sobre certas abordagens de classificação e também propõe um novo quadro que classifica as abordagens existentes, bem como uma revisão das comparações empíricas dos métodos existentes na literatura, discutindo vantagens e desvantagens.

O segundo trabalho demonstra as abordagens *top-down* e *big-bang* utilizadas na classificação hierárquica de proteínas. De forma resumida é possível entender conceitos de classificação hierárquica, métodos mais utilizados e métricas envolvendo arvores de decisão. Dentre as métricas, pode-se cita três principais: Medidas baseadas em distância, em que a medida de desempenho é feita de acordo com a distância entre a classe verdadeira e a classe predita, tendo como uma de suas desvantagens a falta de diferenciação entre distancias de níveis mais profundos e níveis mais elevados. A segunda métrica é a medida dependente de profundidade, a qual tenta contornar a desvantagem da medida baseada em distância, ponderando níveis mais elevados, tornando seus custos de classificação mais altos. A última métrica é a medida baseada na hierarquia, a qual leva em consideração a descendência entre as classes, formando subarvores, e cada subarvore é formada pela classe utilizada e suas classes descendentes. A medida pode ser feita através da intersecção das subarvores [CER08].

Em *Hierarchical classification of Gene Ontology-based protein functions with neural networks* [CER15] foi descrito a classificação multi-rótulo em redes neurais, neste trabalho é demonstrado a classificação de instâncias simultânea para mais de uma classe e devido a dificuldade encontrada na dependência de classes durante o aprendizado, foi proposta a utilização de uma rede neural. Ainda em redes neurais se destacou *Reduction Strategies for Hierarchical Multi-Label Classification in Protein Function Prediction* [CER16], onde é apresentado um novo método de classificação multi-rótulo hierárquico baseado em múltiplas redes neurais para a tarefa de predição da função proteica. Um conjunto de redes neurais é um treinamento incremental, cada um sendo responsável pela previsão das classes pertencentes a um determinado nível, assim chegaram a conclusão nos experimentos que usar a saída em um nível como entrada para o próximo nível contribuiu para melhores resultados de classificação.

Capítulo 3

Metodologia

No desenvolvimento deste trabalho foram realizadas algumas etapas fundamentais para a descoberta do conhecimento contido nas bases de função protéica. Dentre essas etapas vale ressaltar algumas principais que compões o projeto, são elas: o pré-processamento, a redução de dimensionalidade, a classificação hierárquica e a análise dos resultados. Um fluxo geral pode ser visto na Figura 17.

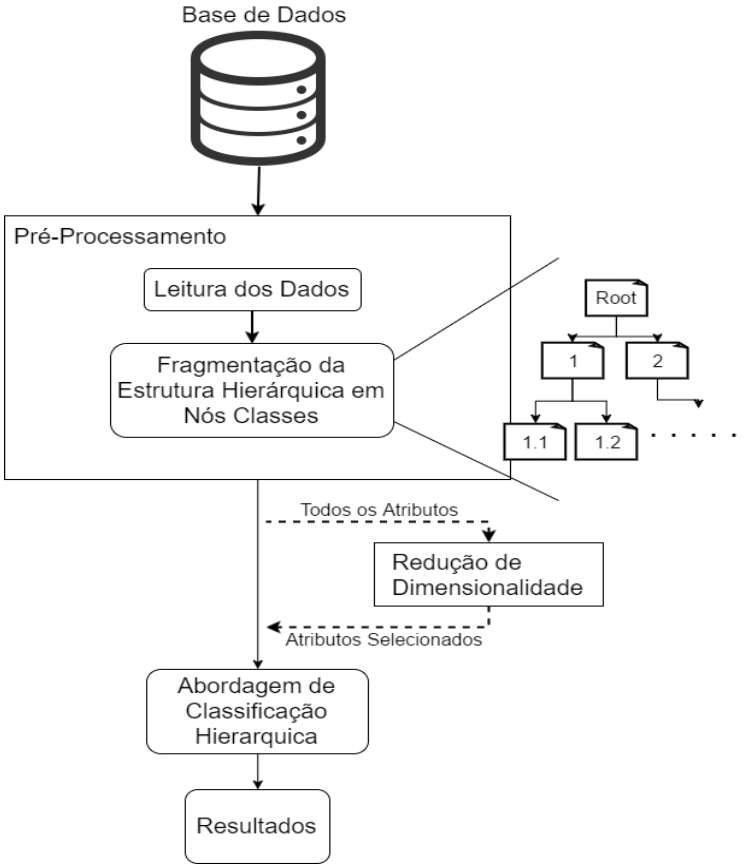

Figura 17: Fluxo com etapas da metodologia.

