

UNIVERSIDADE FEDERAL FLUMINENSE

BRUNO CORDEIRO PAES

**NOVAS ESTRATÉGIAS PARA CLASSIFICAÇÃO
HIERÁRQUICA LOCAL POR NÍVEL**

NITERÓI

2012

UNIVERSIDADE FEDERAL FLUMINENSE

BRUNO CORDEIRO PAES

**NOVAS ESTRATÉGIAS PARA CLASSIFICAÇÃO
HIERÁRQUICA LOCAL POR NÍVEL**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Otimização Combinatória e Inteligência Artificial

Orientador:
ALEXANDRE PLASTINO

NITERÓI

2012

BRUNO CORDEIRO PAES

NOVAS ESTRATÉGIAS PARA CLASSIFICAÇÃO
HIERÁRQUICA LOCAL POR NÍVEL

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Otimização Combinatória e Inteligência Artificial

Aprovada em 08 de Outubro de 2012.

BANCA EXAMINADORA

Prof. ALEXANDRE PLASTINO - Orientador, UFF

Prof. BIANCA ZADROZNY, UFF

Prof. GISELE L. PAPPA, UFMG

Niterói

2012

Àqueles que acreditam que a ousadia e o erro são caminhos para as grandes realizações.

Agradecimentos

Primeiramente a Deus, pela saúde, fé e perseverança que tem me dado.

A meus pais, Edson e Edicea, e minha irmã, Priscilla, pelo permanente apoio e incentivo à busca de novos conhecimentos.

A meu orientador professor, Alexandre Plastino, pelo seu incentivo, cobrança e presença no auxílio em minhas atividades. Muito obrigado pela oportunidade e pelos ensinamentos que levarei por toda minha vida.

A todos os professores pelo carinho, dedicação e entusiasmo demonstrado ao longo da minha pós-graduação.

Ao CNPq, pelo apoio financeiro.

A minha namorada, Amanda, pela paciência e carinho nos momentos difíceis e felizes.

Aos companheiros que seguiram juntos nesta caminhada e àqueles que traçaram outros caminhos, mas que sempre serão lembrados.

Enfim, a todos que sempre acreditaram no meu sucesso e me deram força para lutar pelo meu objetivo. Obrigado por tudo.

Resumo

No contexto de diversos problemas de classificação, as classes estão organizadas em estruturas hierárquicas, as quais representam relações de descendência entre as classes. Esses problemas são conhecidos como problemas de classificação hierárquica. Métodos baseados em diferentes abordagens têm sido utilizados para resolvê-los, tentando atingir melhor desempenho preditivo.

Nesta dissertação, foram propostos dois classificadores hierárquicos locais por nível utilizando-se diferentes estratégias para eliminar previsões inconsistentes, muito comuns na abordagem local por nível. As estratégias propostas foram comparadas com estratégias tradicionais de diferentes abordagens. Os experimentos computacionais foram realizados com 18 bases de dados hierárquicos e foi possível demonstrar que as ideias propostas são capazes de atingir resultados competitivos e robustos em termos de acurácia preditiva.

Explorou-se também a utilização de diferentes técnicas de seleção de atributos com o objetivo de melhorar o desempenho preditivo dos classificadores hierárquicos explorados. Os experimentos computacionais demonstraram que os classificadores hierárquicos propostos atingem melhor desempenho quando considerados os atributos mais relevantes na construção local dos classificadores, para cada nível da hierarquia.

Palavras-chave: Classificação Hierárquica, Seleção de Atributos, Mineração de Dados

Abstract

In the domain of many relevant classification problems, classes are organized in hierarchies, representing specialization relationships between them. These are the so-called hierarchical classification problems. Methods based on different approaches have been used to solve them, trying to achieve better predictive performance.

In this work, we propose two local per level hierarchical classifiers, which contain distinct strategies to solve inconsistent predictions, common to the local per level approach. We have compared the proposed methods with traditional strategies from different paradigms. The computational experiments, conducted over 18 hierarchical classification data sets, showed that the proposed ideas were able to reach competitive and robust results in terms of prediction accuracy.

We also propose the use of different attribute selection methods in order to improve the predictive performance of the hierarchical classifiers implemented. The computational experiments showed that the proposed hierarchical classifiers achieve better performance when considering the most relevant attributes in the local construction of classifiers for each level of the hierarchy.

Keywords: Hierarchical Classification, Attribute Selection, Data Mining

Lista de Figuras

2.1	Tipos de estruturas hierárquicas	5
2.2	Confiabilidade x Usabilidade	6
2.3	Abordagem de Classificação Plana	7
2.4	Abordagem de Classificação Local por Nó	8
2.5	Ilustração das estratégias para definição dos exemplos positivos e negativos	10
2.6	<i>Binarized Structured Label Learning</i>	11
2.7	Abordagem de Classificação Local por Nó Pai	12
2.8	Abordagem de Classificação Local por Nível	13
2.9	Abordagem de Classificador Global	14
2.10	Exemplo de hierarquia de classes	15
3.1	Funcionamento do classificador FL	21
3.2	Funcionamento do classificador PPTD	22
3.3	Classificadores planos C_1 , C_2 , C_3 e C_4 para cada nível da hierarquia	23
3.4	Exemplo de uma classificação inconsistente	23
3.5	Ilustração do Cenário 1 para a estratégia SV	24
3.6	Ilustração do Cenário 2 para a estratégia SV	25
3.7	Ilustração do Cenário 3 para a estratégia SV	25
3.8	Ilustração do Cenário 1 para a estratégia SW	26
3.9	Ilustração do Cenário 2 para a estratégia SW	27
3.10	Ilustração do Cenário 3 para a estratégia SW	28
3.11	Motivação para o Cenário 4 para a estratégia SW	28
3.12	Ilustração do Cenário 4 para a estratégia SW	29

4.1	Seleção de atributos no classificador FL	40
4.2	Seleção de atributos no classificador PPTD	40
4.3	Seleção de atributos no classificador com a estratégia SW	40
4.4	Seleção de atributos <i>lazy</i> no classificador FL	49
4.5	Seleção de atributos <i>lazy</i> no classificador PPTD	49
4.6	Seleção de atributos <i>lazy</i> no classificador com a estratégia SW	50

Lista de Tabelas

3.1	Características das bases de dados	30
3.2	Comparação entre os classificadores SV e SW	32
3.3	Comparação dos classificadores SW, FL e PPTD - Bases do Grupo A . . .	34
3.4	Comparação dos classificadores SW, FL e PPTD - Bases do Grupo B . . .	35
3.5	Melhores resultados encontrados por estratégia hierárquica	36
3.6	Melhores resultados encontrados por base de dados	36
4.1	hF do Classificador FL com e sem seleção de atributos	43
4.2	hF do Classificador PPTD com e sem seleção de atributos	43
4.3	hF da Estratégia SW com e sem seleção de atributos	44
4.4	Influência da seleção de atributos nos classificadores FL, PPTD e SW . . .	45
4.5	Melhores resultados encontrados por base de dados	46
4.6	Base de dados exemplo	47
4.7	hF do Classificador FL com seleção de atributos <i>Eager</i> e <i>Lazy</i>	51
4.8	hF do Classificador PPTD com seleção de atributos <i>Eager</i> e <i>Lazy</i>	52
4.9	hF da Estratégia SW com seleção de atributos <i>Eager</i> e <i>Lazy</i>	52
4.10	Influência da seleção de atributos <i>Eager</i> e <i>Lazy</i> nos classificadores hierárquicos	53
4.11	Melhores resultados dos classificadores hierárquicos FL, PPTD e SW . . .	55
4.12	Melhores resultados encontrados por base de dados	56

Sumário

1	Introdução	1
2	Classificação Hierárquica	4
2.1	Conceitos	4
2.2	Abordagens para Resolução de Problemas Hierárquicos	6
2.2.1	Abordagem de Classificação Plana	7
2.2.2	Abordagens por Classificadores Locais	7
2.2.2.1	Abordagem de Classificação Local por Nó	8
2.2.2.2	Abordagem de Classificação Local por Nó Pai	12
2.2.2.3	Abordagem de Classificação Local por Nível	13
2.2.3	Abordagem de Classificação Global	14
2.3	Medidas de Avaliação	15
2.3.1	Custo Uniforme	15
2.3.2	Baseada em Distância	16
2.3.3	Baseada na Hierarquia	17
3	Novas Estratégias Hierárquicas	20
3.1	Classificadores Hierárquicos Tradicionais	20
3.1.1	Classificador Hierárquico <i>Flat on Leaves</i>	20
3.1.2	Classificador Hierárquico <i>Per Parent Top-Down</i>	21
3.2	Estratégias propostas	22
3.2.1	Classificador Hierárquico <i>Sum of Votes</i>	23

3.2.2	Classificador Hierárquico <i>Sum of Weighted Votes</i>	25
3.3	Descrição dos Experimentos	29
3.4	Resultados Computacionais	31
3.4.1	Comparação entre as Estratégias Propostas	31
3.4.2	Comparação da Estratégia Proposta SW com as Estratégias Tradi- cionais	33
4	Seleção de Atributos Aplicada à Classificação Hierárquica	37
4.1	Introdução à Seleção de Atributos	37
4.2	Aplicando Seleção de Atributos Tradicional	39
4.2.1	Descrição dos Experimentos	41
4.2.2	Resultados Computacionais	41
4.3	Aplicando Seleção de Atributos <i>Lazy</i>	46
4.3.1	Introdução à Seleção de Atributos <i>Lazy</i>	46
4.3.2	Aplicando Seleção de Atributos <i>Lazy</i>	48
4.3.3	Resultados Computacionais	49
5	Conclusões	57
	Referências	59

Capítulo 1

Introdução

Mineração de dados refere-se à extração de conhecimento novo e útil, muitas vezes na forma de regras e padrões, a partir de bases de dados. Dentre as diversas tarefas de mineração de dados, destaca-se, pela sua aplicabilidade e capacidade preditiva, a tarefa de classificação [16]. Esta consiste, basicamente, em atribuir uma classe, dentre um conjunto pré-definido de classes, a uma determinada instância a partir de seus atributos. Implementar um classificador significa gerar ou descobrir a função que realiza tal mapeamento, dada uma base de treinamento com instâncias do problema.

Em grande parte dos problemas de classificação, as classes conhecidas não possuem relação de descendência entre si. Estes são conhecidos como problemas de classificação plana (*Flat Classification Problems*). No entanto, existem diversos problemas nos quais as classes conhecidas estão organizadas em uma estrutura hierárquica, que define relações de descendência entre as classes. Estes são conhecidos como problemas de classificação hierárquica (*Hierarchical Classification Problems*). Métodos de classificação hierárquica devem ser capazes de considerar a organização hierárquica de classes com o objetivo de se obter uma maior capacidade preditiva.

Exemplos de problemas que possuem suas classes organizadas hierarquicamente podem ser encontrados em diferentes áreas de aplicação. Na área da bioinformática, existem importantes trabalhos que visam a classificação de proteínas e enzimas em classes funcionais, as quais encontram-se organizadas hierarquicamente. As funções de proteínas e enzimas possuem papel essencial em reações químicas intracelulares e, dessa forma, guardam informações que contribuem em descobertas de medicamentos [7] [8] [19] [17] [18] [29]. Na área de classificação de documentos, textos podem ser caracterizados considerando-se uma estrutura hierárquica de assuntos [9] [10] [32]. Em aplicações de reconhecimento de imagens, objetos podem ser categorizados em formas geométricas que possuem relações

de descendência [2].

Os algoritmos de classificação hierárquica se organizam em diferentes categorias [30]. Cada uma das categorias se diferencia em relação à forma em que a estrutura hierárquica é explorada, seja na simplificação da hierarquia (abordagem de classificação plana), na utilização de um conjunto de classificadores planos tradicionais (abordagem por classificadores locais) ou na construção de um único classificador que considera toda a hierarquia de classes (abordagem de classificação global)

As abordagens por classificadores locais (*Local Classifier Approaches*) são as mais comumente exploradas e consideram a hierarquia de classes através de uma perspectiva local, combinando classificadores que consideram, isoladamente, diferentes partes da hierarquia. Em [30], os classificadores locais são categorizados de acordo com a forma com que essas informações locais são exploradas: abordagem de classificação local por nó (*local classifier per node*), abordagem de classificação local por nó pai (*local classifier per parent node*) e abordagem de classificação local por nível (*local classifier per level*).

Na abordagem de classificação local por nível, classificadores planos são treinados, de forma independente, um para cada nível da hierarquia. O principal problema desse tipo de classificação é a ocorrência de inconsistências nos resultados obtidos entre os diferentes classificadores independentes, associados aos diferentes níveis da hierarquia. Nesta dissertação, são propostos dois algoritmos locais por nível com tratamentos distintos de classificações inconsistentes. As estratégias propostas são comparadas com outros algoritmos de classificação hierárquica pertencentes a diferentes abordagens, utilizando-se oito bases de dados com informações de funções de proteínas e dez bases de dados com informações de funções gênicas.

Existem diversas medidas desenvolvidas para avaliar a capacidade de predição de classificadores. No entanto, grande parte dessas medidas foi projetada para avaliar classificadores planos, não considerando as relações de descendência entre as classes. Com o surgimento de problemas hierárquicos e, por conseguinte, o desenvolvimento de classificadores hierárquicos, foi necessário projetar medidas que considerassem essas relações entre as classes. Neste trabalho, utilizou-se a medida *hierarchical F-measure* (hF). Trata-se de uma adaptação, apresentada em [21], da medida *F-measure* para o contexto hierárquico. A avaliação dos classificadores foi efetuada utilizando-se validação cruzada com 10 partições e a medida hierárquica hF. Para avaliar a significância estatística na comparação entre dois classificadores hierárquicos, utilizou-se o teste-t de Student [20].

Os experimentos computacionais realizados com as 18 bases de classificação hierár-

quica evidenciaram que os algoritmos propostos são capazes de atingir resultados competitivos e robustos em termos de acurácia preditiva.

Outra importante contribuição deste trabalho está centrada na utilização de diferentes técnicas de seleção de atributos com o objetivo de melhorar o desempenho dos classificadores hierárquicos explorados. Tentativas semelhantes foram realizadas em [22] e [28].

Seleção de atributos é uma técnica muito explorada na área de mineração de dados, principalmente na tarefa de classificação [14]. Nesse contexto, seu objetivo é identificar atributos relevantes, visando obter um ou mais dentre os seguintes benefícios: redução do tempo de execução do processo de classificação, aumento da capacidade preditiva do classificador e obtenção de uma representação mais compacta do conceito a ser aprendido.

Neste trabalho, foi explorada uma estratégia para melhorar a acurácia da classificação hierárquica usando métodos de seleção de atributos na construção local dos classificadores, para cada nível da hierarquia. Dois métodos foram utilizados: um método de seleção de atributos tradicional, denominado *eager*, onde os atributos são selecionados em uma fase de pré-processamento, e um novo método de seleção de atributos, proposto em [26], denominado *lazy*, que adia a identificação dos atributos relevantes para o momento da classificação de uma instância.

Nos experimentos computacionais realizados nas mesmas 18 bases de dados, verificou-se que os classificadores hierárquicos propostos atingem melhor desempenho considerando somente os atributos mais relevantes em cada nível da hierarquia nos classificadores locais na abordagem de classificação local por nível. Na maioria dos casos, a utilização dos atributos selecionados levou ao melhor resultado, tendo o método de seleção *lazy* apresentado um desempenho superior em relação à técnica de seleção de atributos *eager*.

Para apresentar e discutir o trabalho desenvolvido, esta dissertação foi organizada da seguinte forma. No Capítulo 2, apresentam-se os conceitos básicos de classificação hierárquica, as diferentes abordagens de estratégias utilizadas nesse contexto, assim como medidas de avaliação aplicadas à classificação hierárquica. No Capítulo 3, apresenta-se a principal contribuição deste trabalho: dois novos algoritmos de classificação hierárquica local por nível, assim como os experimentos computacionais realizados para avaliá-los. No Capítulo 4, explora-se a aplicação de métodos de seleção de atributos em conjunto com os classificadores hierárquicos utilizados neste trabalho, com o objetivo de melhorar as respectivas acurácias preditivas. No Capítulo 5, são destacadas as conclusões do trabalho e apontados possíveis trabalhos futuros.

Capítulo 2

Classificação Hierárquica

Neste capítulo, apresentam-se os conceitos básicos de classificação hierárquica, as abordagens utilizadas para resolução de problemas hierárquicos, assim como medidas de avaliação aplicadas à classificação hierárquica.

A Seção 2.1 apresenta os conceitos e características dos problemas de classificação hierárquica e, na Seção 2.2, são descritas algumas das principais abordagens para construção de classificadores hierárquicos. Na Seção 2.3, são apresentadas medidas utilizadas para avaliar a capacidade preditiva desses classificadores.

2.1 Conceitos

Classificação é uma tarefa muito explorada na área de mineração de dados. Seu objetivo é identificar as classes ainda não conhecidas de novas instâncias, baseando-se nas características de outras instâncias da base de dados com classe já conhecida. Em grande parte desses problemas, as classes do domínio de aplicação não apresentam relações de descendência entre si. Esses problemas são conhecidos como problemas de classificação plana (*Flat Classification Problem*). No entanto, existem diversos problemas nos quais as classes conhecidas estão organizadas em uma estrutura hierárquica, definida a partir de relações de descendência entre as classes. Esses problemas são conhecidos como problemas de classificação hierárquica (*Hierarchical Classification Problem*) [30].

Em problemas hierárquicos, a organização das classes pode ser definida em dois tipos de estruturas: estrutura de grafo acíclico ou árvore, representada por uma hierarquia de classes na qual cada classe possui somente uma classe pai (Figura 2.1(a)), e estrutura de grafo cíclico, representada por uma hierarquia de classes na qual uma classe filha pode

ser associada a diversas classes pai (Figura 2.1(b)).

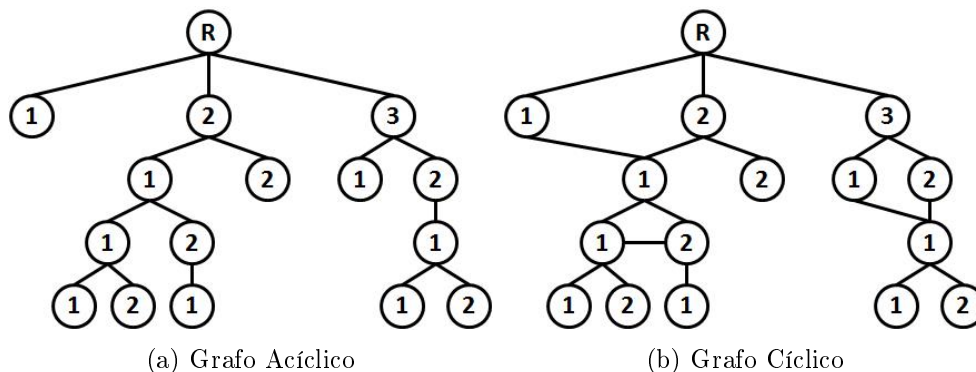


Figura 2.1: Tipos de estruturas hierárquicas

Problemas de classificação hierárquica podem ser categorizados de acordo com a obrigatoriedade (ou não) de cada instância estar associada a uma classe folha [13]:

- Há casos em que todas as instâncias são obrigatoriamente associadas a classes que são nós folhas da hierarquia de classes (*mandatory leaf-node prediction*). O método de classificação hierárquica implementado deve sempre classificar as instâncias em nós folhas.
- Em outros casos, a classe mais específica de uma instância pode ser um nó interno da hierarquia de classes (*optional leaf-node prediction*). O método de classificação hierárquica decide até que nível da hierarquia será realizada a previsão de classes.

Os problemas hierárquicos também podem ser agrupados de acordo com o número de classes a que cada instância pode ser associada: problemas cujas instâncias só podem ser classificadas em uma classe por nível da hierarquia (*Single Label Problems*) e problemas em que, para cada instância, é previsto um conjunto de classes por nível da hierarquia (*Multi Label Problems*) [30].