Para realizar as etapas de redução de dimensionalidade e classificação local, foi utilizado, como auxílio, o pacote do software Weka (Waikato Environment for Knowledge Analysis) para a plataforma Java, o qual é uma plataforma de software para aprendizado de máquina [SCU07] (o conjunto de algoritmos de aprendizado de máquina para tarefas de mineração de dados do software Weka, foi a ferramenta mais adequada para o desenvolvimento dos experimentos).

Vale salientar que foi criado um software na linguagem Java para a automatização da criação da estrutura hierárquica, redução de dimensionalidade e classificação local, existindo a possibilidade de selecionar quais classificadores utilizar, mínimo de exemplos em cada nó classe, a forma de classificação hierárquica e se haverá ou não a seleção de atributos. Esse sistema também grava os valores gerados em um banco de dados para a análise dos dados obtidos nos experimentos.

As etapas da Figura 17 serão melhor detalhadas nas sessões seguintes.

3.1 Conjunto de Dados

Para realizar os experimentos foram selecionadas relações da base de dados GPCR, a qual contém dados da família de proteínas G-Protein Coupled Receptor. As relações foram convertidas para o formato de arquivo ARFF (Attribute-Relation File Format). Este tipo de arquivo é composto por uma lista de instâncias, vistas na Figura 18, que compartilham um conjunto de atributos, Figura 19, ressaltando o atributo classe, o qual é fornecido na forma hierárquica, Figura 20. O formato ARFF foi desenvolvido para ser trabalhado em conjunto com o software de aprendizado de máquina Weka.

```
82  @DATA
83  0.65,1.39,-0.29,-0.54,-0.6,-0.45,-0.13,0.35,-0.01,0.49,0.18,0.43
84  -0.15,-0.71,-0.15,-0.25,-0.31,-0.43,-0.3,-0.23,-0.13,-0.07,0.08,
85  -0.34,-1.06,-0.45,-0.29,-0.36,-0.19,0,-0.32,-0.27,-0.12,0.04,0.1
86  -0.42,0.23,0.14,0.32,0.06,0.01,0.17,-0.14,0.01,-0.24,0.15,-1.34,
87  0.26,-0.17,-0.23,-0.12,-0.24,-0.95,-0.23,0.12,-0.02,0.23,-0.11,0
88  0.26,-0.58,0.06,-0.25,-0.11,-0.17,-0.16,0.04,0.1,-0.02,0.08,0.13
89  0.11,-0.86,-0.22,-0.03,0.04,-0.24,-0.21,-0.23,-0.12,-0.31,-0.06,
```

Figura 18. Exemplo de um conjunto de instâncias em um arquivo arff.

```
75  @ATTRIBUTE elu270       numeric
76  @ATTRIBUTE elu300       numeric
77  @ATTRIBUTE elu330       numeric
78  @ATTRIBUTE elu360       numeric
79  @ATTRIBUTE elu390       numeric
```

Figura 19. Exemplo de atributos em um arquivo arff.

```
80  @ATTRIBUTE class        hierarchical root/GO0003674,GO0003674/GO0003774,
```

Figura 20. Exemplo do atributo classe na forma hierárquica em um arquivo arff.

Neste sentido foram analisadas três relações: GPCRpfam, GPCRprints e GPCRprosite, onde estas relações foram divididas em treinamento (2/3 das amostras) e teste (1/3 das amostras), sendo estas amostras selecionadas aleatoriamente para cada conjunto.

No conjunto GPCRprosite foram analisadas 187 classes, 127 atributos, 4174 exemplos para treinamento e 2087 exemplos para teste. No conjunto GPCRprints foram analisadas 179 classes, 281 atributos, 3615 exemplos para treinamento e 1807 exemplos para teste. Já no

conjunto GPCRpfam foram analisadas 192 classes, 73 atributos, 4718 exemplos para treinamento e 2359 exemplos para teste.

3.2 Pré-Processamento

Durante o pré-processamento foi realizada a formatação e a leitura dos dados contidos nos arquivos ARFF. Desta forma, foi realizada a criação da estrutura hierárquica conforme o atributo classe de cada arquivo.

3.3 Redução de Dimensionalidade

Após o armazenamento na forma hierárquica dos dados, é feita a redução de dimensionalidade. Vale salientar a grande quantidade de atributos presentes nas relações utilizadas, neste sentido, a medida em que a hierarquia é construída menos dados estão disponíveis para auxiliar a escolha do atributo, desta forma em ramificações mais inferiores certos atributos se tornam irrelevantes ou escolhidos ao acaso, fazendo previsões de classes erradas.

Então, na tentativa de melhorar a eficiência da classificação foram realizados experimentos com seleção de atributos e comparados com experimentos sem a seleção de atributos. A seleção de atributos foi aplicada de duas formas diferentes: A seleção de forma hierárquica local, e a seleção de atributos aplicada na base completa antes de iniciar a classificação hierárquica.