Uma outra importante característica da classificação hierárquica é a relação entre confiabilidade e usabilidade nas previsões dos métodos hierárquicos [13]. A Figura 2.2 ilustra um exemplo de uma hierarquia de classes. À medida que a previsão é dada em um nível mais perto da raiz da hierarquia, mais confiável será essa previsão. Baseando-se no fato de que, em níveis mais próximos à raiz, normalmente, as instâncias são distribuídas em um número menor de classes, o processo de aprendizado e classificação nesses níveis é facilitado. Por outro lado, à medida que a previsão é dada em níveis mais específicos da hierarquia (mais distantes da raiz), maior utilidade terá essa previsão. Baseando-se

no fato de que, nos níveis mais distantes da raiz, a informação associada a cada classe é mais específica e inclui as informações das classes dos níveis próximos à raiz.

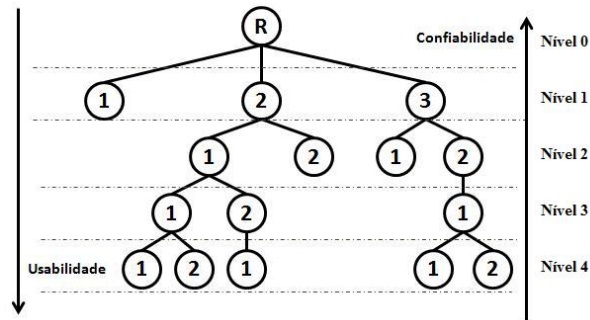


Figura 2.2: Confiabilidade x Usabilidade

Observa-se que a principal motivação da área de classificação hierárquica é a criação de métodos que possuam a capacidade de extrair informações da organização hierárquica de classes, com o objetivo de se obter uma maior capacidade preditiva.

O escopo deste trabalho está definido no tratamento de problemas hierárquicos organizados através de grafos acíclicos, nos quais cada instância está associada a uma única classe por nível da hierarquia (*Single Label Problem*) e nos quais a previsão é realizada sempre até o nível mais específico da hierarquia de classes (*mandatory leaf-node prediction*).

2.2 Abordagens para Resolução de Problemas Hierárquicos

Nesta seção, são apresentadas as abordagens para resolução de problemas de classificação hierárquica de acordo com a categorização descrita em [30]. Cada uma das abordagens se diferencia na forma em que a estrutura hierárquica é explorada, seja na simplificação da hierarquia (abordagem de classificação plana), na utilização de um conjunto de classificadores planos tradicionais (abordagem por classificadores locais) ou na construção de um único classificador que considera toda a hierarquia de classes (abordagem de classificação global). Nas próximas subseções, descreve-se o funcionamento da cada abordagem e realiza-se uma revisão dos trabalhos relacionados.

2.2.1 Abordagem de Classificação Plana

A abordagem de classificação plana (*Flat Classification Approach*) simplifica o problema de classificação hierárquica, transformando-o em um problema de classificação tradicional que ignora completamente a hierarquia de classes. Geralmente, somente as classes folhas da hierarquia são consideradas no treinamento e no processo de classificação (Figura 2.3), porém, em alguns casos, baseando-se em um conhecimento prévio, selecionam-se classes de outros níveis.

No processo de classificação das instâncias, a abordagem prevê indiretamente a solução para o problema de classificação hierárquica, pois, a partir da previsão de uma classe folha, é possível inferir todas as classes ancestrais percorrendo-se o caminho da classe folha em questão até a raiz da hierarquia.

A abordagem de classificação plana também é denominada na literatura como uma abordagem direta (*Direct Approach*) [3] e considerada também como um "classificador global" em [36]. Em [1], o autor a considera uma abordagem *bottom-up*, onde a previsão é dada inicialmente nas classes folhas e, a partir dessas, inferem-se as classes internas até o nível mais alto da hierarquia.

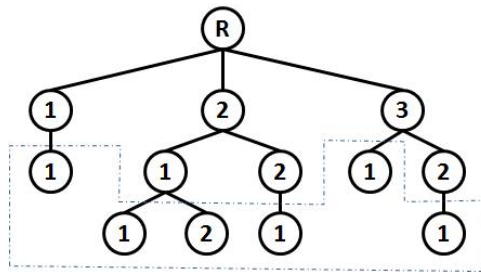


Figura 2.3: Abordagem de Classificação Plana

A vantagem dessa abordagem é a simplicidade de aplicação, pois pode ser utilizado um único classificador plano tradicional na previsão de um conjunto de classes da hierarquia. No entanto, tem-se a desvantagem de não explorar as informações provenientes das relações de descendência entre as classes presentes na hierarquia.

2.2.2 Abordagens por Classificadores Locais

As abordagens por classificadores locais (*Local Classifier Approaches*) exploram a hierarquia de classes através de uma perspectiva local, combinando classificadores que consideram, isoladamente, diferentes partes da hierarquia. Em [30], os classificadores locais são

categorizados de acordo com a forma com que essas informações locais são exploradas: abordagem de classificação local por nó (*local classifier per node*), abordagem de classificação local por nó pai (*local classifier per parent node*) e abordagem de classificação local por nível (*local classifier per level*).

Em [22], foi apresentado o primeiro classificador hierárquico baseado em informação local, com o treinamento de um classificador plano tradicional para cada nó pai. A classificação da nova instância é realizada de maneira *top-down*, ou seja, para cada nova instância, a previsão é feita do nível mais alto da hierarquia até o nível mais baixo, considerando os resultados encontrados em cada nível.

Na literatura, é comum encontrar versões de abordagens locais denominando-se *top-down*, devido ao uso dessa estratégia na fase de classificação. Porém, é importante ressaltar que a abordagem é definida de acordo com o uso das informações locais na fase de treinamento, sendo o método *top-down* uma estratégia para se obter previsões consistentes na fase de classificação.

As próximas subseções apresentam as descrições dos três tipos de abordagens por classificadores locais.

2.2.2.1 Abordagem de Classificação Local por Nó

A abordagem de classificação local por nó consiste no treinamento de um classificador binário associado a cada nó da hierarquia de classes (exceto o nó raiz). Após o treinamento, uma hierarquia de classificadores planos é formada.

A Figura 2.4 apresenta um exemplo de hierarquia de classes, com a representação dos classificadores binários através de pontilhados em cada nó. Basicamente, cada classificador binário prevê se uma instância pertence ou não à classe à qual está associado.

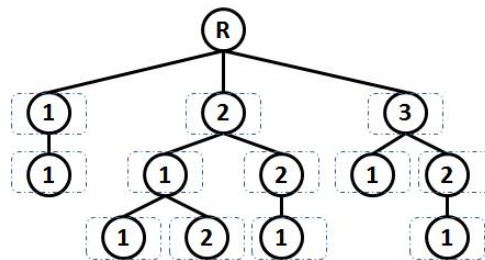


Figura 2.4: Abordagem de Classificação Local por Nó

Para a realização do treinamento do classificador binário associado a cada nó, definem-se inicialmente os conjuntos das instâncias positivas e negativas para o treinamento dos

classificadores binários. Existem diferentes formas para definir esses subconjuntos, apresentados nos trabalhos [4, 11, 12].

Para exemplificação, a Figura 2.5 ilustra a aplicação de diferentes formas de agrupamento das instâncias para o treinamento do classificador binário associado à classe 2.1. As instâncias associadas aos nós com um sombreado mais claro são consideradas positivas e as associadas aos nós com um sombreado mais escuro são consideradas negativas. Apresenta-se a seguir uma breve descrição dessas formas de agrupamentos.

- *Exclusive* - Somente as instâncias associadas à classe 2.1 serão consideradas positivas. As instâncias associadas às demais classes serão consideradas negativas.
- *Less Exclusive* - Somente as instâncias associadas à classe 2.1 serão consideradas positivas. As instâncias associadas às demais classes, excluindo as classes filhas da classe 2.1 (classes 2.1.1 e 2.1.2) serão consideradas negativas.
- *Less Inclusive* - As instâncias associadas à classe 2.1 e às suas classes filhas 2.1.1 e 2.1.2 serão consideradas positivas. As instâncias associadas às demais classes serão consideradas negativas.
- *Inclusive* - As instâncias associadas à classe 2.1 e às suas classes filhas 2.1.1 e 2.1.2 serão consideradas positivas. As instâncias associadas às demais classes, excluindo as classes ancestrais da classe 2.1 (a classe 2) serão consideradas negativas.
- *Siblings* - As instâncias associadas a classe 2.1 e às suas classes filhas 2.1.1 e 2.1.2 serão consideradas positivas. Somente as instâncias associadas às classes "primas" da classe 2.1 (no caso, apenas a classe 2.2) serão consideradas negativas.
- *Exclusive Siblings* - Somente as instâncias associadas à classe 2.1 serão consideradas positivas. Somente as instâncias associadas às classes "primas" da classe 2.1 serão consideradas negativas.

A abordagem de classificação local por nó possui a vantagem de ser modularizada, ou seja, cada classificador pode ser treinado utilizando diferentes algoritmos de classificação. No entanto, dependendo do tamanho da hierarquia, um grande número de classificadores deve ser treinado, impactando o desempenho. Outro problema é relacionado às múltiplas classificações obtidas nos classificadores binários, permitindo que uma instância seja associada a diferentes classes de um mesmo nível (inconsistência horizontal) ou de diferentes

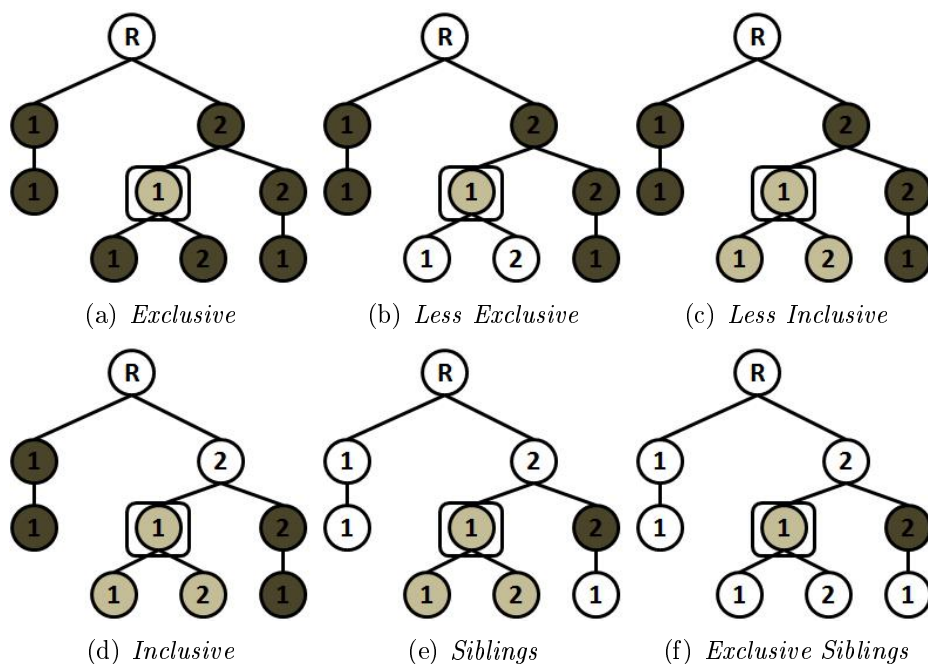


Figura 2.5: Ilustração das estratégias para definição dos exemplos positivos e negativos

ramos da hierarquia (inconsistência vertical). Por exemplo, na Figura 2.4, após a realização da classificação de uma instância teste, os resultados gerados pelos classificadores binários podem ser positivos para as classes 2, 2.1, 2.2 e 3.2.1. Dessa forma, no nível 2, têm-se as classes previstas 2.1 e 2.2 (inconsistência horizontal) e, além disso, tem-se a classe 3.2.1 do nível 3 que não é consistente com a classificação do nível 2 (inconsistência vertical).

Portanto, as classificações geradas pelos classificadores locais por nó podem ser inconsistentes e, assim, é preciso incluir métodos de tratamento ou correção das inconsistências verticais (em problemas multirrótulo e monorrótulo) e horizontais (em problemas monorrótulo). A seguir, apresenta-se uma revisão desses métodos para a abordagem local por nó.

O método proposto em [35], denominado *Binarized Structured Label Learning*, aplicado a um problema multirrótulo, consiste no treinamento de classificadores binários, um para cada nó da hierarquia. A instância é classificada percorrendo cada ramo de classificadores da hierarquia. A ideia, basicamente, é interromper o processo de classificação quando o classificador binário associado a uma determinada classe da hierarquia obtiver uma previsão negativa para uma instância que está sendo classificada. Por exemplo, utilizando a hierarquia exemplo da Figura 2.6, na qual os nós com um sombreado mais claro são considerados classificações positivas para a instância, os nós com um sombreado mais escuro são considerados classificações negativas e nos nós em branco não ocorreu classi-

ficação para a instância. Percorrendo o primeiro ramo, tem-se a saída do classificador associado à classe 1 positiva e a saída do classificador associado a classe 1.1 negativa. O método, então, não obtém as classificações para os classificadores associados às classes descendentes 1.1.1 e 1.1.2 (Figura 2.6 (a)). Repetindo esse procedimento para os demais ramos da hierarquia, constroem-se caminhos compostos por classes que obtiveram classificações positivas, representados na Figura 2.6 (b). No final desse processo, será obtida uma hierarquia estruturada que indicará as classes previstas para a instância analisada. Dessa forma, o método *Binarized Structured Label Learning*, aplicado a um problema multirrótulo, evita que ocorram inconsistências verticais.

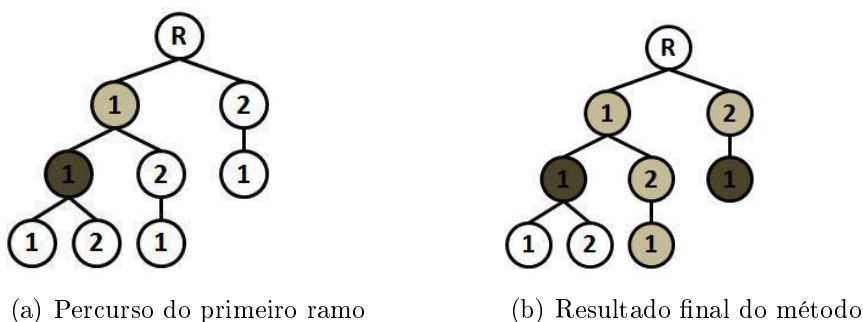


Figura 2.6: *Binarized Structured Label Learning*

Em [10], são propostos dois métodos que evitam inconsistências verticais em problemas multirrótulo, considerando-se as relações de descendência e valores limites. O primeiro método usa uma condição em que todas as probabilidades das classes previstas em um caminho têm que ser maiores que um valor limite. O segundo método usa um limite multiplicativo para ser comparado com o produto das probabilidades das classes previstas de dois níveis consecutivos. Por exemplo, na Figura 2.4, se $p(c_2) = 0,7$, $p(c_{2.1}) = 0,6$, $p(c_{2.1.1}) = 0,6$ e o limite é $0,6$, esse caminho é escolhido no primeiro método, pois todas as probabilidades são maiores que $0,6$. No segundo método, associa-se à instância somente a classe 2, pois no segundo nível a multiplicação das probabilidades associadas às classes 2 e 2.1 resulta em $0,42$ ($0,7 \times 0,6$), estando abaixo do valor limite. Esses métodos foram aplicados em um problema de classificação hierárquica multirrótulo cujos elementos envolvidos eram páginas *web*.

Em [33], foi proposto um método, aplicado a um problema multirrótulo, que consiste na correção dos resultados dos classificadores binários locais associados a cada nó, combinando, através de uma função, as probabilidades das previsões positivas, de forma *bottom-up*. Por exemplo, na Figura 2.4, os classificadores associados às classes folhas 2.1.1 e 2.1.2 obtêm as classificações positivas pois suas probabilidades estimadas foram maiores que um valor limite t . Subindo para o próximo nível, a probabilidade do classificador

associado à classe 2.1 é estimada utilizando a função com os valores das probabilidades originadas das suas classes filhas com classificações positivas. Caso o valor da probabilidade associado ao classificador da classe 2.1 seja maior que o valor limite t , atribui-se a classificação positiva a essa classe. Em caso contrário, atribui-se a classificação negativa para a classe 2.1 e transformam-se todas as classificações de suas classes filhas em negativas. Dessa forma, uma hierarquia de classificações consistentes é construída, pois as inconsistências verticais são corrigidas.

Em [2], mais um método, aplicado a um problema multirrótulo, tenta corrigir classificações inconsistentes. Neste, as inconsistências verticais são eliminadas utilizando-se uma rede Bayesiana extraída considerando-se a hierarquia de classes e a base de treinamento.

2.2.2.2 Abordagem de Classificação Local por Nó Pai

A abordagem de classificação local por nó pai consiste no treinamento de um classificador plano tradicional associado a cada classe pai da hierarquia de classes. Dessa forma, para cada classe pai tem-se um classificador plano que considera somente as suas classes filhas.

A Figura 2.7 apresenta um exemplo de hierarquia de classes na qual as classes pais possuem um envoltório pontilhado representando que possuem classificadores associados.

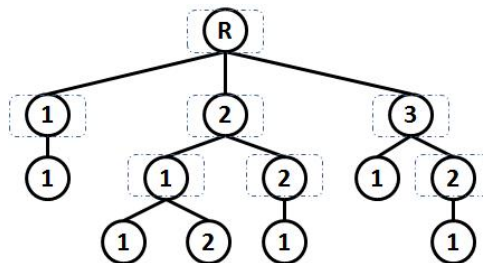


Figura 2.7: Abordagem de Classificação Local por Nó Pai

Essa abordagem também é comumente encontrada na literatura como uma abordagem *top-down*. No entanto, como explicado anteriormente, o método *top-down* é aplicado na fase de classificação para evitar classificações inconsistentes.

Aplicações dessa abordagem podem ser encontradas nos seguintes trabalhos. Em [29], foi proposto um classificador, denominado *Selective Classifier*, baseado na abordagem local por nó pai com a utilização do método *top-down* na fase de classificação. Basicamente, ela está baseada na hipótese de que, utilizando-se diferentes algoritmos de classificação em cada nó pai da hierarquia, pode-se melhorar a acurácia preditiva do classificador hierárquico. Para cada nó pai da hierarquia, diferentes classificadores são treinados com

subconjuntos distintos da base de treinamento e, então, são avaliados com uma base de teste. Assim, o classificador que obtém a maior acurácia é associado àquele nó pai. Em [19], é apresentada uma melhoria ao *selective classifier* proposto, na qual a escolha dos classificadores associados aos nós pai é realizada de forma global, de maneira que se encontre a melhor combinação de classificadores para os nós pai, que resulte em uma maior acurácia. Adicionalmente, em [28], foi proposta a inclusão da seleção de atributos para cada classificador associado a cada nó pai.

2.2.2.3 Abordagem de Classificação Local por Nível

A abordagem de classificação local por nível consiste no treinamento de um classificador plano para cada nível da hierarquia de classes. A Figura 2.8 ilustra essa abordagem. Para cada nível da hierarquia, existe um classificador responsável por selecionar uma das classes pertencente ao nível em questão.

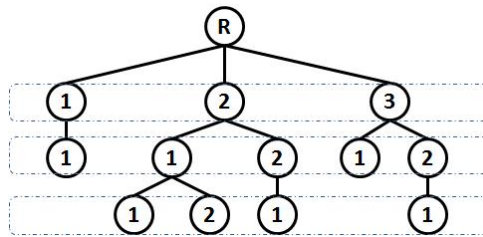


Figura 2.8: Abordagem de Classificação Local por Nível

Em [5, 8], essa abordagem foi utilizada como *baseline* para avaliação de outros classificadores hierárquicos, na comparação dos resultados obtidos nível a nível, sem tratamento de inconsistências.

O principal problema desse tipo de classificação hierárquica é a ocorrência de inconsistências nos resultados obtidos pelos classificadores associados aos diferentes níveis da hierarquia. Por exemplo, considerando a hierarquia da Figura 2.8, é possível que a classe 3 seja prevista no nível 1, a classe 2.1 seja prevista no nível 2 e, finalmente, a classe 3.2.1 seja prevista no nível 3. Dessa forma, o resultado gerado no segundo nível não é consistente com os resultados previstos no primeiro e no terceiro níveis, pois a classe 2.1 não é descendente da classe 3 e não é pai da classe 3.2.1

Conforme destacado anteriormente, para que essa abordagem possa ser utilizada, é necessário um tratamento posterior seja realizado para corrigir possíveis classificações inconsistentes nos diferentes níveis da hierarquia. Uma das contribuições deste trabalho é a proposta de um método de tratamento posterior de inconsistências para a abordagem

local por nível, dado que todos os métodos de tratamento de inconsistências identificados são aplicados à abordagem local por nó e, à abordagem local por nó pai.