Na seleção de atributos foi utilizado o algoritmo CFS com o algoritmo de busca *Greedy Stepwise*. CFS foi escolhida pois trabalha sem a dependência de um classificador, apresenta seu *score* a partir de um subconjunto de atributos e informa o melhor subconjunto encontrado, o que é essencial para manter as escolhas amplas na classificação, pela grande quantidade de atributos. Os algoritmos de busca e validação são vistos na seção 2.2.3.

3.4 Classificação Hierárquica

Pode-se citar três abordagens de classificação local hierárquica empregadas nos experimentos: por nó, por nó pai e por camadas.

Para o desenvolvimento do método de classificação local por nó, foi empregada uma seleção binária dos exemplos de cada arquivo, 1 para classe presente e 0 para classe não presente, para cada nó da hierarquia. Vale salientar que para fins de otimização, foram

utilizados apenas parte dos exemplos, ou seja, foi feita uma poda no nó pai da classe trabalhada, Figura 24. Essa poda evita a utilização de exemplos que não tem qualquer vínculo com a classe e iriam “atrapalhar” o classificador em seu treinamento.

Também foi explorado na abordagem hierárquica por nó a utilização apenas dos exemplos de até dois níveis descendentes do nó trabalhado, pelo mesmo motivo de evitar nós com níveis de relevância no treinamento insignificante.

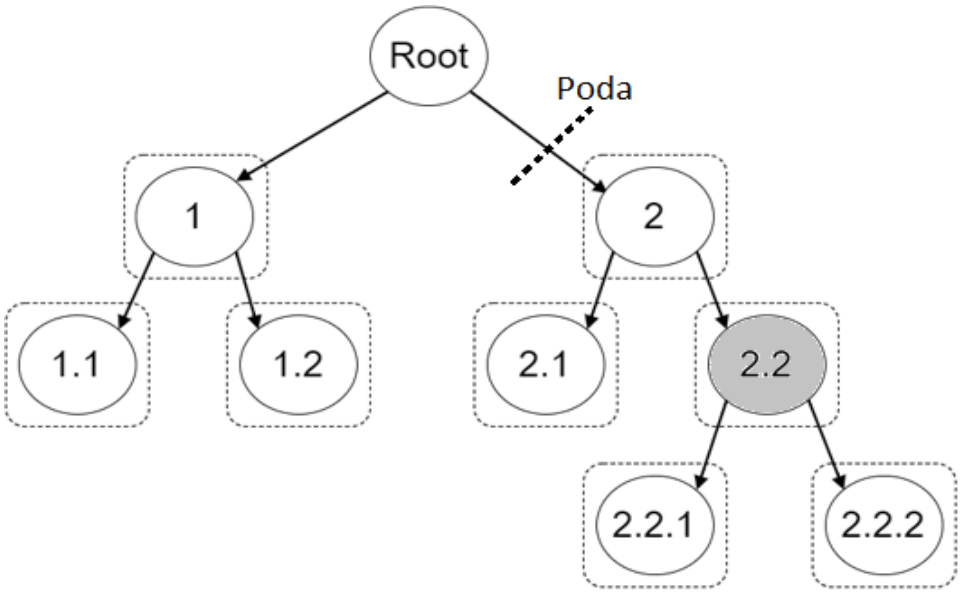

Figura 21: Abordagem de classificação local por nó com poda no nó pai.

No caso da Figura 24, está sendo trabalhado a classe 2.2 e foi feita uma poda em seu nó pai (2).

Vale ressaltar que foi delimitado um número mínimo de exemplos para realizar as classificações planas, igual a cinco, pois o algoritmo Ripper exige uma certa quantidade de amostras para poder executar a tarefa de classificação.

Na classificação hierárquica, o software desenvolvido para executar os experimentos, faz o uso do conjunto de treinamento para realizar a classificação plana e a seleção de atributos em cada nó da estrutura hierárquica de acordo com a abordagem de algoritmo hierárquico selecionada (por nó, por nó pai ou por camada).

Para se ter uma ideia da estrutura criada, pode-se observar a Figura 25, a qual trata da criação dos modelos de cada classificador após a aplicação da classificação plana nos nós classes.