2.2.3 Abordagem de Classificação Global

A abordagem de classificação global consiste basicamente na construção de um único modelo que considera toda a hierarquia de classes durante uma única execução do algoritmo de classificação. Ilustrada na Figura 2.9, essa abordagem é encontrada em diversos trabalhos com a denominação de abordagem *big-bang* [8, 13].

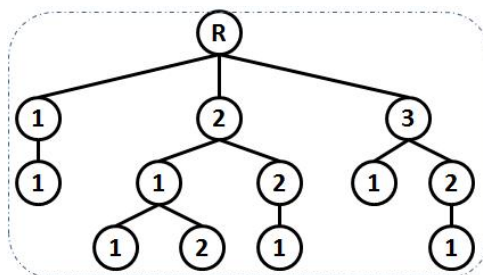


Figura 2.9: Abordagem de Classificador Global

A principal característica dessa abordagem é sua capacidade de considerar todas as classes da hierarquia em um único treinamento, diferente das abordagens locais que dividem o treinamento em módulos, considerando partes da hierarquia de classes. Consequentemente, a abordagem global sofre com a falta de modularização, gerando um modelo com maior complexidade.

Alguns trabalhos que utilizam essa abordagem são descritas a seguir.

O primeiro baseia-se no classificador Rocchio, proposto em [27]. A idéia básica é, através de um cálculo de distância entre a instância a ser classificada e cada classe da hierarquia, associar a instância à classe mais próxima.

Em [23], é apresentado um sistema de classificação de páginas *web* em múltiplas categorias hierárquicas, ou seja, um sistema multirrótulo. Basicamente, durante a fase de classificação, cada página é comparada, através de uma medida de similaridade com cada categoria da hierarquia. Caso o valor de similaridade medido esteja acima de um valor limite, a página *web* é associada à categoria.

Outra estratégia para construção de um classificador global consiste na modificação de classificadores planos tradicionais para que considerem a estrutura hierárquica de classes. Em [5], foi proposta uma modificação no algoritmo de indução de árvores de decisão

C4.5 para considerar a hierarquia de classes (HC4.5). Em [31], modificou-se o tradicional algoritmo *Naive Bayes* para considerar as relações de descendência entre as classes nas probabilidades estimadas.

2.3 Medidas de Avaliação

Existem diversas medidas desenvolvidas para avaliar a capacidade de predição de classificadores. No entanto, grande parte dessas medidas foi projetada para avaliar classificadores planos, não considerando as relações de descendência entre as classes.

Com o surgimento de problemas hierárquicos e, por conseguinte, o desenvolvimento de classificadores hierárquicos, foi necessário projetar medidas que considerassem as relações de descendência entre as classes.

A seguir, serão brevemente descritas três importantes tipos de medidas hierárquicas: custo uniforme, baseada em distância e baseada na hierarquia. Todas estão detalhadamente definidas em [6, 13, 30]. Para exemplificação, será utilizada a hierarquia ilustrada na Figura 2.10.

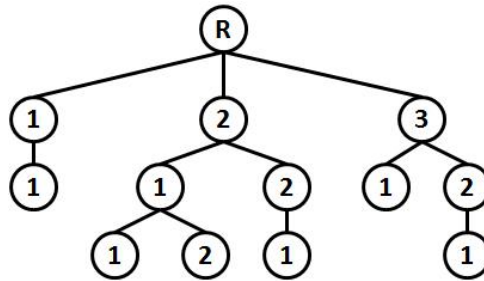


Figura 2.10: Exemplo de hierarquia de classes

2.3.1 Custo Uniforme

A medida de custo uniforme, basicamente, considera que a previsão para uma instância está correta se as previsões de todos os níveis estão corretas (somando custo zero ao total de erros) e que a previsão está incorreta (somando custo 1 ao total de erros) se a previsão em pelo menos um nível está incorreta. Dessa forma, se em pelo menos um dos níveis da hierarquia a classe prevista não corresponde à classe real, a classificação é penalizada com o custo unitário. Caso contrário, a classificação ocorreu corretamente, não sendo penalizada.

Para exemplificar, considere a hierarquia da Figura 2.10. Suponha que a classe real de uma instância seja a classe 2.1.1 e a classe prevista obtida por um classificador seja a classe 2.1.2. Nesse caso, como existe pelo menos um erro, no valor da classe prevista do terceiro nível, essa classificação é penalizada com o custo unitário.

A Fórmula 2.1 define essa medida. Seja P_i o conjunto das classes previstas em todos os níveis da hierarquia para uma instância i e seja T_i o conjunto das classes reais em todos os níveis da hierarquia para a instância i . Seja n o número total de instâncias. O valor da medida de custo uniforme U_i para uma instância é calculado da seguinte forma: se o conjunto P_i for igual ao conjunto T_i , U_i é igual a 0, caso contrário será 1.

O valor U da medida uniforme para um conjunto de n instâncias é dado pela seguinte fórmula.

$$U = \frac{\sum_i^n U_i}{n} \quad (2.1)$$

Essa é a medida mais simples pois ignora completamente as relações de descendência entre as classes. O erro de classificação entre classes filhas do mesmo nó pai tende a ser menos grave que erros entre classes primas, de nós pais distintos. Por exemplo, se a classe real de uma instância é 2.1, a classe irmã prevista 2.2 caracteriza um erro menos grave do que a classe prima prevista 3.1.

2.3.2 Baseada em Distância

A medida baseada em distância consiste em atribuir, para cada erro de classificação, um custo proporcional à distância entre a classe prevista e a classe real de uma instância na hierarquia de classes.

Essas medidas podem ser categorizadas em dois tipos:

- Medidas de custo de classificação independente da profundidade

O custo do erro de classificação entre a classe prevista e classe real é obtida através do número de arestas do menor caminho que as interligam. Exemplificando, observe a Figura 2.10 e suponha que a classe prevista seja 2.1.1 e a classe real seja 2.2.1, o custo proporcional será igual ao valor 4, obtido pelo número de arestas no menor caminho entre as classes real e prevista. Essa medida não considera que erros em níveis mais próximos à raiz são piores que erros em níveis mais profundos.

- Medida de custo de classificação dependente da profundidade

O custo do erro de classificação entre as classes prevista e real é obtido através dos custos das arestas do menor caminho que as interligam. Além disso, os custos são atribuídos de acordo com o nível de profundidade de cada aresta, ou seja, as arestas dos níveis mais distantes da raiz possuem um custo menor. Desse modo, erros em níveis superiores apresentam valores maiores que erros em níveis inferiores. Exemplificando com a Figura 2.10, suponha que as arestas dos níveis 1, 2 e 3 possuam custos associados de 3, 2 e 1, respectivamente. Considerando a classe prevista 2.1.1 e a classe real 2.2.1, o custo do erro atribuído será 6. No caso da classe prevista ser 2.1 e a real 3.1, o custo do erro atribuído será 8. Observa-se que erros nos níveis mais próximos à raiz possuem maiores penalizações.

A Fórmula 2.2 define uma medida baseada em distância. Seja p_i a classe prevista mais específica e t_i a classe real mais específica para uma instância i . Seja $dist(x, y)$ uma função que retorna o custo da distância entre duas classes x e y da hierarquia.

O valor da medida baseada em distância D para um conjunto de n instâncias é dado pela seguinte fórmula.

$$D = \sum_i^n \frac{1}{dist(p_i, t_i)} \quad (2.2)$$

Para a medida ser independente da profundidade, a função $dist(x, y)$ deve contabilizar o número de arestas entre a classe real e a classe prevista. Para a medida ser dependente da profundidade, a função $dist(x, y)$ deve considerar os custos associados às arestas entre a classe prevista e a classe real.

2.3.3 Baseada na Hierarquia

Medidas baseadas na hierarquia consideram as relações de descendência da estrutura hierárquica. Em [21], foi proposto um método para avaliar o desempenho de classificadores hierárquicos, baseando-se nas conhecidas medidas de *precision*, *recall* e *f-measure*. A extensão dessas medidas, considerando agora a estrutura hierárquica, resultou nas medidas chamadas de *hierarchical precision* (hP), *hierarchical recall* (hR) e *hierarchical f-measure* (hF).

A vantagem dessas medidas hierárquicas é sua aplicação em diversos cenários de clas-

sificação hierárquica: em estruturas de árvores, grafos cíclicos, multirrótulo, problemas com classificação em nós internos da hierarquia e outras. Além disso, elas são baseadas em medidas bem conhecidas e testadas na literatura.

Apresenta-se, a seguir, a definição formal dessas medidas. Considere i , P_i , T_i e n como definidos na Subseção 2.3.1.

A medida *hierarchical precision* (hP), definida na Fórmula 2.3, é o resultado da divisão do somatório (para todas as instâncias) do número de classes comuns entre as classes previstas e reais da instância i pelo número de classes previstas para a instância i .

$$hP = \frac{\sum_i |P_i \cap T_i|}{\sum_i |P_i|} \quad (2.3)$$

A medida *hierarchical recall* (hR), definida na Fórmula 2.4, é o resultado da divisão do somatório (para todas as instâncias) do número de classes comuns entre as classes previstas e reais da instância i pelo número de classes reais da instância i .

$$hR = \frac{\sum_i |P_i \cap T_i|}{\sum_i |T_i|} \quad (2.4)$$

A medida *hierarchical f-measure* (hF), definida na Fórmula 2.5, é calculada como a média harmônica das medidas hP e hR .

$$hF = \frac{2 * hP * hR}{hP + hR} \quad (2.5)$$

Para exemplificar, considera-se a hierarquia da Figura 2.10 e a avaliação de duas instâncias teste. A instância teste 1 possui o conjunto P_1 com as classes previstas 2, 2.1 e 2.1.1 e o conjunto T_1 com as classes reais 2, 2.1 e 2.1.2. A instância teste 2 possui o conjunto P_2 com as classes previstas 3 e 3.1 e o conjunto T_2 com as classes reais 3, 3.2 e 3.2.1.

Os valores das medidas *hierarchical precision* (hP), *hierarchical recall* (hR) e *hierarchical f-measure* (hF), considerando-se as instâncias teste 1 e 2, são calculados a seguir.

$$hP = \frac{\sum_i |P_i \cap T_i|}{\sum_i |P_i|} = \frac{|P_1 \cap T_1| + |P_2 \cap T_2|}{|P_1| + |P_2|} = \frac{2 + 1}{3 + 2} = \frac{3}{5} = 0,6 \quad (2.6)$$

$$hR = \frac{\sum_i |P_i \cap T_i|}{\sum_i |T_i|} = \frac{|P_1 \cap T_1| + |P_2 \cap T_2|}{|T_1| + |T_2|} = \frac{2 + 1}{3 + 3} = \frac{3}{6} = 0,5 \quad (2.7)$$

$$hF = \frac{2 * hP * hR}{hP + hR} = \frac{2 * 0,6 * 0,5}{0,6 + 0,5} = \frac{0,6}{1,1} = 0,54 \quad (2.8)$$

Capítulo 3

Novas Estratégias Hierárquicas

Neste capítulo, apresentam-se as propostas dos classificadores hierárquicos baseados em tratamento de inconsistências, assim como as versões de classificadores hierárquicos tradicionais que foram adotados como base de comparação nos experimentos computacionais realizados neste trabalho.

A Seção 3.1 apresenta os classificadores tradicionais utilizados e, na Seção 3.2, são descritos os classificadores hierárquicos propostos. A Seção 3.3 descreve os experimentos computacionais realizados e, finalmente, na Seção 3.4, é feita uma análise comparativa entre os classificadores hierárquicos explorados.

3.1 Classificadores Hierárquicos Tradicionais

Com o objetivo de avaliar o desempenho dos classificadores propostos, foram implementados dois classificadores hierárquicos baseados em abordagens hierárquicas tradicionais descritas no Capítulo 2: classificação plana e classificação local por nó pai.

3.1.1 Classificador Hierárquico *Flat on Leaves*

O classificador hierárquico *Flat on Leaves* (FL) baseia-se nos conceitos da abordagem de classificação plana (*Flat Classification Approach*). Basicamente, realiza-se o treinamento de um único classificador plano, considerando-se apenas as classes folha da hierarquia de classes. Portanto, o resultado obtido por esse classificador plano é uma classe folha e, a partir dela, inferem-se as classes dos níveis superiores (mais próximos à raiz) da hierarquia. Para facilitar o entendimento, neste trabalho definiu-se que, os níveis superiores ou maiores são os níveis mais próximos da raiz, e os níveis inferiores ou menores representam os níveis

mais próximos às folhas.

A Figura 3.1 apresenta um exemplo do funcionamento do classificador FL implementado. As classes sombreadas na hierarquia da Figura 3.1(a) representam as classes folha que fazem parte do treinamento. A Figura 3.1(b) ilustra que, para uma dada instância de entrada, o classificador prevê a classe folha 2.1.1.2 do quarto nível da hierarquia (sombreada). A partir dessa classe, é possível inferir as classes dos níveis superiores (sombreadas). Neste exemplo, tem-se no nível 3 a classe 2.1.1, no nível 2 a classe 2.1 e, finalmente, no nível 1 a classe 2.

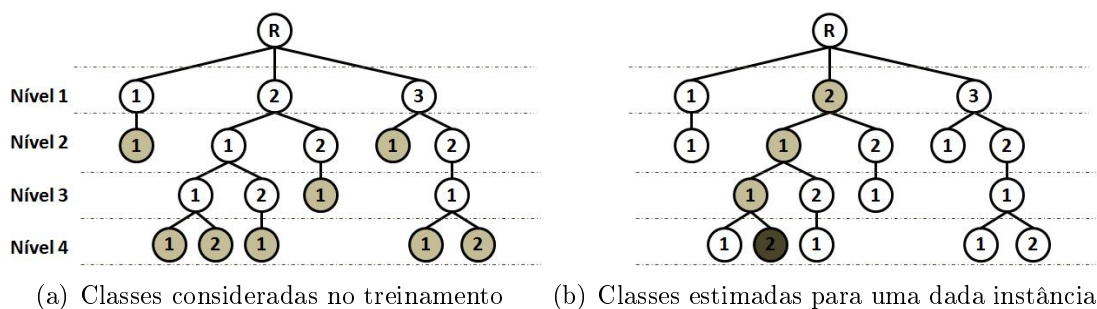


Figura 3.1: Funcionamento do classificador FL

3.1.2 Classificador Hierárquico *Per Parent Top-Down*

O classificador hierárquico *Per Parent Top-Down* (PPTD) baseia-se nos conceitos da abordagem de classificação local por nó pai (*Local Classification Per Parent Approach*). Nessa abordagem, realiza-se o treinamento de um classificador plano para cada classe não folha (nó interno) da hierarquia, incluindo o nó raiz. Em cada classificador plano, consideram-se apenas as classes filhas da classe pai. Dessa forma, obtém-se uma hierarquia de classificadores planos.

A classificação de uma nova instância é realizada de maneira *top-down*. Inicialmente, a instância é avaliada pelo classificador do nó raiz, que escolhe uma entre as suas classes filhas. A partir da classe resultante, o processo avança para o primeiro nível e o classificador do nó associado à classe resultante escolhe uma entre as suas classes filhas. Seguindo assim, sucessivamente, até se chegar a uma classe folha.

Na Figura 3.2, é ilustrado o funcionamento do classificador PPTD implementado. Inicialmente, o classificador associado ao nó raiz prevê a classe 2 (Figura 3.2(a)). Em seguida, o classificador associado ao nó que representa a classe 2 é ativado e prevê a classe 2.1 (Figura 3.2(b)). Seguindo procedimento semelhante, o classificador associado

ao nó da classe 2.1 é executado, resultando a classe 2.1.2 (Figura 3.2(c)) e, finalmente, o classificador do nó da classe 2.1.2 prevê a classe 2.1.2.1 (Figura 3.2(d)).

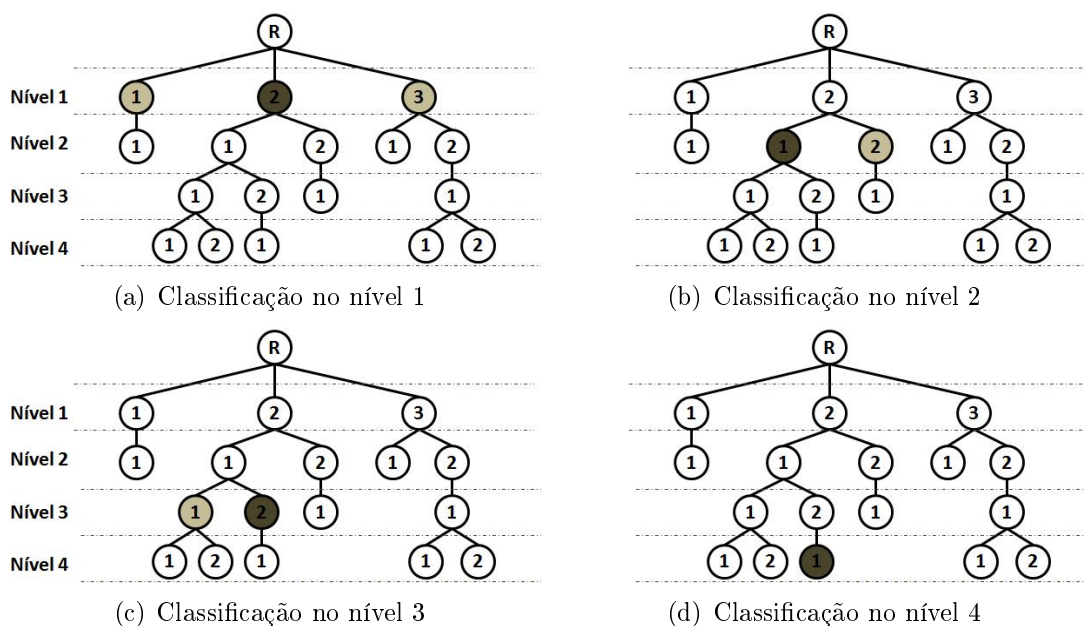


Figura 3.2: Funcionamento do classificador PPTD

3.2 Estratégias propostas

Os classificadores hierárquicos propostos neste trabalho têm como base os conceitos da abordagem de classificação local por nível (*Local Classification Per Level Approach*), apresentada no Capítulo 2. Basicamente, essa abordagem treina e executa um classificador plano para cada nível da hierarquia.

A Figura 3.3 ilustra uma hierarquia de classes de quatro níveis e, para cada um, foi treinado um classificador plano. Os diferentes classificadores, embora aplicados em níveis distintos, podem ser treinados a partir de um mesmo algoritmo de classificação. Por exemplo, os quatro classificadores C_1 , C_2 , C_3 e C_4 podem ser árvores de decisão produzidas pelo algoritmo C4.5.

O principal problema dessa abordagem, como visto no Capítulo 2, são as classificações inconsistentes. Quando o resultado dos classificadores gera um conjunto de classes que não constituem uma relação de descendência entre si, forma-se uma inconsistência. A Figura 3.4 ilustra uma inconsistência, na qual os nós sombreados representam os resultados dos classificadores planos aplicados. Observa-se que a classificação do terceiro nível não é consistente com o classificação do nível 2, pois a classe 3.2.1 não é descendente da classe 2.1.