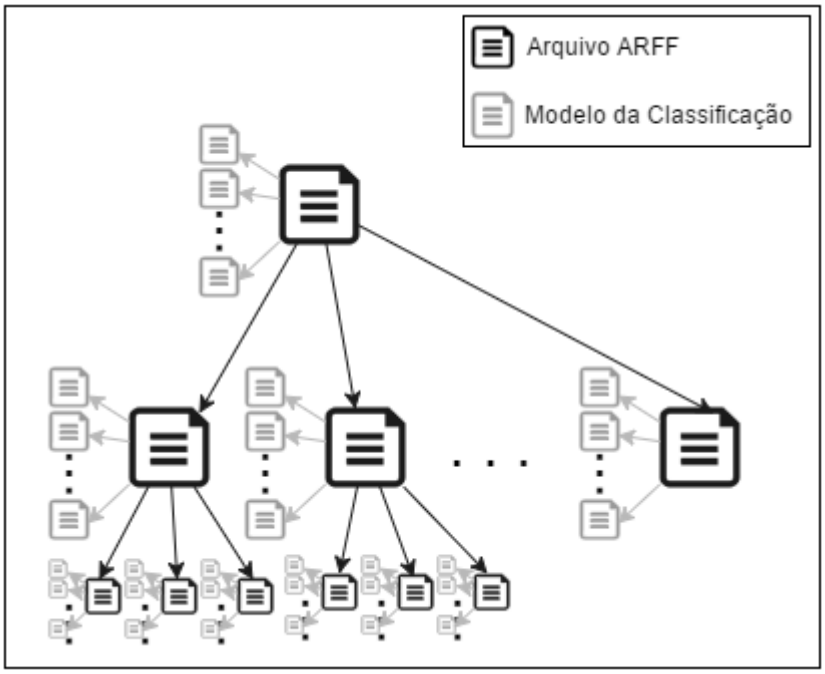

Figura 22: Estrutura modelo após a aplicação dos classificadores.

Após gerar a estrutura hierárquica, para fins de análise, o software utiliza o conjunto de testes para gerar a média hierárquica e o desvio padrão das taxas de acerto.

No conjunto de dados GPCR abordado neste documento, existem quatro níveis na hierarquia de classes. Para manter o valor do grau de erro da classificação para uma determinada função de proteína normalizada no intervalo de 0 a 1 e para fazer com que os valores de peso dos nós diminuam exponencialmente, à medida que é feita a classificação na árvore de classes, foram atribuído pesos para cada camada, sendo eles 0.26 para a primeira camada, 0.13 na segunda camada, 0.07 na terceira camada e 0.04 na quarta e última camada [HOL06].

Capítulo 4

Resultados

Os resultados foram obtidos através da classificação hierárquica das funções de proteínas sobre a base de dados GPCR, utilizando as relações: GPCRpfam, GPCRprints e GPCRprosite. Foram utilizados os classificadores Ripper, C4.5 e NaiveBayes e na seleção de atributos o algoritmo CFS com o algoritmo de busca *Greedy Stepwise*.

Nas subseções abaixo serão vistos os experimentos obtidos através das análises do acerto por nível, média hierárquica e desvio padrão, obtidos através das taxas de acerto em cada nó da hierarquia. Os experimentos seguirão o seguinte padrão: Cada abordagem de classificação hierárquica irá ser aplicada na relação GPCR, com e sem a seleção de atributos local ou por completa, a qual é feita apenas uma vez antes da classificação hierárquica, assim a relação é submetida aos algoritmos de classificação local. Na continuação serão mostradas a análise das comparações entre as técnicas de classificação hierárquica. Finaliza o capitulo com as considerações finais, descrevendo alguns fatos importantes sobre as análises.

4.1 Resultado da Classificação Hierárquica

Os resultados da média hierárquica das relações GPCRpfam, GPCRprints e GPCRprosite são mostrados nas subseções 4.1.1, 4.1.2 e 4.1.3, respectivamente, onde as siglas nas colunas S/A significa sem seleção de atributos, C/A significa com seleção de atributos e C/AU significa com seleção única de atributos, já nas linhas as siglas Med significando média e Des. P significando desvio padrão.

4.1.1 Conjunto de dados GPCRpfam

Nas tabelas 2, 3 e 4 são mostrados os dados da classificação hierárquica por nó, por nó pai e por camada, respectivamente, do conjunto GPCRpfam.

Tabela 2 – Taxa de precisão de acerto hierárquico por nó (%) - GPCRpfam.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	92,67%	91,90%	81,56%	91,61%	91,65%	83,55%	89,32%	89,74%	75,07%
Nível: 2	58,51%	52,08%	49,59%	59,78%	55,57%	54,66%	54,08%	53,62%	44,38%
Nível: 3	13,85%	8,11%	12,69%	17,75%	9,58%	18,00%	8,42%	5,61%	7,38%
Nível: 4	26,75%	13,77%	26,75%	32,93%	16,77%	36,53%	9,18%	9,38%	10,98%
Med.	73,50%	70,52%	62,31%	73,59%	71,45%	65,91%	69,41%	69,26%	55,68%
Des. P.	26,66%	26,97%	33,40%	27,84%	27,26%	33,19%	29,08%	28,49%	35,02%

Tabela 3 – Taxa de precisão de acerto hierárquico por nó pai (%) - GPCRpfam.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	92,67%	91,39%	83,89%	92,33%	91,61%	83,04%	90,08%	90,00%	79,53%
Nível: 2	70,79%	68,61%	62,23%	67,26%	64,81%	58,61%	59,83%	60,14%	51,40%
Nível: 3	38,01%	37,46%	36,18%	29,35%	28,00%	26,72%	18,30%	20,81%	19,10%
Nível: 4	51,50%	49,50%	51,10%	39,72%	37,52%	37,92%	15,17%	16,17%	15,77%
Med.	79,66%	78,07%	70,21%	77,48%	76,04%	67,58%	72,52%	72,80%	62,50%
Des. P.	27,53%	28,81%	34,53%	27,58%	28,25%	34,40%	29,23%	29,35%	35,16%

Tabela 4 – Taxa de precisão de acerto hierárquico por camada (%) - GPCRpfam.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	92,67%	91,39%	83,89%	92,33%	91,61%	83,04%	90,08%	90,00%	79,53%
Nível: 2	72,01%	64,04%	63,27%	59,92%	51,81%	51,18%	60,42%	54,08%	53,89%
Nível: 3	40,82%	39,23%	38,87%	28,19%	27,58%	27,15%	21,90%	22,70%	22,45%
Nível: 4	62,87%	62,87%	62,67%	49,50%	50,10%	49,90%	34,93%	35,33%	35,33%
Med.	82,24%	79,05%	73,14%	77,28%	74,06%	67,71%	75,46%	72,85%	66,08%
Des. P.	23,77%	24,97%	31,60%	25,81%	26,74%	32,77%	26,22%	27,03%	32,96%

Pode-se observar que ao utilizar o classificador local C4.5 na classificação hierárquica por nó pai e por camada tivemos médias maiores, já por nó o classificador local Ripper foi o melhor. Comparando as métricas de classificação hierárquica, a classificação por camada teve o melhor resultado sem a seleção de atributos, utilizando o classificador local C4.5, com sua média de acerto em 82,24%.

Analisando a utilização de seleção de atributos, neste conjunto seguiu um padrão das melhores médias pertencerem a não utilização da seleção (primeira coluna), seguida da

utilização da seleção de atributos por nó da hierárquica (segunda coluna) e por último a seleção única (terceira coluna).

4.1.2 Conjunto de dados GPCRprints

Nas tabelas 5, 6 e 7 são mostrados os dados da classificação hierárquica por nó, por nó pai e por camada, respectivamente, do conjunto GPCRprints.

Tabela 5 – Taxa de precisão de acerto hierárquico por nó (%) - GPCRprints.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	91,31%	89,82%	85,89%	90,76%	89,65%	86,88%	86,05%	87,11%	80,24%
Nível: 2	79,88%	74,20%	73,52%	75,89%	72,63%	70,94%	68,58%	70,99%	65,60%
Nível: 3	48,87%	45,23%	43,04%	44,72%	44,87%	42,24%	30,74%	43,26%	36,78%
Nível: 4	79,14%	73,82%	67,89%	73,62%	74,64%	61,96%	47,03%	72,80%	48,06%
Med.	81,75%	78,75%	75,41%	79,60%	78,06%	74,99%	71,94%	75,73%	68,25%
Des. P.	28,29%	30,03%	33,18%	29,02%	30,22%	32,46%	31,55%	32,35%	36,17%

Tabela 6 – Taxa de precisão de acerto hierárquico por nó pai (%) - GPCRprints.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	91,42%	91,26%	87,94%	91,09%	90,87%	87,49%	88,05%	89,15%	83,23%
Nível: 2	83,70%	80,89%	78,02%	81,17%	77,63%	74,37%	72,18%	75,89%	68,02%
Nível: 3	59,07%	57,32%	55,21%	54,12%	51,20%	49,89%	41,01%	50,18%	43,70%
Nível: 4	86,09%	79,75%	75,26%	79,96%	75,66%	68,71%	64,21%	75,87%	53,99%
Med.	84,33%	83,02%	79,40%	82,72%	80,99%	77,35%	75,96%	79,23%	71,79%
Des. P.	28,25%	28,55%	31,95%	28,68%	29,01%	32,30%	31,00%	30,63%	34,92%

Tabela 7 – Taxa de precisão de acerto hierárquico por camada (%) - GPCRprints.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	91,42%	91,26%	87,94%	91,09%	90,87%	87,49%	88,05%	89,15%	83,23%
Nível: 2	86,45%	80,10%	81,51%	76,67%	68,35%	68,47%	73,64%	72,46%	70,10%
Nível: 3	60,67%	57,83%	57,47%	49,89%	47,05%	45,30%	40,86%	46,61%	45,30%
Nível: 4	88,55%	79,55%	76,48%	85,89%	75,05%	69,53%	56,24%	58,90%	50,10%
Med.	87,01%	85,02%	83,11%	82,47%	79,49%	77,00%	78,13%	78,57%	74,14%
Des. P.	22,91%	22,79%	25,72%	26,17%	26,79%	29,48%	25,54%	27,51%	31,56%

No caso da relação GPCRprints quando se utiliza a classificação local com o algoritmo C4.5 nas abordagens de classificação hierárquica por nó, por nó pai e por camada, os resultados são melhores, onde o classificador por camada se destaca com 87,01%. Analisando por técnica de seleção de atributos, as classificações locais C4.5 e Riper sem a seleção de atributos as

médias ficam maiores, porém com o classificador Naive Bayes com seleção de atributos no formato hierárquico (segunda coluna) os resultados são melhores.