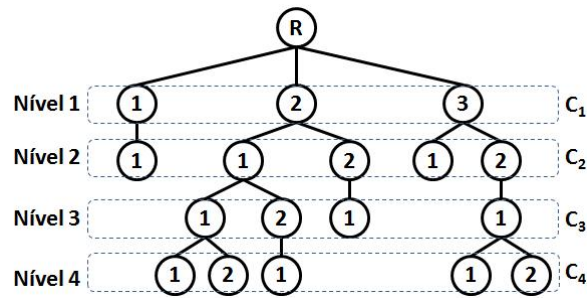


Figura 3.3: Classificadores planos C_1 , C_2 , C_3 e C_4 para cada nível da hierarquia

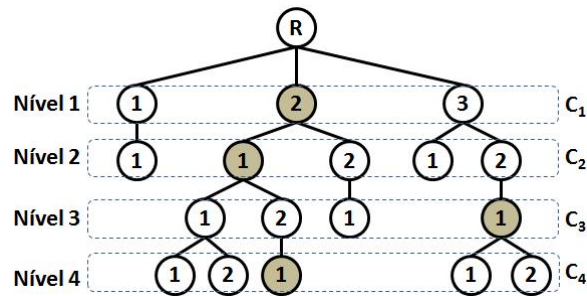


Figura 3.4: Exemplo de uma classificação inconsistente

Para tratar o problema em questão, este trabalho propõe e explora estratégias que decompõem o problema de classificação hierárquica em um conjunto de problemas de classificação plana e aplicam um tratamento posterior de inconsistências. Nas próximas subseções, serão apresentadas as duas estratégias hierárquicas propostas.

3.2.1 Classificador Hierárquico *Sum of Votes*

O primeiro classificador utiliza a estratégia hierárquica proposta *Sum of Votes* (SV), no qual o tratamento de inconsistência prioriza a ramificação ou caminho (definido da raiz até uma classe folha) com o maior número de classificações – ou votos – resultantes dos classificadores planos aplicados.

Inicialmente, identificam-se as ramificações que contêm pelo menos um voto, ou seja, pelo menos uma classe prevista por um dos classificadores planos. A partir dessas ramificações, diferentes cenários podem ocorrer, o que será discutido a seguir.

Cenário 1: Caracterizado pela existência de um caminho P com maior número de votos que os demais, de tal forma que a classe prevista do menor nível em P seja uma classe folha. Nesse caso, o tratamento de inconsistência consiste em, para cada nível N em que a classe prevista não pertence ao caminho P , trocar a classe prevista de N pela classe no caminho P pertencente ao nível N . A Figura 3.5 apresenta um exemplo deste

cenário. As classes previstas, em cada nível, são (Figura 3.5(a)): 2, 2.1, 3.2.1 e 2.1.2.1. O caminho com maior número de votos possui as classes previstas 2, 2.1 e 2.1.2.1, sendo a última uma classe folha. A classe prevista no nível 3, 3.2.1, não pertence a esse caminho e a eliminação de inconsistência transforma essa classe na classe 2.1.2 (Figura 3.5(b)).

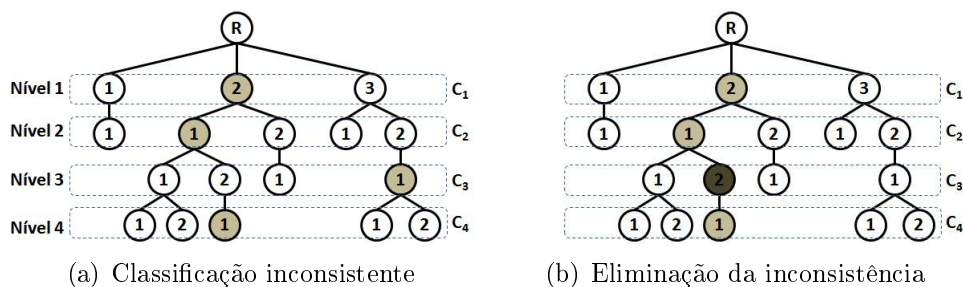


Figura 3.5: Ilustração do Cenário 1 para a estratégia SV

Cenário 2: Caracterizado pela existência de um caminho P com maior número de votos que os demais, como o Cenário 1, porém sem uma classe folha prevista em P . Nesse caso, o tratamento de inconsistência consiste em duas etapas: (1) para cada nível N superior ao menor nível (mais distante da raiz) que tem classe prevista em P , se a classe N não pertence a P , trocar a classe prevista de N pela classe no caminho P pertencente ao nível N ; (2) para os níveis inferiores ao menor nível com classe prevista em P , as classes serão estabelecidas de forma análoga ao método *Per Parent Top Down*, ou seja, para o nó da classe em P com menor nível, é criado um classificador plano considerando-se apenas as classes dos seus nós filhos. Esse classificador escolhe uma entre as suas classes filhas, incluindo essa classe resultante no caminho P . Caso essa classe incluída seja uma classe folha, o tratamento é concluído, caso contrário o processo descrito é repetido para o próximo nível, e é finalizado quando uma classe folha é incluída no caminho P . A Figura 3.6 apresenta um exemplo deste cenário. As classes previstas, em cada nível, são (Figura 3.6(a)): 3, 1.1, 3.2.1 e 2.1.1.2. O caminho com maior número de votos possui as classes previstas 3 e 3.2.1, sendo o nível 3 o menor nível com classe prevista em P . O tratamento, na primeira etapa, verifica que existe uma classe prevista no nível 2 (1.1) que não pertence a esse caminho e elimina a inconsistência transformando a classe prevista deste nível na classe 3.2 (Figura 3.6(b)). Na segunda etapa, a eliminação de inconsistência realiza o treinamento de um classificador plano associado a classe não folha 3.2.1, que escolhe entre uma das suas classes filhas, 3.2.1.1 e 3.2.1.2. A classe resultante, 3.2.1.2, é incluída no caminho e, como ela é uma classe folha, o tratamento de inconsistência é concluído (Figura 3.6(b)).

Cenário 3: Caracterizado pela existência de vários caminhos com maior número de votos.

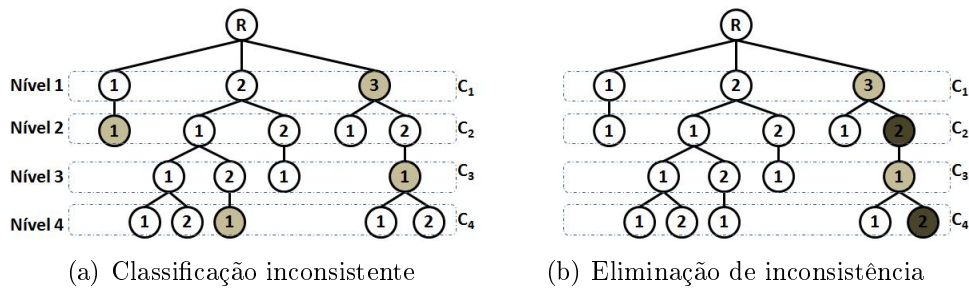


Figura 3.6: Ilustração do Cenário 2 para a estratégia SV

Nesse caso, o tratamento de inconsistência, inicialmente, escolhe, entre os caminhos com maior número de votos, aquele que possui a classe prevista pertencente ao maior nível (mais próximo à raiz) da hierarquia. A partir da escolha desse caminho P , se a classe prevista do menor nível em P for uma classe folha, o mesmo procedimento do Cenário 1 será adotado. Caso contrário, o mesmo procedimento do Cenário 2 será adotado. A Figura 3.7 apresenta um exemplo deste terceiro cenário. As classes previstas, em cada nível, são (Figura 3.7(a)): 3, 2.1, 2.1.2 e 3.2.1.2. Existem dois caminhos com dois votos cada, um possui as classes previstas 3 e 3.2.1.2, e o segundo, as classes 2.1 e 2.1.2. A eliminação de inconsistência inicialmente escolhe, entre os dois caminhos com dois votos, aquele que possui as classes previstas 3 e 3.2.1.2, pois a classe 3 é a classe prevista do maior nível. Após a escolha desse caminho, o exemplo segue o Cenário 1 descrito anteriormente, pois uma classe folha pertence a esse caminho. Dessa forma, a eliminação de inconsistência transforma as classes previstas 2.1 e 2.1.2, que não pertencem ao caminho escolhido, nas classes 3.2 e 3.2.1 (Figura 3.7(b)).

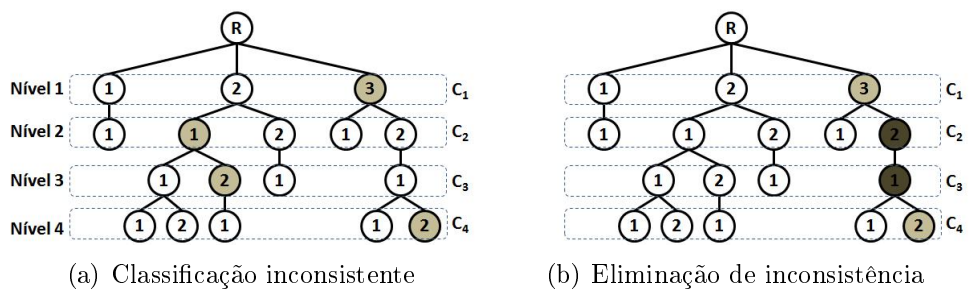


Figura 3.7: Ilustração do Cenário 3 para a estratégia SV

3.2.2 Classificador Hierárquico *Sum of Weighted Votes*

O segundo classificador utiliza a estratégia proposta *Sum of Weighted Votes* (SW), que consiste basicamente em ponderar os votos contabilizados pela estratégia anterior com as probabilidades estimadas pelos classificadores planos, quando realizadas as previsões das

classes dos diferentes níveis. Dessa forma, identificam-se as ramificações que contêm pelo menos um voto, ou seja, pelo menos uma classe prevista por um dos classificadores planos. A partir dessas ramificações, os votos são ponderados pelas probabilidades estimadas, e a ramificação que possuir a maior soma de votos ponderados será adotada para eliminar as inconsistências e definir as classes da instância. Para esta estratégia são definidos quatro cenários. Os três primeiros são similares aos definidos para a estratégia SV e serão redefinidos por questões de clareza.

Cenário 1: Caracterizado pela existência de um caminho P com maior soma de votos ponderados que os demais, de tal forma que a classe prevista do menor nível em P seja uma classe folha. Nesse caso, o tratamento de inconsistência consiste em, para cada nível N em que a classe prevista não pertence ao caminho P , trocar a classe prevista de N pela classe no caminho P pertencente ao nível N . A Figura 3.8 apresenta um exemplo deste cenário. As classes previstas, em cada nível, são (Figura 3.8(a)): 2, 2.1, 3.2.1 e 2.1.2.1. O caminho com maior soma de votos ponderados ($0,9+0,7+0,9=2,5$) possui as classes previstas 2, 2.1 e 2.1.2.1, sendo a última uma folha. A classe prevista no nível 3, 3.2.1, não pertence a esse caminho e a eliminação de inconsistência transforma essa classe na classe 2.1.2 (Figura 3.8(b)).

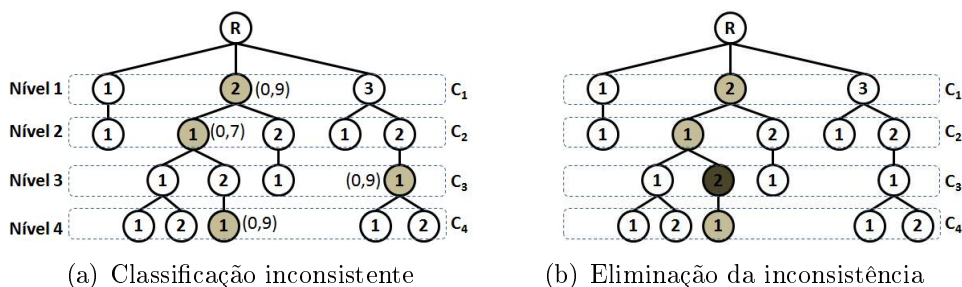


Figura 3.8: Ilustração do Cenário 1 para a estratégia SW

Cenário 2: Caracterizado pela existência de um caminho P com maior soma de votos ponderados que os demais, como o Cenário 1, porém sem uma classe folha prevista em P . Nesse caso, o tratamento de inconsistência consiste em duas etapas: (1) para cada nível N superior ao menor nível (mais distante da raiz) que tem classe prevista em P , se a classe N não pertence a P , trocar a classe prevista de N pela classe no caminho P pertencente ao nível N ; (2) para os níveis inferiores ao menor nível com classe prevista em P , as classes serão estabelecidas de forma análoga ao método *Per Parent Top Down*, ou seja, para o nó da classe em P com menor nível, é criado um classificador plano considerando-se apenas as classes dos seus nós filhos. Esse classificador escolhe uma entre as suas classes filhas, incluindo essa classe resultante no caminho P . Caso essa

classe incluída seja uma classe folha, o tratamento é concluído, caso contrário o processo descrito é repetido para o próximo nível, e é finalizado quando uma classe folha é incluída no caminho P . A Figura 3.9 apresenta um exemplo deste cenário. As classes previstas, em cada nível, são (Figura 3.9(a)): 3, 1.1, 3.2.1 e 2.1.1.2. O caminho com maior soma de votos ponderados ($0,8+0,7=1,5$) possui as classes previstas 3 e 3.2.1, sendo o nível 3 o menor nível com classe prevista em P . O tratamento, na primeira etapa, verifica que existe uma classe prevista no nível 2 (1.1) que não pertence a esse caminho e elimina a inconsistência transformando a classe prevista deste nível na classe 3.2 (Figura 3.9(b)). Na segunda etapa, a eliminação de inconsistência realiza o treinamento de um classificador plano associado a classe não folha 3.2.1, que escolhe entre uma das suas classes filhas, 3.2.1.1 e 3.2.1.2. A classe resultante, 3.2.1.2, é incluída no caminho e, como ela é uma classe folha, o tratamento de inconsistência é concluído (Figura 3.9(b)).

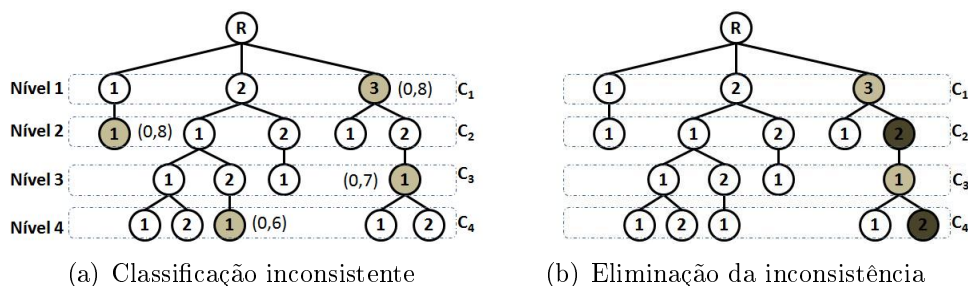


Figura 3.9: Ilustração do Cenário 2 para a estratégia SW

Cenário 3: Caracterizado pela existência de vários caminhos com maior soma de votos ponderados. Nesse caso, o tratamento de inconsistência, inicialmente, escolhe, entre os caminhos com maior número de votos ponderados, aquele que possui a classe prevista pertencente ao maior nível (mais próximo à raiz) da hierarquia. A partir da escolha desse caminho P , se a classe prevista do menor nível em P for uma classe folha, o mesmo procedimento do Cenário 1 será adotado. Caso contrário, o mesmo procedimento do Cenário 2 será adotado. A Figura 3.10 apresenta um exemplo deste cenário. As classes previstas, em cada nível, são (Figura 3.10(a)): 3, 2.1, 2.1.2 e 3.2.1.2. Existem dois caminhos com soma de votos ponderados iguais, um possui as classes previstas 3 e 3.2.1.2, e o segundo, as classes 2.1 e 2.1.2. A eliminação de inconsistência inicialmente escolhe o caminho que possui as classes previstas 3 e 3.2.1.2, pois a classe 3 é a classe prevista do maior nível. Após a escolha desse caminho, o exemplo segue o Cenário 1 descrito anteriormente, pois uma classe folha pertence a esse caminho. Dessa forma, a eliminação de inconsistência transforma as classes previstas 2.1 e 2.1.2, que não pertencem ao caminho escolhido, nas classes 3.2 e 3.2.1 (Figura 3.10(b)).

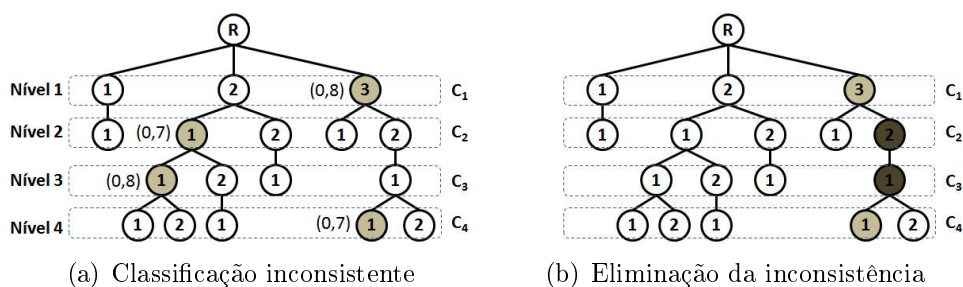


Figura 3.10: Ilustração do Cenário 3 para a estratégia SW

Além desses três cenários, foi definido um quarto cenário motivado pela situação ilustrada pela Figura 3.11, que apresenta o resultado da classificação obtida em cada nível da hierarquia para uma determinada instância. Considere que, nos dois primeiros níveis, as classes 1 e 1.1 obtidas sejam as duas classes reais da instância. Observe que apenas essas duas classes constituem um caminho de classes consistentes até um nó folha. Porém, como o classificador hierárquico por nível obtém sempre a classificação para todos os níveis, as classes 3.2.1 e 3.2.1.1, obtidas para os outros dois níveis, também são analisadas para escolha do caminho com maior soma de probabilidades. Nota-se que o caminho constituído pelas classes 3.2.1 e 3.2.1.1 seria o escolhido segundo o critério de tratamento de inconsistência estabelecido no Cenário 1. Portanto, considerando que a instância em questão pertence às classes 1 e 1.1, o processo de eliminação de inconsistência acabou gerando um erro em uma classificação correta por ser obrigado a prever classe em todos os níveis. Dessa forma, faz-se necessária a definição do tratamento de um quarto cenário, descrito detalhadamente a seguir, para considerar o caso em que a classe folha de uma instância não pertence ao último nível da hierarquia.

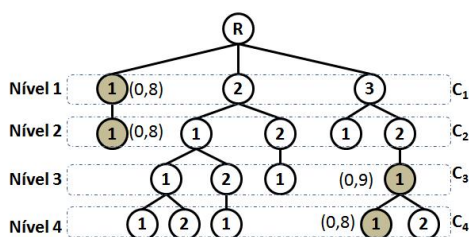


Figura 3.11: Motivação para o Cenário 4 para a estratégia SW

Cenário 4: Caracterizado pela existência de um caminho P com classe prevista no nível 1, com uma classe folha prevista no nível i e com classes previstas e consistentes com as duas classes anteriores em cada nível entre 1 e i . Nesse caso, o tratamento de inconsistência escolhe as classes do caminho P , mesmo que existam outros caminhos com soma maior de votos ponderados, e não atribui classes nos níveis descendentes da classe folha, caso existam. Com esse caminho P , o tratamento de inconsistência é finalizado.

Este quarto cenário sempre ocorrerá simultaneamente com um dos outros três cenários descritos anteriormente e terá prioridade no tratamento de inconsistência sempre que ele ocorrer.

A Figura 3.12 apresenta um exemplo deste cenário. As classes previstas, em cada nível, são (Figura 3.12(a)): 1, 1.1, 3.2.1 e 3.2.1.2. A eliminação de inconsistência verifica a existência de um caminho que possui as classes previstas 1 e 1.1, sendo a classe 1 pertencente ao primeiro nível da hierarquia e a classe filha 1.1 uma classe folha. Dessa forma, mesmo com a existência de outro caminho com maior soma de votos ponderados, a eliminação de inconsistência considera as classes previstas pertencentes a esse caminho (Figura 3.12(b)).