4.1.3 Conjunto de dados GPCRprosite

Nas tabelas 8, 9 e 10 são mostrados os dados da classificação hierárquica por nó, por nó pai e por camada, respectivamente, do conjunto GPCRprosite.

Tabela 8 – Taxa de precisão de acerto hierárquico por nó (%) - GPCRprosite.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	86,68%	78,10%	80,74%	85,77%	77,96%	81,60%	82,13%	75,66%	76,57%
Nível: 2	64,88%	57,68%	49,32%	60,60%	53,16%	52,48%	53,84%	49,46%	52,58%
Nível: 3	21,44%	19,56%	16,84%	22,15%	19,04%	18,78%	12,37%	15,22%	14,25%
Nível: 4	36,53%	37,35%	30,61%	39,18%	33,67%	32,86%	16,53%	20,20%	12,86%
Med.	71,00%	63,40%	61,90%	69,28%	61,93%	63,34%	63,07%	58,72%	59,29%
Des. P.	31,50%	36,34%	34,00%	32,17%	36,03%	33,77%	32,55%	36,24%	35,56%

Tabela 9 – Taxa de precisão de acerto hierárquico por nó pai (%) - GPCRprosite.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	88,02%	84,57%	82,94%	87,06%	83,80%	82,37%	82,61%	80,83%	55,25%
Nível: 2	70,53%	67,75%	63,76%	65,71%	64,40%	59,63%	60,31%	59,78%	34,05%
Nível: 3	41,19%	41,06%	38,02%	31,15%	30,63%	28,95%	26,04%	27,01%	19,95%
Nível: 4	55,51%	54,49%	54,08%	42,45%	41,43%	43,88%	29,80%	27,55%	28,57%
Med.	75,97%	72,70%	69,90%	72,63%	70,03%	67,21%	66,97%	65,69%	43,23%
Des. P.	31,84%	34,44%	35,18%	31,97%	34,22%	34,65%	34,05%	35,18%	41,14%

Tabela 10 – Taxa de precisão de acerto hierárquico por camada (%) - GPCRprosite.

	C4.5			Ripper			Naive Bayes		
	S/A	C/A	C/AU	S/A	C/A	C/AU	S/A	C/A	C/AU
Nível: 1	88,02%	84,57%	82,94%	87,06%	83,80%	82,37%	82,61%	80,83%	55,25%
Nível: 2	70,57%	67,95%	65,95%	61,28%	59,29%	53,31%	64,20%	61,92%	57,83%
Nível: 3	42,49%	40,80%	39,64%	30,96%	30,89%	28,63%	27,85%	30,05%	27,59%
Nível: 4	66,33%	66,33%	64,08%	52,45%	51,84%	54,69%	38,78%	42,65%	42,86%
Med.	78,09%	75,28%	73,35%	72,88%	70,19%	67,35%	71,15%	69,64%	59,99%
Des. P.	28,72%	30,24%	31,45%	30,39%	32,17%	32,86%	29,43%	31,58%	32,32%

Pode-se observar sobre o conjunto de dados GPCRprosite a classificação hierárquica por camada sendo superior estatisticamente em sua média comparando com as demais classificações hierárquicas, sendo notável também o algoritmo local C4.5 superior ao Ripper e Naive Bayes.

Comparando as técnicas de utilização de seleções de atributos, a técnica sem a seleção de atributos nessa base os resultados são melhores, em comparações gerais, na relação GPCRprosite, a seleção de atributos hierárquica é superior a seleção única.

4.1.4 Comparação Entre Técnicas de Classificação Hierárquica

Inicialmente, analisando a comparação entre os classificadores hierárquicos, pode-se perceber que a classificação hierárquica por camada é estatisticamente melhor se comparado com média de acerto hierárquico, em seguida está a classificação por nó pai e em último a classificação por nó, estes dados se refletem para as três utilizações da seleção de atributos. Também é observado que de modo geral a seleção de atributos única gera as piores médias, tendo como a não utilização de seleção de atributos como a melhor técnica.

Vale a pena salientar que existe uma perda significativa de acerto ao descer dos níveis quando foi feita a classificação hierárquica, isto ocorre devido ao grau de especificidade de tornar maior em níveis inferiores. Se vê no nível 3 uma baixa média de acerto, isto devido à maior quantidade de exemplos estarem presentes neste nível e em uma das situações onde se tem muitos exemplos negativos, dificulta o classificador.