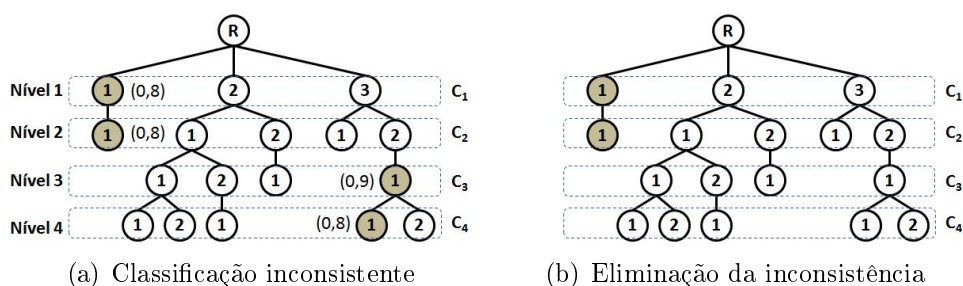


Figura 3.12: Ilustração do Cenário 4 para a estratégia SW

3.3 Descrição dos Experimentos

Nos experimentos computacionais, foram utilizadas 18 bases de dados para avaliação do desempenho dos classificadores hierárquicos. Encontra-se a seguir uma breve descrição dessas bases de dados, que pertencem a dois grandes grupos:

- **Grupo A:** Oito bases de dados contendo informações de funções de proteínas.

Essas bases se dividem em dois subgrupos: GPCR (*G-Protein-Coupled Receptor*) e EC (*Enzyme Commission*). O grupo GPCR é composto por quatro bases de dados (GPCRpfam, GPCRprints, GPCRprosite e GPCRinterpro). GPCRs são proteínas que transmitem sinais do meio externo para dentro da célula, influenciando reações químicas na célula. Os diferentes nomes das bases referem-se aos diferentes tipos de atributos preditivos usados em cada base de dados, onde Pfam, Prints, Prosite e Interpro são diferentes tipos de *protein motifs*. O grupo EC é composto por quatro bases (ECpfam, ECprints, ECprosite e ECinterpro), que representam funções de enzimas. Uma enzima é um tipo de proteína que acelera reações químicas. Essas bases

estão disponíveis em <http://www.cs.kent.ac.uk/archive/people/rpg/nh56/datasets.zip>, um repositório da Universidade de Kent. As bases do grupo GPCR e EC têm sido utilizadas em diversos trabalhos que tratam de problemas de classificação hierárquica [8, 7, 13, 19, 30]. Para os experimentos realizados neste trabalho, um pré-processamento foi realizado para retirar todas as instâncias (de cada base) cuja classe mais específica não estivesse associada a um nó folha.

- **Grupo B:** Dez bases de dados contendo informações de funções gênicas.

As bases deste grupo são provenientes do campo da genômica funcional, relacionando-se ao fungo *Saccharomyces cerevisiae* ou *Yeast* (Levedura) e são apresentadas em [5]. São originalmente multirrótulo e para utilização neste trabalho (onde assumi-se que as instâncias são monorrótulo), essas bases foram convertidas através das escolha aleatória de uma das classes associadas a cada instância.

Todas as bases possuem a hierarquia de classes representada por uma estrutura de árvore não completa, composta por quatro níveis. Além disso, as classes mais específicas das instâncias estão associadas somente a nós folha da hierarquia de classes. As características das bases são apresentadas na Tabela 3.1. Pode-se observar nessa tabela: o grupo ao qual a base pertence (Grupo), o nome da base (Base), o número de classes para cada nível da hierarquia (#Classes) e o total de instâncias de cada base (#Instâncias).

Tabela 3.1: Características das bases de dados

Grupo	Base	#Classes	#Instâncias
A	GPCRpfam	12/52/79/49	6524
	GPCRprints	8/46/76/49	4880
	GPCRprosite	9/50/79/49	5728
	GPCRinterpro	12/54/82/50	6935
	ECpfam	6/41/96/190	11057
	ECprints	6/45/92/208	11048
	ECprosite	6/42/89/187	11328
	ECinterpro	6/41/96/187	11101
	B	Church	4/18/36/27
CellCycle		4/17/34/23	1711
Derisi		4/18/35/25	1661
Eisen		4/15/29/17	1163
Expr		4/17/34/25	1688
Gasch1		4/17/34/25	1660
Gasch2		4/17/33/25	1678
Phenotype		4/12/21/13	621
Sequence		4/17/32/24	1680
SPO		4/17/34/25	1649

Todos os classificadores hierárquicos foram implementados na linguagem de programação *JAVA*, utilizando algoritmos e funções da ferramenta de mineração de dados *WEKA 3.7.0 (Waikato Environment for Knowledge Analysis)* [34]. Foram utilizados dois classificadores planos tradicionais: um do tipo *eager*, C4.5, e outro do tipo *lazy*, k-NN. Para

representar esses classificadores planos utilizados nos experimentos, foram adotadas as versões disponibilizadas na ferramenta WEKA denominados, respectivamente, J48 e Ibk.

A avaliação dos classificadores foi efetuada utilizando-se validação cruzada com 10 partições e foi adotada a medida hierárquica hF, apresentada no Capítulo 2. Para avaliar a significância estatística na comparação entre dois classificadores hierárquicos, utilizou-se o teste-t de *Student* [20], bicaudal e pareado (com nível de confiança de 95%, ou seja, p-valor de 5%).

3.4 Resultados Computacionais

Nesta seção, são apresentados os resultados e análises dos experimentos computacionais. Inicialmente, compara-se o desempenho dos dois classificadores com as estratégias propostas (*Sum of Votes* e *Sum of Weighted Votes*) e, posteriormente, são realizadas as comparações entre os classificadores hierárquicos tradicionais (*Flat on Leaves* e *Per Parent Top-Down*) e o classificador hierárquico com a estratégia proposta que obteve o melhor desempenho.

3.4.1 Comparação entre as Estratégias Propostas

As estratégias propostas *Sum of Votes*(SV) e *Sum of Weighted Votes*(SW) foram comparadas considerando-se cada um dos seis classificadores planos utilizados: 1-NN, 3-NN, 5-NN, 7-NN, 9-NN e C4.5. Ou seja, cada um desses seis classificadores planos foi aplicado, separadamente, em cada um dos níveis da hierarquia e, em seguida, as estratégias hierárquicas SV e SW adotam suas heurísticas para a eliminação de inconsistências.

Para cada base de dados e classificador plano adotado, na Tabela 3.2, são apresentados os valores da medida hF obtidos por validação cruzada com 10 partições (com o respectivo desvio padrão entre parênteses). Em negrito estão marcados os melhores resultados para cada classificador plano aplicado. Os melhores resultados por base de dados aparecem sublinhados. O símbolo (●) entre os dois valores de hF indica que a diferença entre esses valores possui significância estatística. O símbolo (-) indica que não houve significância estatística. Por exemplo, para a base *GPRCpfam*, com classificador plano 1-NN, a estratégia SW obteve hF de 70,31, superando com significância estatística a estratégia SV, que obteve hF de 67,31. Além disso, o valor 70,31 está sublinhado pois foi o melhor resultado encontrado para a base *GPCRpfam*, nesta análise. Por fim, abaixo de cada grupo de bases de dados, tem-se uma linha de totais, que apresenta o número de

vezes que uma das estratégias hierárquicas apresentou o valor hF maior ou igual ao valor hF da outra, para cada classificador plano adotado.

Tabela 3.2: Comparação entre os classificadores SV e SW

Bases	1-NN		3-NN		5-NN	
	SV	SW	SV	SW	SV	SW
GPCRpfam	67,31 (1,55) ● 70,31 (1,52)		66,60 (0,94) ● 69,72 (1,08)		66,44 (1,02) ● 69,20 (1,32)	
GPCRprints	82,17 (1,15) ● 83,00 (1,2)		81,42 (0,79) ● 82,09 (0,82)		80,41 (0,82) ● 81,05 (0,87)	
GPCRprosite	68,92 (1,18) ● 69,26 (1,19)		67,46 (0,92) ● 68,05 (0,86)		67,66 (0,49) - 67,83 (0,58)	
GPCRinterpro	80,14 (0,81) ● 83,09 (0,7)		79,85 (0,94) ● 82,73 (0,91)		79,87 (0,91) ● 82,15 (0,73)	
ECpfam	98,77 (0,17) - 98,77 (0,17)		98,45 (0,33) - 98,44 (0,31)		98,27 (0,36) ● 98,37 (0,33)	
ECprints	98,16 (0,24) - 98,19 (0,24)		97,86 (0,25) - 97,86 (0,26)		97,63 (0,31) - 97,64 (0,3)	
ECprosite	98,78 (0,27) ● 98,80 (0,26)		98,66 (0,21) - 98,67 (0,2)		98,46 (0,27) - 98,47 (0,27)	
ECinterpro	99,08 (0,29) - 99,08 (0,29)		98,81 (0,4) - 98,83 (0,4)		98,78 (0,42) - 98,83 (0,41)	
Total A	2	8	2	7	0	8
Church	19,70 (1,97) - 19,70 (1,97)		19,63 (2,19) ● 20,27 (1,94)		20,27 (2,02) ● 20,53 (2,18)	
CellCycle	24,82 (2,34) - 24,82 (2,34)		23,77 (3,38) ● 24,52 (3,59)		27,52 (2,46) - 28,15 (2,98)	
Derisi	18,73 (2,33) - 18,73 (2,33)		16,58 (2,26) ● 17,94 (2,57)		18,88 (2,46) ● 20,06 (2,54)	
Eisen	24,54 (2,32) - 24,54 (2,32)		23,66 (2,87) ● 25,19 (3,4)		27,36 (3,14) - 28,31 (2,95)	
Expr	25,62 (3,56) - 25,62 (3,56)		22,42 (1,48) ● 23,44 (1,92)		26,37 (2,36) - 27,13 (2,63)	
Gasch1	28,98 (2,01) - 28,98 (2,01)		27,29 (1,51) ● 28,62 (1,77)		29,50 (2,42) - 30,07 (1,88)	
Gasch2	25,04 (3,03) - 25,03 (3,04)		22,92 (2,95) - 23,26 (2,71)		23,57 (3,43) - 23,92 (3,16)	
Phenotype	22,04 (2,57) - 22,46 (2,91)		23,14 (2,05) - 23,66 (3,24)		22,75 (1,82) - 23,23 (2,11)	
Sequence	23,08 (3,14) - 23,08 (3,14)		21,73 (2,25) - 22,38 (2,56)		22,22 (2,66) ● 22,82 (2,69)	
SPO	18,49 (2,66) - 18,50 (2,68)		17,81 (1,6) ● 19,15 (1,89)		19,42 (2,63) ● 19,92 (2,73)	
Total B	8	9	0	10	0	10

Bases	7-NN		9-NN		C4.5	
	SV	SW	SV	SW	SV	SW
GPCRpfam	66,15 (0,97) ● 68,68 (1,16)		65,58 (0,93) ● 68,07 (1,1)		68,79 (1,24) ● 68,70 (1,21)	
GPCRprints	80,33 (1,07) ● 80,86 (1,12)		79,75 (0,91) ● 80,28 (0,89)		78,70 (0,83) ● 79,33 (0,85)	
GPCRprosite	66,67 (1,03) - 66,83 (1)		65,95 (0,9) - 66,00 (0,87)		67,24 (0,85) - 67,08 (1)	
GPCRinterpro	79,57 (0,68) ● 81,68 (0,53)		79,17 (1,18) ● 81,31 (0,91)		81,26 (1,09) ● 81,80 (1,03)	
ECpfam	98,27 (0,34) - 98,30 (0,29)		98,19 (0,31) - 98,19 (0,31)		98,42 (0,23) - 98,43 (0,23)	
ECprints	97,45 (0,26) - 97,45 (0,26)		97,25 (0,37) - 97,22 (0,41)		97,47 (0,26) - 97,51 (0,31)	
ECprosite	98,33 (0,32) - 98,33 (0,33)		98,09 (0,3) - 98,08 (0,3)		98,57 (0,18) - 98,57 (0,18)	
ECinterpro	98,80 (0,32) - 98,82 (0,34)		98,71 (0,34) - 98,70 (0,32)		98,79 (0,38) - 98,79 (0,38)	
Total A	2	8	4	5	4	6
Church	20,25 (2,1) - 20,19 (2,1)		20,40 (2,05) - 20,66 (2,04)		21,29 (2,39) - 21,29 (2,47)	
CellCycle	28,67 (3,05) - 28,60 (3,28)		29,85 (4,24) - 29,87 (3,86)		23,96 (2,7) ● 24,83 (3,1)	
Derisi	20,38 (2,61) ● 21,16 (2,6)		20,71 (2,43) - 21,20 (2,66)		21,59 (2,14) - 21,78 (1,97)	
Eisen	29,14 (2,69) - 29,56 (2,8)		30,39 (2,05) - 30,77 (1,83)		25,27 (2,86) - 25,79 (2,67)	
Expr	27,94 (2,99) - 28,12 (3,08)		29,22 (3,26) - 29,26 (2,52)		26,72 (2,45) - 26,61 (2,53)	
Gasch1	31,33 (2,23) - 31,24 (2,22)		30,59 (2,62) - 30,80 (2,56)		25,24 (2,69) - 25,86 (2,28)	
Gasch2	25,28 (2,54) - 25,55 (2)		25,47 (1,68) - 26,21 (1,34)		23,45 (2,63) - 23,39 (2,21)	
Phenotype	24,15 (2,87) - 24,42 (3,01)		25,46 (2,96) - 25,45 (3,32)		26,50 (2,58) - 26,37 (3,35)	
Sequence	22,78 (3,32) - 22,76 (2,94)		23,70 (2,57) - 23,75 (2,65)		24,44 (3,01) ● 25,71 (2,87)	
SPO	21,40 (2,3) - 21,59 (2,25)		22,01 (1,63) - 22,31 (1,64)		20,90 (2,2) - 21,50 (2,58)	
Total B	4	6	1	9	4	7

Analisando-se a Tabela 3.2, observa-se que a estratégia SW apresentou um desempenho superior a da estratégia SV em vários aspectos. Para os dois grupos de bases e todos os classificadores planos considerados, observa-se através das linhas totais que o número de melhores resultados (negritos) obtidos pela estratégia SW é sempre maior do que o da estratégia SV. Observa-se também que a quantidade de melhores resultados por base (sublinhados) obtidos pela estratégia SW (16 no total) é maior do que a obtida pela SV (cinco no total).

Percebe-se ainda que os melhores resultados por base de dados (sublinhados) para

as bases do Grupo A concentram-se quando o classificador plano 1-NN é aplicado. Por outro lado, para as bases dos Grupo B, os melhores resultados foram obtidos quando os classificadores planos 9-NN e C4.5 foram adotados.

Baseando-se nesses resultados, conclui-se que a estratégia *Sum of Weighted Votes* apresenta um melhor desempenho, e dessa forma, será utilizado para as próximas avaliações com os classificadores hierárquicos tradicionais (*Flat on Leaves* e *Per Parent Top-Down*).

3.4.2 Comparação da Estratégia Proposta SW com as Estratégias Tradicionais

O objetivo da próxima avaliação experimental é realizar uma comparação entre o classificador hierárquico com a melhor estratégia proposta *Sum of Weighted Votes* e os classificadores hierárquicos tradicionais *Per Parent Top-Down*(PPTD) e *Flat on Leaves*(FL).

Nas Tabelas 3.3 e 3.4, para as bases dos Grupos A e B, respectivamente, são apresentados os resultados dos classificadores tradicionais PPTD e FL, e do classificador com a estratégia proposta SW, considerando cada um dos classificadores planos adotados. Para cada base de dados e classificador plano adotado têm-se os três valores da medida hF e seus respectivos desvios padrões para as três estratégias. O melhor resultado entre as três estratégias, para cada classificador plano, está marcado em negrito. O símbolo (\blacktriangle) ao lado direito do valor hF obtido pelos classificadores tradicionais FL e PPTD indica que o valor obtido pela estratégia proposta SW superou com significância estatística o valor obtido pelo respectivo classificador tradicional. O símbolo (\blacktriangledown) indica que o valor hF obtido pelo classificador tradicional superou, com significância estatística, o resultado da estratégia SW. O símbolo (-) indica que a diferença de desempenho dos classificadores comparados não apresentou significância estatística. Por exemplo, na Tabela 3.3, para a base *GPCRpfam*, considerando o classificador plano 3-NN, o símbolo (\blacktriangle) ao lado do resultado do classificador FL representa que o resultado do classificador com a estratégia SW foi superior ao resultado obtido pelo classificador FL, com significância estatística.

Apresenta-se, a seguir, a análise dos resultados obtidos pelos classificadores hierárquicos, para cada grupo de bases aplicadas:

- Grupo A: Analisando-se a Tabela 3.3, observa-se que, para as bases do Grupo A, não houve uma diferença significativa de desempenho entre os três classificadores. Através das linhas de totais, observa-se que o número de melhores resultados (negritos) obtidos pela estratégia SW foi igual ou superior ao dos classificadores hierárquicos

tradicionais FL e PPTD quando foram adotados os classificadores planos 1-NN e 5-NN. Por outro lado, o classificador PPTD obteve um desempenho superior quando o classificador plano 3-NN foi utilizado. Já o classificador FL foi superior quando os classificadores planos 7-NN, 9-NN e C4.5 foram adotados. Observa-se, também, que todos os melhores resultados por base (sublinhados) foram obtidos com a aplicação do classificador plano 1-NN e, nesse caso, as três estratégias obtiveram desempenhos semelhantes. Conclui-se, portanto, que, para as bases do Grupo A, o classificador com a estratégia proposta SW mostrou-se competitivo em comparação com os classificadores tradicionais FL e PPTD.

Tabela 3.3: Comparação dos classificadores SW, FL e PPTD - Bases do Grupo A

Bases	1-NN			3-NN		
	FL	PPTD	SW	FL	PPTD	SW
GPCRpfam	70,31 (1,51) - 70,32 (1,52)	70,31 (1,52)	70,31 (1,52)	69,34 (1,14) ▲ 70,07 (0,88)	69,72 (1,08)	69,72 (1,08)
GPCRprints	82,98 (1,21) - <u>82,97</u> (1,19) - 83,00 (1,2)	82,97 (1,19)	83,00 (1,2)	82,00 (0,79) ▲ 82,06 (0,7)	82,09 (0,82)	82,09 (0,82)
GPCRprosite	69,28 (1,19) - 69,25 (1,23) - <u>69,26</u> (1,19)	69,25 (1,23)	69,26 (1,19)	67,81 (0,99) ▲ 68,39 (1,11)	68,05 (0,86)	68,05 (0,86)
GPCRinterpro	83,09 (0,71) - 83,09 (0,7) - 83,09 (0,7)	83,09 (0,7)	83,09 (0,7)	82,63 (0,83) - 82,85 (0,94)	82,73 (0,91)	82,73 (0,91)
ECpfam	98,77 (0,17) - 98,77 (0,17) - 98,77 (0,17)	98,77 (0,17)	98,77 (0,17)	98,44 (0,33) - 98,42 (0,31) - 98,44 (0,31)	98,44 (0,31)	98,44 (0,31)
ECprints	98,19 (0,24) - 98,19 (0,23) - 98,19 (0,24)	98,19 (0,23)	98,19 (0,24)	97,87 (0,27) - 97,87 (0,25)	97,86 (0,26)	97,86 (0,26)
ECprosite	98,80 (0,26) - 98,80 (0,26) - 98,80 (0,26)	98,80 (0,26)	98,80 (0,26)	98,70 (0,19) - 98,66 (0,21) - 98,67 (0,2)	98,67 (0,2)	98,67 (0,2)
ECinterpro	99,08 (0,29) - 99,07 (0,29) - 99,08 (0,29)	99,07 (0,29)	99,08 (0,29)	98,79 (0,42) ▲ 98,85 (0,37)	98,83 (0,4)	98,83 (0,4)
Total A	6	5	6	3	5	2

Bases	5-NN			7-NN		
	FL	PPTD	SW	FL	PPTD	SW
GPCRpfam	69,24 (1,21) - 69,53 (0,93) - 69,20 (1,32)	69,20 (1,32)	69,20 (1,32)	69,33 (1,26) ▼ 69,04 (0,9)	68,68 (1,16)	68,68 (1,16)
GPCRprints	81,55 (0,84) ▼ 81,03 (0,89) - 81,05 (0,87)	81,03 (0,89)	81,05 (0,87)	81,28 (1,06) ▼ 80,89 (1,08)	80,86 (1,12)	80,86 (1,12)
GPCRprosite	67,97 (0,58) - 68,34 (0,64) ▼ 67,83 (0,58)	67,83 (0,58)	67,83 (0,58)	67,66 (0,91) ▼ 67,31 (0,98)	66,83 (1)	66,83 (1)
GPCRinterpro	82,43 (0,75) ▼ 82,20 (0,78) - 82,15 (0,73)	82,20 (0,78)	82,15 (0,73)	82,03 (0,58) - 81,95 (0,69)	81,68 (0,53)	81,68 (0,53)
ECpfam	98,26 (0,42) ▲ 98,24 (0,38) ▲ 98,37 (0,33)	98,24 (0,38)	98,37 (0,33)	98,27 (0,41) - 98,16 (0,36) ▲ 98,30 (0,29)	98,30 (0,29)	98,30 (0,29)
ECprints	97,63 (0,3) - 97,60 (0,32) - 97,64 (0,3)	97,60 (0,32)	97,64 (0,3)	97,49 (0,28) - 97,37 (0,28)	97,45 (0,26)	97,45 (0,26)
ECprosite	98,47 (0,25) - 98,45 (0,27) - 98,47 (0,27)	98,45 (0,27)	98,47 (0,27)	98,37 (0,31) ▼ 98,29 (0,32)	98,33 (0,33)	98,33 (0,33)
ECinterpro	98,70 (0,4) ▲ 98,76 (0,41) - 98,83 (0,41)	98,76 (0,41)	98,83 (0,41)	98,73 (0,3) ▼ 98,62 (0,32)	98,82 (0,34)	98,82 (0,34)
Total A	3	2	4	6	0	2

Bases	9-NN			C4.5		
	FL	PPTD	SW	FL	PPTD	SW
GPCRpfam	68,62 (1,34) - 68,47 (0,83) - 68,07 (1,1)	68,47 (0,83)	68,07 (1,1)	69,32 (1,23) - 68,84 (1,14) - 68,70 (1,21)	68,70 (1,21)	68,70 (1,21)
GPCRprints	80,54 (1,06) - 80,41 (0,96) - 80,28 (0,89)	80,41 (0,96)	80,28 (0,89)	81,76 (0,52) ▼ 79,19 (0,94) - 79,33 (0,85)	79,33 (0,85)	79,33 (0,85)
GPCRprosite	66,67 (0,87) ▼ 66,57 (0,7) ▼ 66,00 (0,87)	66,57 (0,7)	66,00 (0,87)	68,03 (1,01) ▼ 67,63 (1,17) ▼ 67,08 (1)	67,08 (1)	67,08 (1)
GPCRinterpro	81,99 (0,9) ▼ 81,29 (1,05) - 81,31 (0,91)	81,29 (1,05)	81,31 (0,91)	82,66 (0,72) ▼ 81,54 (0,57) - 81,80 (1,03)	81,80 (1,03)	81,80 (1,03)
ECpfam	98,18 (0,33) - 97,88 (0,33) ▲ 98,19 (0,31)	97,88 (0,33)	98,19 (0,31)	98,33 (0,24) ▲ 98,39 (0,24) - 98,43 (0,23)	98,43 (0,23)	98,43 (0,23)
ECprints	97,27 (0,38) - 97,05 (0,34) ▲ 97,22 (0,41)	97,05 (0,34)	97,22 (0,41)	97,56 (0,3) - 97,35 (0,27) ▲ 97,51 (0,31)	97,51 (0,31)	97,51 (0,31)
ECprosite	98,16 (0,29) ▼ 98,03 (0,31) ▲ 98,08 (0,3)	98,03 (0,31)	98,08 (0,3)	98,50 (0,19) - 98,46 (0,17) ▲ 98,57 (0,18)	98,57 (0,18)	98,57 (0,18)
ECinterpro	98,72 (0,35) - 98,32 (0,4) ▲ 98,70 (0,32)	98,32 (0,4)	98,70 (0,32)	98,74 (0,38) ▲ 98,73 (0,34) - 98,79 (0,38)	98,79 (0,38)	98,79 (0,38)
Total A	7	0	1	5	0	3

- Grupo B: Analisando-se a Tabela 3.4, percebe-se que, para as bases do Grupo B, a estratégia proposta SW obteve um desempenho superior ao das outras duas. Através das linhas de totais, observa-se que o número de melhores resultados (negritos) obtidos pela estratégia SW foi superior ao dos classificadores FL e PPTD quando foram adotados os classificadores planos 5-NN, 7-NN, 9-NN e C4.5, muitos desses

com significância estatística. Observa-se, principalmente, que nove dos 10 melhores resultados por base (sublinhados) foram obtidos pela estratégia SW. Conclui-se que, para as bases do Grupo B, o classificador com a estratégia SW mostrou-se superior em comparação aos classificadores tradicionais FL e PPTD.

Tabela 3.4: Comparação dos classificadores SW, FL e PPTD - Bases do Grupo B

Bases	1-NN			3-NN		
	FL	PPTD	SW	FL	PPTD	SW
Church	21,26 (1,58) ▼	19,38 (1,9)	19,70 (1,97)	20,44 (1,72)	20,20 (2,24)	20,27 (1,94)
CellCycle	24,83 (2,55)	24,56 (2,1)	24,82 (2,34)	23,00 (3,27)	24,79 (3,62)	24,52 (3,59)
Derisi	18,87 (2,5)	18,89 (2,23)	18,73 (2,33)	15,84 (2,05) ▲	19,05 (2,5)	17,94 (2,57) ▼
Eisen	24,72 (2,88)	24,70 (2,02)	24,54 (2,32)	23,22 (2,6)	25,31 (3,97)	25,19 (3,4)
Expr	25,05 (3,4)	▲ 25,35 (3,28)	25,62 (3,56)	21,65 (1,16) ▲	24,11 (2,72)	23,44 (1,92)
Gasch1	28,84 (1,7)	28,29 (1,47)	28,98 (2,01)	25,85 (2,56) ▲	29,25 (2,45)	28,62 (1,77)
Gasch2	24,58 (3,31)	25,23 (2,97)	25,03 (3,04)	21,51 (2,86) ▲	23,98 (2,54)	23,26 (2,71)
Phenotype	20,67 (3,75) ▲	20,27 (3,66)	22,46 (2,91)	22,29 (3,8)	21,51 (2,76)	23,66 (3,24)
Sequence	22,43 (2,97)	24,02 (3,42)	23,08 (3,14) ▼	20,44 (1,94) ▲	23,01 (2,7)	22,38 (2,56)
SPO	18,23 (2,26)	18,86 (2,41)	18,50 (2,68)	16,89 (2,11) ▲	19,94 (1,74)	19,15 (1,89)
Total B	3	4	3	1	8	1

Bases	5-NN			7-NN		
	FL	PPTD	SW	FL	PPTD	SW
Church	21,82 (1,53)	19,53 (2,84)	20,53 (2,18)	21,72 (1,11) ▲	19,66 (2,27) ▲	20,19 (2,1)
CellCycle	25,52 (2,44) ▲	27,36 (2,78)	28,15 (2,98)	27,47 (2,16)	28,38 (3,33)	28,60 (3,28)
Derisi	17,11 (2,53) ▲	18,83 (2,51)	20,06 (2,54)	18,88 (2,83) ▲	20,21 (2,61)	21,16 (2,6)
Eisen	25,24 (2,63) ▲	26,40 (3,36) ▲	28,31 (2,95)	28,28 (2,49)	29,25 (2,47)	29,56 (2,8)
Expr	23,87 (2,33) ▲	26,47 (1,99)	27,13 (2,63)	25,92 (1,84) ▲	27,21 (3,11)	28,12 (3,08)
Gasch1	26,62 (2,39) ▲	29,76 (2,55)	30,07 (1,88)	29,16 (2,86)	30,97 (2,67)	31,24 (2,22)
Gasch2	22,56 (3,11) ▲	24,79 (3,52)	23,92 (3,16)	24,37 (2,67)	26,12 (2,66)	25,55 (2)
Phenotype	23,10 (4,06)	21,33 (3,13)	23,23 (2,11)	24,57 (3,51)	22,52 (4,06)	24,42 (3,01)
Sequence	21,85 (2,93)	23,60 (2,24)	22,82 (2,69)	23,59 (2,3)	23,73 (3,39)	22,76 (2,94)
SPO	17,35 (2,16) ▲	20,41 (2,54)	19,92 (2,73)	19,14 (1,79) ▲	21,10 (3,19)	21,59 (2,25)
Total B	1	3	6	2	2	6

Bases	9-NN			C4.5		
	FL	PPTD	SW	FL	PPTD	SW
Church	22,10 (1,41)	19,82 (2,66)	20,66 (2,04)	21,74 (1,76)	21,53 (1,93)	21,29 (2,47)
CellCycle	29,08 (2,3)	29,40 (2,95)	29,87 (3,86)	21,86 (3,75) ▲	22,19 (2,41) ▲	24,83 (3,1)
Derisi	20,17 (3,14)	20,89 (2,36)	21,20 (2,66)	18,28 (2,77) ▲	20,47 (2,7)	21,78 (1,97)
Eisen	28,73 (3,02) ▲	29,16 (2,63)	30,77 (1,83)	22,96 (3,14)	24,24 (3,19)	25,79 (2,67)
Expr	27,58 (2,69) ▲	28,29 (2,19)	29,26 (2,52)	24,98 (2,93)	24,74 (2,22) ▲	26,61 (2,53)
Gasch1	30,40 (2,41)	30,08 (2,9)	30,80 (2,56)	23,09 (2,95) ▲	22,90 (2,05) ▲	25,86 (2,28)
Gasch2	25,89 (2,24)	26,00 (2,24)	26,21 (1,34)	20,76 (3,11) ▲	22,52 (3,58)	23,39 (2,21)
Phenotype	25,77 (3,38)	22,95 (4,75)	25,45 (3,32)	22,86 (3,86) ▲	21,39 (3,94) ▲	26,37 (3,35)
Sequence	23,64 (2,41)	23,90 (2,58)	23,75 (2,65)	19,68 (2,31) ▲	22,86 (2,59) ▲	25,71 (2,87)
SPO	20,47 (2,08) ▲	22,29 (2,49)	22,31 (1,64)	19,86 (2,45)	19,36 (2,13) ▲	21,50 (2,58)
Total B	2	1	7	1	0	9

A Tabela 3.5 exibe a consolidação dos resultados das tabelas anteriores considerando-se o classificador com a estratégia SW, o classificador FL e o PPTD. Para cada base de dados, tem-se o melhor resultado de cada classificador hierárquico e o respectivo classificador plano aplicado que o encontrou.

Para as bases do Grupo A, observa-se que os resultados dos três classificadores são muito próximos. Porém, percebe-se que o classificador com a estratégia SW apresenta-se competitivo, pois encontra, no total, seis melhores resultados para as oito bases. Consi-

derando as bases do Grupo B, o classificador com a estratégia SW apresentou um desempenho superior ao dos demais classificadores em nove das 10 bases avaliadas.

Em vista desses resultados, conclui-se que o classificador hierárquico proposto apresenta um desempenho satisfatório, alcançando resultados iguais ou superiores em 15 das 18 bases utilizadas nos experimentos.

Tabela 3.5: Melhores resultados encontrados por estratégia hierárquica

Bases	FL		PPTD		SW	
	hF	Classificador Plano	hF	Classificador Plano	hF	Classificador Plano
GPCRpfam	70,31	1-NN	70,32	1-NN	70,31	1-NN
GPCRprints	82,98	1-NN	82,97	1-NN	83,00	1-NN
GPCRprosite	69,28	1-NN	69,25	1-NN	69,26	1-NN
GPCRinterpro	83,09	1-NN	83,09	1-NN	83,09	1-NN
ECpfam	98,77	1-NN	98,77	1-NN	98,77	1-NN
ECprints	98,19	1-NN	98,19	1-NN	98,19	1-NN
ECprosite	98,80	1-NN	98,80	1-NN	98,80	1-NN
ECinterpro	99,08	1-NN	99,07	1-NN	99,08	1-NN
Total A		6		5		6
Church	22,10	9-NN	21,53	C4.5	21,29	C4.5
CellCycle	29,08	9-NN	29,40	9-NN	29,87	9-NN
Derisi	20,17	9-NN	20,89	9-NN	21,78	C4.5
Eisen	28,73	9-NN	29,25	7-NN	30,77	9-NN
Expr	27,58	9-NN	28,29	9-NN	29,26	9-NN
Gasch1	30,40	9-NN	30,97	7-NN	31,24	7-NN
Gasch2	25,89	9-NN	26,12	7-NN	26,21	9-NN
Phenotype	25,77	9-NN	22,95	9-NN	26,37	C4.5
Sequence	23,64	9-NN	24,02	1-NN	25,71	C4.5
SPO	20,47	9-NN	22,29	9-NN	22,31	9-NN
Total B		1		0		9

A Tabela 3.6 apresenta os melhores resultados obtidos para cada uma das bases de dados e as estratégias que os atingiram. Observa-se novamente um equilíbrio entre as estratégias para as bases do Grupo A e um predomínio da estratégia SW para as bases do Grupo B.

Tabela 3.6: Melhores resultados encontrados por base de dados

Grupo	Bases	hF	Estratégia(s)
A	GPCRpfam	70,32	PPTD/1-NN
	GPCRprints	83,00	SW/1-NN
	GPCRprosite	69,28	FL/1-NN
	GPCRinterpro	83,09	FL/1-NN, PPTD/1-NN e SW/1-NN
	ECpfam	98,77	FL/1-NN, PPTD/1-NN e SW/1-NN
	ECprints	98,19	FL/1-NN, PPTD/1-NN e SW/1-NN
	ECprosite	98,80	FL/1-NN, PPTD/1-NN e SW/1-NN
	ECinterpro	99,08	FL/1-NN e SW/1-NN
B	Church	22,10	FL/9-NN
	CellCycle	29,87	SW/9-NN
	Derisi	21,78	SW/C4.5
	Eisen	30,77	SW/9-NN
	Expr	29,26	SW/9-NN
	Gasch1	31,24	SW/7-NN
	Gasch2	26,21	SW/9-NN
	Phenotype	26,37	SW/C4.5
	Sequence	25,71	SW/C4.5
	SPO	22,31	SW/9-NN

Os resultados apresentados neste capítulo estão publicados em [25].

Capítulo 4

Seleção de Atributos Aplicada à Classificação Hierárquica

Neste capítulo, propõe-se a aplicação de métodos de seleção de atributos nos classificadores hierárquicos explorados neste trabalho com o objetivo de melhorar as respectivas acurácias preditivas.

A Seção 4.1 introduz o processo de seleção de atributos. A Seção 4.2 descreve a aplicação de um método de seleção de atributos tradicional nos classificadores hierárquicos apresentados no Capítulo 3, além de realizar uma análise dos resultados computacionais obtidos. A Seção 4.3 apresenta a utilização de um método de seleção de atributos *lazy* nos classificadores hierárquicos explorados e avalia os resultados computacionais obtidos.

4.1 Introdução à Seleção de Atributos

Seleção de atributos [14] é uma técnica muito explorada na área de mineração de dados, principalmente na tarefa de classificação. Nesse contexto, seu objetivo é identificar atributos relevantes, visando obter um ou mais dentre os seguintes benefícios:

- Redução do tempo de execução do processo de classificação – Com menos atributos avaliados, o processo de classificação tende a ser executado em um menor tempo de processamento.
- Aumento da capacidade preditiva do classificador – A seleção de atributos procura retirar atributos redundantes ou irrelevantes da base de dados. Consequentemente, permite a geração de um classificador menos propenso a erros.

- Obtenção de uma representação mais compacta do conceito a ser aprendido – O conhecimento ficará concentrado somente nos atributos que realmente são importantes para a tarefa a ser executada.

Em linhas gerais, os métodos de seleção de atributos podem ser categorizados em três grandes tipos:

- Métodos *Wrapper*

Avaliam a qualidade dos subconjuntos de atributos utilizando o próprio algoritmo de classificação adotado. Normalmente, possuem boa capacidade preditiva pois avaliam cada subconjunto de atributos usando o mesmo algoritmo de classificação que será utilizado no processo de classificação. No entanto, requerem várias execuções do algoritmo de classificação, o que eleva o custo computacional em relação aos outros métodos.

- Métodos *Filter*

São métodos independentes do algoritmo de classificação que será aplicado. Utilizam medidas específicas para avaliar a qualidade dos atributos disponíveis. Esses métodos podem avaliar cada atributo independente dos outros, determinando o grau de correlação entre cada atributo e a classe [37], ou podem avaliar subconjuntos de atributos, buscando através de estratégias e heurísticas, aqueles que, em conjunto, melhor identificam as classes [15][24]. Neste trabalho, serão utilizados métodos do tipo *Filter* em conjunto com os classificadores hierárquicos.

- Métodos *Embedded*

São métodos incorporados ao algoritmo de classificação. São aplicados internamente e de forma integrada ao algoritmo de classificação. Algoritmos de indução de árvores de decisão são exemplos típicos pois internamente selecionam os atributos que formarão os rótulos da árvore gerada.

Alguns exemplos de utilização de seleção de atributos podem ser encontrados na área de classificação hierárquica. Em [22], implementa-se um classificador de documentos hierárquico *top-down* no qual a seleção de atributos é executada antes do treinamento do classificador para cada nó da hierarquia. Um único tipo de classificador plano é utilizado em todos os nós e o número de atributos a serem selecionados para geração do classificador de cada nó é um parâmetro definido pelo usuário. A seleção tenta identificar os atributos que melhor capturam a distribuição de classes nos dados associados a cada nó.

Na área da bioinformática, encontra-se outro trabalho importante. Em [28], foi proposto um classificador hierárquico *top-down* com seleção de atributos. Nesse trabalho, os nós da hierarquia poderiam estar associados a classificadores distintos. A hierarquia de classificadores é formada a partir de um método seletivo, que identifica o algoritmo de classificação mais adequado para cada nó. A seleção de atributos é usada para reduzir a dimensionalidade dos dados e melhorar a acurácia preditiva. O método de seleção de atributos *CfsSubsetEval* [15] é utilizado antes do treinamento do classificador em cada nó.

4.2 Aplicando Seleção de Atributos Tradicional

Uma das contribuições deste trabalho está na utilização de métodos de seleção de atributos em conjunto com classificadores hierárquicos, na tentativa de melhorar o desempenho dos classificadores explorados.

O método de seleção de atributos aplicado é considerado do tipo *Filter*. Basicamente, realiza um ranqueamento dos atributos com base na medida de ganho de informação [16], baseado no conceito de entropia. O método foi aplicado em uma etapa de pré-processamento, antes do treinamento dos classificadores planos que compõem os classificadores hierárquicos. Ele possui como parâmetro de entrada o número de atributos a serem selecionados.

Nas Figuras 4.1, 4.2 e 4.3, são apresentadas as propostas de aplicação do método de seleção de atributos nos classificadores hierárquicos apresentados no Capítulo 3: *Flat on Leaves* (FL), *Per Parent Top-Down* (PPTD) e *Sum of Weighted Votes* (SW). Para ilustrar o seu funcionamento, considerou-se o conjunto original de atributos A_1, A_2, A_3, A_4 e A_5 , e que 60% dos atributos devem ser selecionados pelo método de seleção de atributos.

A Figura 4.1 ilustra a aplicação da seleção de atributos no classificador FL. As classes sombreadas na hierarquia representam as classes folha que fazem parte do treinamento. Antes do treinamento do único classificador C , a seleção de atributos é realizada. A figura ilustra que os atributos A_3, A_4 e A_5 são selecionados. A partir desses atributos, o treinamento do classificador é realizado.

A Figura 4.2 ilustra a aplicação da seleção de atributos no classificador hierárquico PPTD. Para cada nó pai da hierarquia, um conjunto de atributos é selecionado antes da realização do treinamento do classificador. Dessa forma, diferentes subconjuntos de atributos são selecionados em cada nó. Todos os atributos estão disponíveis em cada nó para a realização da seleção de atributos, ou seja, atributos não são perdidos pelos nós

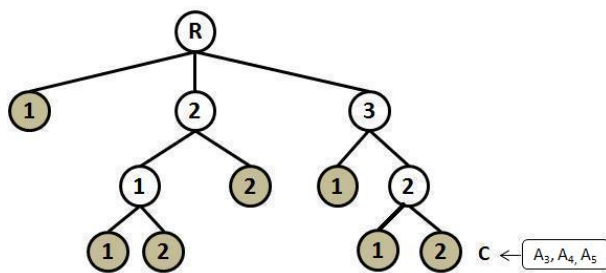


Figura 4.1: Seleção de atributos no classificador FL

filhos quando não selecionados pelo nó pai. A figura apresenta o resultado da seleção de atributos aplicada a cada nó pai. A partir desses diferentes subgrupos de atributos, realizam-se os treinamentos dos classificadores C_1, C_2, C_3, C_4 e C_5 , associados a cada nó pai. Por exemplo, o classificador C_4 associado ao nó representado pela classe 2.1 é treinado com os atributos A_1, A_4 e A_5 .