4.2 Considerações Finais

Neste capitulo foram apresentados os resultados dos experimentos e as informações relevantes para a realização dos mesmos. Nas avaliações foram utilizadas as bases GPCRpfam, GPCRprints e GPCRprosite.

Inicialmente foram realizados os experimentos separando os resultados das médias hierárquicas por base de dados, onde os algoritmos de classificação hierárquica por nó, por nó pai e por camada utilizaram para suas predições os algoritmos de classificação local de diferentes áreas: baseado em arvore de decisão, baseado no teorema bayesiano e baseado em regras. Sendo estes submetidos a técnicas de utilização da seleção de atributos.

Vale salientar a extrema relevância de realizar as médias por nível e ponderá-los, assim é demonstrado a dificuldade de classificação de um exemplo em um nível inferior, por conter um grau de complexidade maior que em níveis superiores. Também foram exploradas, para meios de comparação, as seleções de atributos por nó hierárquico e uma seleção única.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

Este trabalho apresentou análises de técnicas de classificação local hierárquica usando seleção de atributos para a previsão da função de proteínas. As classes envolvidas apresentam relacionamento hierárquicos entre si, em outras palavras, superclasses herdam funções proteicas de subclasses.

Há uma dificuldade em definir abordagens de mineração para serem utilizadas em bases de ontologia genética, pelo fato de terem relativamente poucos exemplos com uma grande quantidade de atributos, trazendo muitas escolhas incorretas ou aleatórias para a classificação hierárquica em camadas inferiores da hierarquia.

Para as classificações locais hierárquicas foram utilizadas a classificação por nó, classificação por nó pai e classificação por camada, as quais são as mais comuns na literatura. Na seleção de atributos é utilizado uma abordagem tipo *filter*, sendo ela o algoritmo CFS com o algoritmo de busca *Greedy Stepwise*. Para a classificação local foram utilizados os algoritmos de diferentes tipos de classificação, sendo eles: C4.5, Ripper e Naive Bayes.

No geral obteve um destaque na classificação hierárquica por camadas em todas as relações, onde o algoritmo local C4.5 foi melhor que Ripper e Naive Bayes. Comparando a utilização de seleção de atributos, foi observado que a seleção de atributos dificultou a previsão da função de proteína para problemas hierárquicos.

Desta forma, para as condições citadas neste trabalho, no cenário de predição da função de proteína, bases de proteínas (GPCR) que dispõem de suas classes no formato hierárquico, se tornam muito mais favoráveis com a abordagem de classificação hierárquica local por camada e não utilizando a seleção de atributos.

5.2 Trabalhos Futuros

Várias direções podem ser exploradas a partir deste trabalho. Entre elas, pode-se citar a utilização das técnicas de classificação hierárquicas, aqui trabalhadas, em outros tipos de bases, permitindo assim melhorar o nível de confiança destas abordagens de uma forma genérica.

Mesmo que nesse trabalho tenha sido trabalhado a combinação de diferentes técnicas para chegar a predição de uma função de proteína, nada impede do estudo das abordagens aqui citadas com enfoque mais específico.

Outra possibilidade de exploração, é fazer a comparação entre abordagens de classificação local hierárquica e classificação global hierárquica em bases de ontologias gênicas, ambos com aplicação de seleção de atributos, pois ainda é uma barreira escolher quais melhores algoritmos de mineração utilizar para esse tipo de dado.

Referências Bibliográficas

[ALB97] ALBERTS, B. et al. Biologia Molecular da Célula. Artes Médicas, 3ª Edição, 1997.

[BAL01] BALDI, P & BRUNAK, S. Bioinformatics: The machine Learning Approach. MIT Press, 2ª Edição, 2001.

[BLO02] Blockeel, H., et al. Hierarchical multi-classification. In Proceedings of the ACM SIGKDD 2002. Workshop on Multi-Relational Data Mining (MRDM 2002), Edmonton, Canada, Jul. p. 21–35.

[BUE12] Bueno, M. F & Viana, M. R. Mineração de Dados: Aplicação, Eficiência e Usabilidade. Anais do congresso de iniciação científica do INATEL, 2012.

[CER08] CERRI, R., et al. Classificação Hierárquica de Proteínas Utilizando Abordagens Top-Down e Big-Bang. IV Workshop em Algoritmos e Aplicação de Mineração de Dados, 2008.

[CER15] CERRI, R., et al. Hierarchical classification of Gene Ontology-based protein functions with neural networks. Neural Networks (IJCNN), 2015 International Joint Conference, 12-17 July 2015.

[CER16] CERRI, R., et al. Reduction Strategies for Hierarchical Multi-Label Classification in Protein Function Prediction. BMC Bioinformatics, 17 (373). 2016.

[COH04] COHEN, J. Bioinformatics: An Introduction for Computer Scientists. ACM Computing Surveys, Vol. 36, No. 2, Jun. 2004, p 122-158.