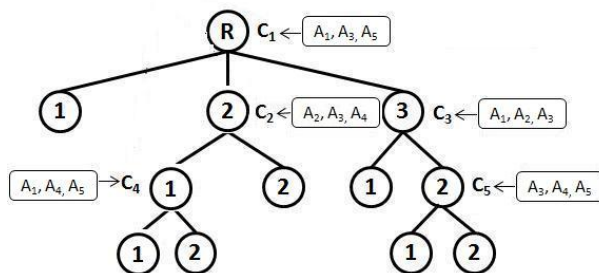


Figura 4.2: Seleção de atributos no classificador PPTD

A Figura 4.3 ilustra a aplicação da seleção de atributos no classificador com a estratégia SW. Para cada nível da hierarquia, é selecionado um subconjunto de atributos antes do treinamento do respectivo classificador plano. A figura ilustra os atributos selecionados em cada nível da hierarquia e, a partir desses, o treinamento dos classificadores C_1, C_2 e C_3 , dos diferentes níveis é realizado. Por exemplo, o classificador C_2 , associado ao nível 2 da hierarquia, é treinado com os atributos A_2, A_3 e A_5

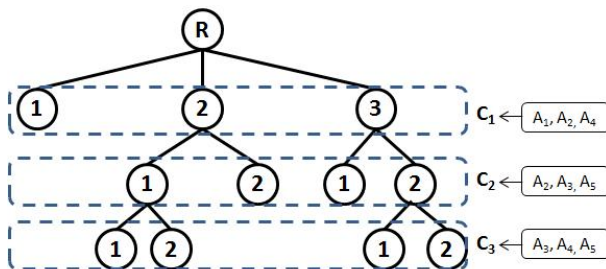


Figura 4.3: Seleção de atributos no classificador com a estratégia SW

4.2.1 Descrição dos Experimentos

Para avaliação do desempenho dos classificadores hierárquicos com a aplicação de seleção de atributos, foram utilizadas as mesmas 18 bases de dados descritas no Capítulo 3.

O método de seleção de atributos *Filter*, disponível na ferramenta WEKA com o nome *InfoGainAttributeEval*, foi aplicado nos classificadores hierárquicos implementados. A escolha baseou-se em sua simplicidade e por ser um método amplamente conhecido. Cabe ressaltar que esse método tem como parâmetro de entrada o número de atributos a serem selecionados.

Os classificadores hierárquicos avaliados no capítulo anterior utilizaram como base os classificadores planos tradicionais: 1-NN, 3-NN, 5-NN, 7-NN, 9-NN e C4.5. Como foi observado nos resultados computacionais, os classificadores planos 3-NN e 5-NN não apresentaram resultados expressivos, ilustrado na Tabela 3.5. Em consequência disso, os classificadores planos 3-NN e 5-NN não farão parte dos experimentos computacionais deste capítulo. Além disso, também constatou-se que o classificador hierárquico com a estratégia SW obteve melhor desempenho quando comparado ao outro classificador SV proposto. Dessa forma, somente os classificadores hierárquicos SW, FL e PPTD farão parte dos experimentos computacionais seguintes.

Novamente, a avaliação dos classificadores será efetuada utilizando-se validação cruzada com 10 partições e foi adotada a medida hierárquica hF, apresentada no Capítulo 2. Para avaliar a significância estatística na comparação entre dois classificadores hierárquicos, utilizou-se o teste-t de *Student* [20], bicaudal e pareado (com nível de confiança de 95%, ou seja, p-valor de 5%).

4.2.2 Resultados Computacionais

Nesta seção, são apresentados os resultados e análises dos experimentos computacionais. O objetivo é avaliar o impacto da seleção de atributos aplicada aos classificadores hierárquicos tradicionais (*Flat On Leaves* e *Per Parent Top-Down*) e ao classificador hierárquico com a estratégia proposta (*Sum of Weighted Votes*).

Nas Tabelas 4.1, 4.2 e 4.3, são apresentadas, respectivamente, as comparações entre os classificadores hierárquicos FL, PPTD e SW, com e sem a aplicação de seleção de atributos. Foram utilizados cada um dos quatro classificadores planos que obtiveram melhores desempenhos nos experimentos do Capítulo 3: 1-NN, 7-NN, 9-NN e C4.5. Para

cada combinação de base de dados e classificador plano adotado, são apresentados os valores da medida hF do classificador hierárquico com seleção de atributos (Sel.) e sem seleção de atributos (Sem Sel.). Ao lado do hF dos classificadores com seleção de atributos, tem-se a porcentagem de atributos (10%, 20%, ..., 80% ou 90%) que levou o classificador a atingir o melhor resultado. Caso dois ou mais valores percentuais tenham originado o melhor resultado, o menor valor percentual será reportado. Em negrito estão marcados os melhores resultados para cada classificador plano aplicado e os melhores resultados por base estão sublinhados. O símbolo (●) entre os dois valores de hF indica que a diferença entre esses valores possui significância estatística. O símbolo (-) indica que não houve significância estatística. Por fim, abaixo de cada grupo de bases de dados, tem-se uma linha de totais, que apresenta o número de vezes que um dos classificadores hierárquicos apresentou o valor hF maior ou igual ao valor hF do outro, para cada classificador plano adotado.

A Tabela 4.1 apresenta os resultados do classificador hierárquico *Flat On Leaves* com e sem seleção de atributos. Analisando as linhas de totais, observa-se que nas bases do Grupo A, o classificador FL com seleção de atributos obteve sempre um número maior ou igual de melhores resultados que o classificador sem seleção de atributos, para todos os classificadores planos adotados. Analisando as bases do Grupo B, observa-se, através das linhas de totais, um equilíbrio nos resultados encontrados quando foram adotados os classificadores planos 1-NN, 7-NN e 9-NN, e um melhor desempenho do classificador FL com seleção quando aplicado o classificador plano C4.5. Cabe observar que, dos 10 resultados com significância estatística obtidos, o classificador FL com seleção de atributos obteve nove e o classificador FL sem seleção de atributos apenas um. Considerando os melhores resultados encontrados por base de dados (sublinhados), o classificador FL com seleção de atributos encontrou 12, enquanto o classificador FL sem seleção de atributos obteve 12 desses resultados.

A Tabela 4.2 apresenta os resultados do classificador hierárquico *Per Parent Top-Down* com e sem seleção de atributos. Observa-se, através das linhas de totais, que, para as bases de dados de ambos os grupos, o classificador hierárquico PPTD obteve maiores valores de hF quando aplicado a seleção de atributos para todos os classificadores planos utilizados. Dos 12 resultados com significância estatística obtidos, o classificador PPTD com seleção de atributos obteve 11 e o classificador PPTD sem seleção de atributos apenas um. Considerando os melhores resultados encontrados por base de dados (sublinhados), o classificador PPTD com seleção de atributos encontrou 18, enquanto o classificador PPTD sem seleção de atributos obteve seis desses resultados.

Tabela 4.1: hF do Classificador FL com e sem seleção de atributos

Bases	1-NN		7-NN		9-NN		C4.5	
	Sel.	Sem Sel.	Sel.	Sem Sel.	Sel.	Sem Sel.	Sel.	Sem Sel.
GPCRpfam	70,31 (90) - 70,31		69,46 (50) - 69,33		68,79 (50) ● 68,62		69,32 (50) - 69,32	
GPCRprints	82,98 (80) - 82,98		81,28 (70) - 81,28		80,68 (70) ● 80,54		81,68 (80) - 81,76	
GPCRprosite	69,35 (50) - 69,28		67,94 (50) ● 67,66		67,02 (30) ● 66,67		68,04 (70) - 68,03	
GPCRinterpro	83,09 (80) - 83,09		82,19 (60) - 82,03		82,05 (60) - 81,99		83,08 (70) - 82,66	
ECpfam	98,77 (70) - 98,77		98,27 (70) - 98,27		98,18 (70) - 98,18		98,35 (70) - 98,33	
ECprints	98,19 (80) - 98,19		97,49 (80) - 97,49		97,27 (80) - 97,27		97,56 (80) - 97,56	
ECprosite	98,80 (70) - 98,80		98,37 (80) - 98,37		98,20 (80) - 98,16		98,50 (70) - 98,50	
ECinterpro	99,07 (70) - 99,08		98,73 (70) - 98,73		98,72 (70) - 98,72		98,73 (70) - 98,74	
Total A	7	7	8	5	8	3	6	5
Church	25,23 (10) ● 21,26		24,55 (10) ● 21,72		24,77 (10) ● 22,10		24,72 (10) ● 21,74	
CellCycle	24,02 (70) - 24,83		27,53 (90) - 27,47		27,88 (80) - 29,08		22,47 (90) - 21,86	
Derise	19,23 (10) - 18,87		19,97 (10) - 18,88		19,89 (90) - 20,17		18,44 (40) - 18,28	
Eisen	24,62 (90) - 24,72		27,88 (60) - 28,28		29,03 (70) - 28,73		23,05 (90) - 22,96	
Expr	24,19 (70) - 25,05		25,58 (50) - 25,92		27,03 (90) - 27,58		24,74 (90) - 24,98	
Gash1	28,43 (80) - 28,84		28,88 (90) - 29,16		25,32 (40) ● 30,40		23,02 (90) - 23,09	
Gash2	24,14 (90) - 24,58		24,72 (30) - 24,37		25,32 (40) - 25,89		21,92 (40) - 20,76	
Phenotype	25,40 (10) - 20,67		25,23 (10) - 24,57		25,77 (80) - 25,77		25,99 (10) ● 22,86	
Sequence	23,71 (70) - 22,43		23,42 (80) - 23,59		24,05 (70) - 23,64		21,65 (80) - 19,68	
SPO	19,01 (90) - 18,23		19,56 (90) - 19,14		21,10 (90) - 20,47		19,62 (90) - 19,86	
Total B	5	5	6	4	5	6	7	3

Tabela 4.2: hF do Classificador PPTD com e sem seleção de atributos

Bases	1-NN		7-NN		9-NN		C4.5	
	Sel.	Sem Sel.	Sel.	Sem Sel.	Sel.	Sem Sel.	Sel.	Sem Sel.
GPCRpfam	70,32 (90) - 70,32		69,09 (70) - 69,04		68,55 (70) - 68,47		68,85 (70) - 68,84	
GPCRprints	82,97 (80) - 82,97		80,95 (70) - 80,89		80,41 (80) - 80,41		79,22 (50) - 79,19	
GPCRprosite	69,26 (70) - 69,25		67,38 (70) ● 67,31		66,57 (90) - 66,57		67,67 (70) - 67,63	
GPCRinterpro	83,09 (90) - 83,09		81,96 (80) - 81,95		81,29 (80) - 81,29		81,52 (80) - 81,54	
ECpfam	98,77 (70) - 98,77		98,16 (70) - 98,16		97,88 (70) - 97,88		98,40 (60) - 98,39	
ECprints	98,19 (80) - 98,19		97,37 (80) - 97,37		97,05 (80) - 97,05		97,34 (80) - 97,35	
ECprosite	98,81 (70) - 98,80		98,29 (70) - 98,29		98,03 (80) - 98,03		98,46 (70) - 98,46	
ECinterpro	99,07 (30) - 99,07		98,62 (70) - 98,62		98,32 (70) - 98,32		98,68 (70) ● 98,73	
Total A	8	6	8	4	8	7	5	4
Church	21,64 (10) ● 19,38		23,06 (10) ● 19,66		23,07 (10) ● 19,82		25,29 (10) ● 21,53	
CellCycle	24,75 (40) - 24,56		29,29 (50) - 28,38		29,91 (40) - 29,40		22,93 (80) - 22,19	
Derise	20,09 (70) - 18,89		22,07 (70) - 20,21		22,71 (30) - 20,89		22,82 (10) - 20,47	
Eisen	25,66 (50) - 24,70		29,53 (70) - 29,25		30,39 (50) - 29,16		26,77 (50) - 24,24	
Expr	26,05 (10) - 25,35		29,33 (70) - 27,21		29,78 (70) - 28,29		26,26 (30) - 24,74	
Gash1	27,81 (70) - 28,29		31,37 (80) - 30,97		27,92 (70) - 30,08		24,34 (90) ● 22,90	
Gash2	24,96 (80) - 25,23		27,89 (80) - 26,12		27,92 (70) ● 26,00		23,39 (40) - 22,52	
Phenotype	21,34 (20) - 20,27		23,23 (20) - 22,52		22,95 (90) - 22,95		21,57 (60) - 21,39	
Sequence	25,01 (90) - 24,02		24,88 (80) - 23,73		25,63 (30) ● 23,90		25,26 (10) - 22,86	
SPO	21,70 (20) - 18,86		24,55 (30) ● 21,10		26,05 (30) ● 22,29		23,83 (10) ● 19,36	
Total B	8	2	10	0	9	2	10	0

A Tabela 4.3 apresenta os resultados do classificador hierárquico com a estratégia *Sum of Weighted Votes* com e sem seleção de atributos. Analisando esses resultados, observa-se, para as bases dos dois grupos, através das linhas de totais, o desempenho superior da estratégia hierárquica SW com a aplicação da seleção de atributos, para todos os classificadores planos utilizados. Todos os 20 resultados com significância estatística foram obtidos com a estratégia SW com seleção de atributos. Considerando os melhores resultados encontrados por base de dados (sublinhados), a estratégia SW com seleção de atributos encontrou 18, enquanto a estratégia SW sem seleção de atributos obteve seis desses resultados.

Tabela 4.3: hF da Estratégia SW com e sem seleção de atributos

Bases	1-NN		7-NN		9-NN		C4.5	
	Sel.	Sem Sel.	Sel.	Sem Sel.	Sel.	Sem Sel.	Sel.	Sem Sel.
GPCRpfam	<u>70,31</u> (90) - <u>70,31</u>		68,78 (50) - 68,68		<u>68,25</u> (40) ● 68,07		<u>68,76</u> (50) - 68,70	
GPCRprints	<u>83,00</u> (80) - <u>83,00</u>		80,97 (70) - 80,86		80,29 (70) - 80,28		<u>79,53</u> (50) - 79,33	
GPCRprosite	<u>69,36</u> (40) - <u>69,26</u>		67,13 (40) ● 66,83		<u>66,14</u> (30) - 66,00		<u>67,13</u> (20) - 67,08	
GPCRinterpro	<u>83,09</u> (90) - <u>83,09</u>		81,69 (80) - 81,68		<u>81,31</u> (80) - <u>81,31</u>		<u>81,98</u> (90) - 81,80	
ECpfam	<u>98,77</u> (70) - <u>98,77</u>		98,30 (70) - <u>98,30</u>		<u>98,19</u> (70) - <u>98,19</u>		<u>98,43</u> (60) - <u>98,43</u>	
ECprints	<u>98,19</u> (80) - <u>98,19</u>		97,45 (80) - <u>97,45</u>		<u>97,22</u> (70) - <u>97,22</u>		<u>97,54</u> (80) ● 97,51	
ECprosite	<u>98,81</u> (70) - <u>98,80</u>		98,35 (60) - 98,33		<u>98,10</u> (70) - 98,08		<u>98,57</u> (70) - <u>98,57</u>	
ECinterpro	<u>99,08</u> (70) - <u>99,08</u>		98,82 (70) - <u>98,82</u>		<u>98,70</u> (70) - <u>98,70</u>		98,78 (70) - <u>98,79</u>	
Total A	8	6	8	3	8	4	7	3
Church	<u>21,88</u> (10) ● 19,70		<u>22,96</u> (10) ● 20,19		<u>23,22</u> (10) ● 20,66		<u>25,58</u> (10) ● 21,29	
CellCycle	<u>25,12</u> (40) - 24,82		<u>30,16</u> (40) - 28,60		<u>30,96</u> (40) - 29,87		<u>24,86</u> (40) - 24,83	
Derise	<u>20,26</u> (60) ● 18,73		<u>23,22</u> (10) - 21,16		<u>23,52</u> (40) ● 21,20		<u>22,44</u> (10) - 21,78	
Eisen	<u>27,26</u> (50) ● 24,54		<u>32,43</u> (50) ● 29,56		<u>32,51</u> (50) - 30,77		<u>28,20</u> (70) ● 25,79	
Expr	<u>26,18</u> (30) - 25,62		<u>29,85</u> (50) ● 28,12		<u>30,42</u> (40) - 29,26		<u>28,92</u> (60) - 26,61	
Gash1	<u>29,60</u> (90) - 28,98		<u>31,67</u> (80) - 31,24		<u>28,93</u> (40) - <u>30,80</u>		<u>27,30</u> (90) - 25,86	
Gash2	<u>26,46</u> (50) - 25,03		<u>28,39</u> (50) ● 25,55		<u>28,93</u> (40) ● 26,21		<u>24,62</u> (30) - 23,39	
Phenotype	<u>26,25</u> (10) ● 22,46		<u>26,69</u> (10) - 24,42		<u>26,46</u> (10) - 25,45		<u>27,94</u> (10) - 26,37	
Sequence	<u>23,99</u> (70) - 23,08		<u>24,19</u> (40) - 22,76		<u>25,48</u> (40) - 23,75		<u>26,18</u> (70) - 25,71	
SPO	<u>21,95</u> (20) ● 18,50		<u>24,98</u> (30) ● 21,59		<u>25,64</u> (20) ● 22,31		<u>23,38</u> (40) ● 21,50	
Total B	10	0	10	0	9	1	10	0

Finalmente, na Tabela 4.4, avalia-se a influência da seleção de atributos em cada um dos classificadores hierárquicos implementados. Consolidam-se, para cada base de dados, os melhores resultados encontrados por cada um dos classificadores hierárquicos (FL, PPTD e SW) com a aplicação de seleção de atributos, apresentados nas tabelas anteriores. Para cada combinação de base de dados e classificador hierárquico analisado, são apresentados os valores da medida hF e, ao lado, têm-se o classificador plano adotado e a porcentagem de atributos que foram selecionados. Nessa análise, considerou-se que 100% dos atributos utilizados representa o valor encontrado sem seleção de atributos. Exemplificando, para a base *Church* e classificador hierárquico FL, o hF obtido foi de 25,23 com seleção de atributos de 10% utilizando o classificador plano 1-NN. Para a base *Expr*, o classificador hierárquico FL obteve hF de 27,58 sem seleção de atributos (100% dos atributos utilizados) com o classificador plano 9-NN .