[COR11] CORADINE, L.C., et al. Mineração de Dados: Uma Introdução. Learning and Nonlinear Models (L&NLM) – Journal of the Brazil Neural Networks Society, Vol. 9, 2011, p. 168-184.

[COS07] Costa, E. P., et al. A review of performance evaluation measures for hierarchical classifiers. In: Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop, AAAI Technical Report WS-07-05, Jul. 2007. P. 182-196.

[COS12] COSTA, E. P., et al. Comparing Several Approaches for Hierarchical Classification of Proteins with Decision Trees. Advances in Bioinformatics and Computational Biology, 29-31, Ago. 2011, p. 126-137.

[DAS97] DASH, M. & LIU H. Feature Selection for Classification. Department of Information Systems & Computer Science, National University of Singapore. 21, Mar. 1997.

[FRE08] FREITAS, A. & CARVALHO, A. P. L. F. A Tutorial on Hierarchical Classification with Applications in Bioinformatics. Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications. Wang, J. IGI Global, 31, Mai. 2008, p. 119-145.

[HAL99] HALL, M. A. Correlation-based Feature Selection for Machine Learning. 198f. 1999. Thesis (PhD in Computer Science) – Waikato University.

[HOG11] HOGEWE, P. The Roots of Bioinformatics in Theoretical Biology. PLoS Comput BioL, 7-10 Mar. 2011.

[HOL06] HOLDEN, Nicholas & FREITAS, A. A. Hierarchical Classification of G-Protein-Coupled Receptors with a PSO/ACO Algorithm. In: Proc. Of the 2006 IEEE Swarm Intelligence Symposium, pp. 77-84, Jan 2006.

[KEI07] KEITH, R. et al. Molecular Biology of the Cell. Garland Science, 5ª edição, 2007.

[LAR06] LARRAÑAGA, P., et al. Machine Learning in Bioinformatics. Oxford University Press, 2006.

[MEI15] MEIDEIROS, C. Síntese de Proteínas. 13, Set. 2015. Disponível em: http://biologiaacidosnucleicos1e.blogspot.com.br/2015/09/sintese-de-proteinas.html. Acessado em: 06/01/2016.

[MEL04] MELTON, L. Protein Arrays: Proteomics in Multiplex. Nature (429), 6, Mai. 2004, p. 101-107.

[PAW02] PAWLAK, M, et al. Zeptosens' protein microarrays: A novel high performance microarray platform for a low abundance protein analysis. Proteomics, Fev. 2002, p. 383-393.

[RUM86] RUMELHART, D. E., HIMTON, G. E., WILLIANS, R. J. Learning Internal Representations by Error Propagation. Explorations in the Microstructure of Cognition, Vol. 1, Fundations MIT Press, Set. 1985.

[SCU07] SCUSE, D. & REUTEMANN, P. WEKA Experimenter Tutorial for Version 3-5-5. The University of Waikato. 26, Jan. 2007.

[SIL11] SILLA, C. N. & FREITAS, A. A Survey of Hierarchical Classification Across Different Application Domains. Data Mining Knowledge Discovery, vol. 22, 7, Jan. 2011, p. 31-72.

[SOU03] SOUTO M. C. P., LORENA A. C., DELBEM A. C. B., CARVALHO A. C. P. L. F. Técnicas de Aprendizado de Máquina para Problemas de Biologia Molecular. p.103-152. Editora SBC.

[SUN01] Sun, A. & Lim, E. Hierarchical Text Classification and Evaluation. ICDM '01 Proceedings of the 2001 IEEE International Conference on Data Mining, 29, Nov. – 02, Dec. 2001, p 521-528.

[TGO08] The Gene Ontology Consortium. The Gene Ontology project in 2008. Nucleic Acids Research, Jan. 2008. Disponível em:

https://academic.oup.com/nar/article/36/suppl_1/D440/2507489/The-Gene-Ontology-project-in-2008. Acessado em 23/11/2016.

[TGO15] Gene Ontology. The Gene Ontology. 2015. Disponível em: http://www.geneontology.org/page/documentation. Acessado em 20/11/2016.

[TSO07] TSOUMAKAS, G. & KATAKIS, I. Multi-label classification: an overview. International Journal of Data Warehousing & Mining. Idea Group Publishing, 2007, p. 1-13.

[WAN99] Wang, K., et al. Building Hierarchical Classifiers Using Class Proximity. VLDB '99 Proceedings of the 25th International Conference on Very Large Data Bases, 07-10, Set. 1999, p. 363-374.

[WEA05] WEAVER, R.F. Molecular Biology. McGraw-Hill, New York, NY, 2005, p. 432-448.

[WOL92] WOLPERT, D. H. Stacked Generalization. Newral Networks, Complex Systems Group, Theoretical Division, Los Alamos, 1992, p 241-259.