Observando a Tabela 4.4, conclui-se que os classificadores hierárquicos FL, PPTD e

Tabela 4.4: Influência da seleção de atributos nos classificadores FL, PPTD e SW

Bases	FL		PPTD		SW	
	hF	Classificador Plano	hF	Classificador Plano	hF	Classificador Plano
GPCRpfam	70,31	1-NN(90)	70,32	1-NN(90)	70,30	1-NN(90)
GPCRprints	82,98	1-NN(80)	82,97	1-NN(80)	83,00	1-NN(80)
GPCRprosite	69,35	1-NN(50)	69,26	1-NN(70)	69,36	1-NN(40)
GPCRinterpro	83,09	1-NN(80)	83,09	1-NN(90)	83,09	1-NN(90)
ECpfam	98,77	1-NN(70)	98,77	1-NN(70)	98,77	1-NN(70)
ECprints	98,19	1-NN(80)	98,19	1-NN(80)	98,19	1-NN(80)
ECprosite	98,80	1-NN(70)	98,81	1-NN(70)	98,81	1-NN(70)
ECinterpro	99,08	1-NN(100)	99,07	1-NN(30)	99,08	1-NN(70)
Total A		4		5		7
Church	25,23	1-NN(10)	25,29	C4.5(10)	25,58	C4.5(10)
CellCycle	29,08	9-NN(100)	29,91	9-NN(40)	30,96	9-NN(40)
Derise	20,17	9-NN(100)	22,82	C4.5(10)	23,52	9-NN(40)
Eisen	29,03	9-NN(70)	30,39	9-NN(50)	32,51	9-NN(50)
Expr	27,58	9-NN(100)	29,78	9-NN(70)	30,42	9-NN(40)
Gash1	30,40	9-NN(100)	31,37	7-NN(80)	31,67	7-NN(80)
Gash2	25,89	9-NN(100)	27,92	9-NN(70)	28,93	9-NN(40)
Phenotype	25,99	C4.5(10)	23,71	5-NN(20)	27,94	C4.5(10)
Sequence	24,05	9-NN(70)	25,63	9-NN(30)	26,18	C4.5(70)
SPO	21,10	9-NN(90)	26,05	9-NN(30)	25,64	9-NN(20)
Total B		0		1		9

SW obtiveram um desempenho melhor quando foi aplicado o método de seleção de atributos pois, em poucos casos – apenas para o classificador FL – todos os atributos (100%) foram utilizados. Além disso, através das linhas de totais, observa-se que o classificador com a estratégia SW continua, com seleção de atributos, obtendo melhor desempenho em comparação com os classificadores hierárquicos FL e PPTD. Para as oito bases do Grupo A, a estratégia SW atingiu os melhores resultados em sete bases e, para as 10 bases do Grupo B, a estratégia SW atingiu os melhores valores em nove bases.

A Tabela 4.5 apresenta, para cada base de dados, os melhores resultados obtidos e as estratégias que os atingiram. A estratégia está representada pelo classificador hierárquico aplicado, classificador plano utilizado e a porcentagem de atributos que foram selecionados. Observa-se, que para todas as 18 bases de dados, o melhor resultado foi obtido por uma estratégia hierárquica com seleção de atributos. Apenas para a base ECinterpro, a estratégia FL/1-NN sem seleção de atributos também atingiu o melhor resultado.

Observa-se novamente que, para as oito bases do Grupo A, com a aplicação da seleção de atributos, houve uma pequena superioridade de desempenho da estratégia proposta SW, que atingiu sete melhores resultados, enquanto as estratégias FL e PPTD obtiveram quatro e cinco vezes o melhor valor de hF. Para as bases do Grupo B, há uma clara superioridade de desempenho da estratégia proposta SW.

Tabela 4.5: Melhores resultados encontrados por base de dados

Grupo	Bases	hF	Estratégia(s)
A	GPCRpfam	70,32	PPTD/1-NN(90)
	GPCRprints	83,00	SW/1-NN(80)
	GPCRprosite	69,36	SW/1-NN(40)
	GPCRinterpro	83,09	FL/1-NN(80), PPTD/1-NN(90) e SW/1-NN(90)
	ECpfam	98,77	FL/1-NN(70), PPTD/1-NN(70) e SW/1-NN(70)
	ECprints	98,19	FL/1-NN(80), PPTD/1-NN(80) e SW/1-NN(80)
	ECprosite	98,81	PPTD/1-NN(70) e SW/1-NN(70)
	ECinterpro	99,08	FL/1-NN(100) e SW/1-NN(70)
B	Church	25,58	SW/C4.5(10)
	CellCycle	30,96	SW/9-NN(40)
	Derisi	23,52	SW/9-NN(40)
	Eisen	32,51	SW/9-NN(50)
	Expr	30,42	SW/9-NN(40)
	Gasch1	31,67	SW/7-NN(80)
	Gasch2	28,93	SW/9-NN(40)
	Phenotype	27,94	SW/C4.5(10)
	Sequence	26,18	SW/C4.5(70)
	SPO	26,05	PPTD/9-NN(30)

4.3 Aplicando Seleção de Atributos *Lazy*

Nesta seção, é reportada outra contribuição deste trabalho: a utilização, no contexto de classificação hierárquica, de um paradigma proposto recentemente para realizar seleção de atributos, denominada seleção de atributos *lazy* [26].

4.3.1 Introdução à Seleção de Atributos *Lazy*

Tradicionalmente, os métodos de seleção de atributos são aplicados em uma etapa de pré-processamento, como descrito na seção anterior. Dessa forma, os atributos não escolhidos pelo método de seleção não são considerados no treinamento e na utilização do classificador.

Em [26], foi proposto um novo método de seleção de atributos cujo principal objetivo é adiar a realização da seleção de atributos para o momento da classificação de uma nova instância. A ideia básica é considerar os valores dos atributos da instância a ser classificada para, assim, escolher os atributos que farão parte da classificação. Dessa forma, os atributos selecionados serão específicos para cada instância e, conseqüentemente, poderão aumentar a capacidade preditiva do classificador.

Com base nas mesmas ideias utilizadas para categorizar os classificadores nos tipos *eager* e *lazy*, serão categorizados os métodos de seleção de atributos. Os métodos tradicionais de seleção de atributos serão denominados do tipo *eager*, pois a seleção de atributos é realizada uma única vez, anteriormente a fase de classificação das instâncias. O novo método de seleção de atributos [26], será considerado *lazy*, pois adia a seleção de atributos até o momento da classificação de novas instâncias.

O método de seleção de atributos *lazy* proposto em [26] baseia-se na hipótese de que a seleção de atributos pode ser mais eficiente se considerar os valores dos atributos da instância analisada no momento da classificação. Dessa forma, é possível selecionar subconjuntos distintos de atributos para cada instância considerando seus valores específicos.

Para ilustrar a motivação dessa proposta, a Tabela 4.6, retirada de [26], apresenta, de duas formas diferentes, um conjunto de instâncias que possuem três atributos: X , Y e C , sendo esse último o atributo classe. Na representação da esquerda, as instâncias estão ordenadas pelos valores do atributo X e, na representação da direita, as instâncias estão ordenadas pelos valores do atributo Y . Caso seja necessária a escolha de um dos atributos, com a aplicação da seleção de atributos tradicional *eager*, o atributo X seria o escolhido, pois seus valores conseguem distinguir bem os valores da classe: o atributo X com o valor 1 determina a classe B, com valor 2 determina a classe A e com o valor 3 determina a classe B. Somente o valor 4 não indica um único valor de classe. O atributo Y seria descartado, pois os valores têm menor capacidade de identificar as classes. Porém, nesse caso, o fato de o atributo Y com o valor 4 determinar a classe A seria perdido da base. No caso da aplicação da seleção de atributos *lazy*, o atributo Y poderia ser escolhido quando a instância a ser classificada tivesse o valor 4 nesse atributo.

Tabela 4.6: Base de dados exemplo

Ordenação por X			Ordenação por Y		
X	Y	C	X	Y	C
1	2	B	2	1	A
1	3	B	3	1	B
1	4	B	4	1	A
2	1	A	1	2	B
2	2	A	2	2	A
2	3	A	3	2	B
3	1	B	1	3	B
3	2	B	2	3	A
3	4	B	4	3	B
4	1	A	1	4	B
4	3	B	3	4	B
4	4	B	4	4	B

Esse método de seleção *lazy*, proposto em [26], e adotado neste trabalho, utiliza uma adaptação do conceito de entropia para avaliar, individualmente, cada valor de atributo da instância a ser classificada no que diz respeito a qualidade de discriminar as classes. Essa estratégia também é do tipo *Filter* e, além disso, recebe como parâmetro de entrada o número de atributos a serem selecionados.

4.3.2 Aplicando Seleção de Atributos *Lazy*

Os classificadores hierárquicos apresentados neste trabalho foram definidos a partir dos seguintes classificadores planos tradicionais: o classificador *lazy* k-NN e o classificador *eager* C4.5. Classificadores *eager* constroem o modelo previamente antes da classificação de instâncias, exigindo que os atributos também sejam selecionados com antecedência. Por outro lado, os classificadores *lazy* deixam para realizar o treinamento no momento em que a instância será classificada. Nesse caso, os atributos poderão ser escolhidos antes ou no momento da classificação das instâncias. Dessa forma, o método de seleção de atributos *lazy* pode ser aplicado em classificadores hierárquicos compostos por classificadores planos *lazy*. Neste trabalho, na Seção 4.2, um método de seleção de atributos *eager* foi utilizado em conjunto com os classificadores hierárquicos FL, PPTD e SW compostos por classificadores *lazy* (k-NN) e *eager* (C4.5). Nesta seção, o método de seleção de atributos *lazy* será utilizado em conjunto com os classificadores hierárquicos FL, PPTD e SW compostos por um classificador *lazy* (k-NN).

Nas Figuras 4.4, 4.5 e 4.6, são apresentadas as propostas de aplicação do método de seleção de atributos *lazy* nos classificadores hierárquicos apresentados no Capítulo 3: *Flat on Leaves* (FL), *Per Parent Top-Down* (PPTD) e *Sum of Weighted Votes* (SW). Para ilustrar o seu funcionamento, considerou-se o conjunto original de atributos A_1, A_2, A_3, A_4 e A_5 , e que 60% dos atributos devem ser selecionados pelo método de seleção de atributos.

A Figura 4.4 ilustra a aplicação da seleção de atributos *lazy* no classificador FL. As classes sombreadas na hierarquia representam as classes folhas que fazem parte do treinamento do classificador C . Na Figura 4.4 (a), no momento da classificação da Instância 1, o método de seleção *lazy* é aplicado, considerando os valores dos seus atributos, selecionando para essa instância específica os atributos A_3, A_4 e A_5 . Na Figura 4.4 (b), no momento da classificação de uma outra instância, o método de seleção *lazy*, baseado nos valores dos atributos dessa instância, seleciona um novo subconjunto de atributos: A_1, A_2 e A_4 .

A Figura 4.5 ilustra a aplicação da seleção de atributos *lazy* no classificador hierárquico PPTD. Na Figura 4.5 (a), para cada nó pai da hierarquia, um conjunto de atributos é selecionado no momento da classificação da Instância 1, considerando os valores de seus atributos. Inicialmente, selecionam-se os atributos A_1, A_2 e A_4 para o classificador C_1 . Em seguida, o classificador C_1 prevê a classe 2. Novamente, considerando os valores dos atributos da Instância 1, selecionam-se os atributos A_2, A_3 e A_5 para o classificador C_2 associado à classe 2. Seguindo procedimento semelhante, o classificador associado

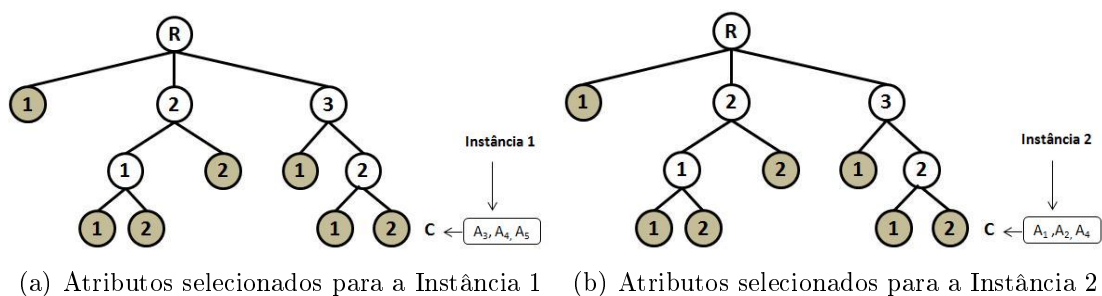


Figura 4.4: Seleção de atributos *lazy* no classificador FL

ao nó da classe 2 é executado, resultando a classe 2.1. Assim, os atributos A_1, A_4 e A_5 são selecionados e utilizados no classificador C_3 , associado a classe 2.1. Finalmente, o classificador C_3 prevê a classe 2.1.1. Da mesma forma, a Figura 4.5 (b) apresenta o resultado da seleção *lazy* de atributos, considerando os valores dos atributos da Instância 2, para cada nó pai da hierarquia.

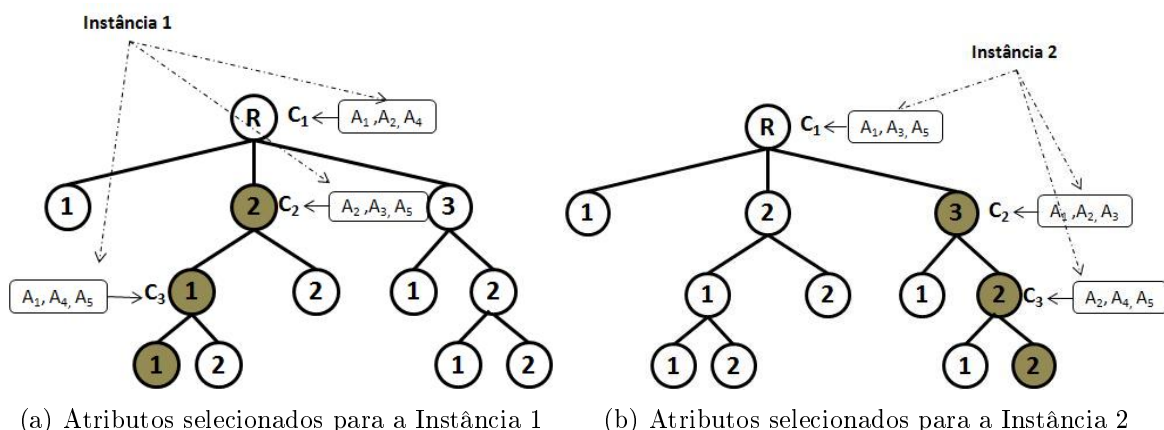


Figura 4.5: Seleção de atributos *lazy* no classificador PPTD

A Figura 4.6 ilustra a aplicação da seleção de atributos *lazy* no classificador hierárquico com a estratégia SW. Na Figura 4.6 (a), para cada nível da hierarquia, é realizada a seleção de atributos no momento em que a Instância 1 é submetida à classificação, selecionando subconjunto distintos de atributos para cada nível da hierarquia. A Figura 4.6 (b) ilustra que outros subconjuntos de atributos podem ser selecionados em cada nível da hierarquia, considerando agora os valores dos atributos de outra instância.

4.3.3 Resultados Computacionais

Nesta seção, são apresentados os resultados e análises dos experimentos computacionais que foram realizados para avaliar o impacto da seleção de atributos *lazy* aplicada aos classificadores hierárquicos tradicionais (*Flat on Leaves* e *Per Parent Top-Down*) e ao

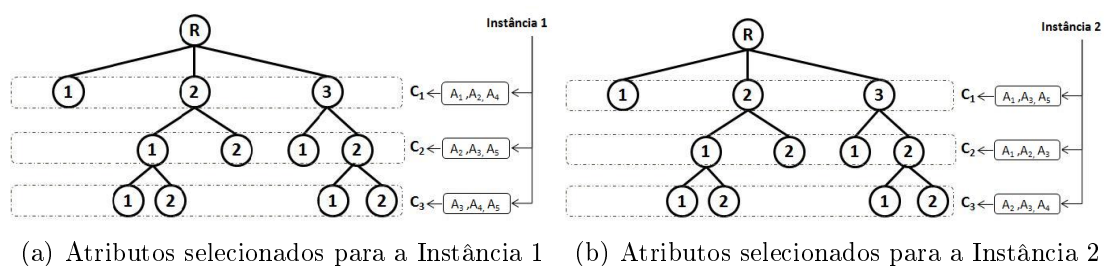


Figura 4.6: Seleção de atributos *lazy* no classificador com a estratégia SW

classificador hierárquico com a estratégia proposta (*Sum of Weighted Votes*). Os experimentos computacionais seguiram as mesmas condições apresentadas na Seção 4.2.1, porém é aplicado o método de seleção de atributos *lazy*.

Nas Tabelas 4.7, 4.8 e 4.9, são apresentadas, respectivamente, as comparações entre os classificadores hierárquicos FL, PPTD e SW, com a aplicação de seleção de atributos *eager* e *lazy*. Foram utilizados cada um dos quatro classificadores planos que obtiveram melhores desempenhos nos experimentos do Capítulo 3: 1-NN, 7-NN, 9-NN e C4.5. Para cada combinação de base de dados e classificador plano adotado, são apresentados os valores da medida hF do classificador hierárquico com seleção de atributos *eager* e com seleção de atributos *lazy*. Ao lado do hF dos classificadores hierárquicos com seleção de atributos, tem-se a porcentagem de atributos (10%, 20%, ..., 80%, 90% ou 100%) que levou o classificador a atingir o melhor resultado. Caso dois ou mais valores percentuais tenham originado o melhor resultado, o menor valor percentual será reportado. Observar que 100% dos atributos selecionados indica o valor encontrado sem seleção de atributos. Em negrito estão marcados os melhores resultados para cada classificador plano aplicado e os melhores resultados por base estão sublinhados. O símbolo (●) entre os dois valores de hF indica que a diferença entre esses valores possui significância estatística. O símbolo (-) indica que não houve significância estatística. Por fim, abaixo de cada grupo de bases de dados, tem-se uma linha de totais, que apresenta o número de vezes que um dos classificadores hierárquicos apresentou o valor hF maior ou igual ao valor hF do outro, para cada classificador plano adotado. É importante lembrar que quando adotado o classificador plano C4.5, somente é aplicada a seleção de atributos *eager*.

A Tabela 4.7 apresenta os resultados do classificador hierárquico *Flat on Leaves* com seleção de atributos *eager* e *lazy*. Analisando as linhas de totais, observa-se que, nas bases do Grupo A, o classificador FL com seleção de atributos *lazy* obteve um melhor desempenho que o classificador com seleção *eager*. Analisando as bases do Grupo B, observa-se, através das linhas de totais, um equilíbrio nos resultados encontrados quando foram ado-

tados os classificadores planos 1-NN, 7-NN e 9-NN. Considerando os melhores resultados encontrados por base de dados (sublinhados), o classificador FL com seleção *eager* encontrou sete, enquanto o classificador FL com seleção *lazy* obteve 13 desses resultados.

Tabela 4.7: hF do Classificador FL com seleção de atributos *Eager* e *Lazy*

Bases	1-NN		7-NN		9-NN		C4.5
	<u>Eager</u>	<u>Lazy</u>	<u>Eager</u>	<u>Lazy</u>	<u>Eager</u>	<u>Lazy</u>	<u>Eager</u>
GPCRpfam	70,31 (90)	- <u>70,32</u> (70)	<u>69,46</u> (50)	- 69,45 (50)	<u>68,79</u> (50)	- 68,77 (50)	69,32 (50)
GPCRprints	82,98 (80)	- <u>82,99</u> (70)	81,28 (70)	- <u>81,31</u> (70)	80,68 (70)	- <u>80,72</u> (60)	81,76 (100)
GPCRprosite	69,35 (50)	- <u>69,42</u> (30)	67,94 (50)	- <u>67,96</u> (30)	67,02 (30)	- <u>67,08</u> (30)	68,03 (70)
GPCRinterpro	83,09 (80)	- <u>83,12</u> (70)	82,19 (60)	- <u>82,26</u> (50)	82,05 (60)	- <u>82,18</u> (50)	83,08 (70)
ECpfam	98,77 (70)	- <u>98,80</u> (20)	<u>98,27</u> (70)	- <u>98,27</u> (70)	<u>98,18</u> (70)	- <u>98,18</u> (70)	98,35 (70)
ECprints	98,19 (80)	- <u>98,21</u> (20)	<u>97,49</u> (80)	- <u>97,49</u> (80)	<u>97,27</u> (80)	- <u>97,27</u> (80)	97,56 (80)
ECprosite	<u>98,80</u> (70)	- <u>98,80</u> (80)	<u>98,37</u> (80)	- <u>98,37</u> (70)	<u>98,20</u> (80)	- <u>98,20</u> (70)	98,50 (70)
ECinterpro	99,07 (70)	- <u>99,14</u> (10)	<u>98,73</u> (70)	- <u>98,73</u> (70)	<u>98,72</u> (70)	- <u>98,72</u> (70)	98,74 (100)
Total A	1	8	5	7	5	7	-
Church	<u>25,23</u> (10)	• 24,26 (10)	<u>24,55</u> (10)	- 24,30 (10)	<u>24,77</u> (10)	- 24,30 (10)	24,72 (10)
CellCycle	24,83 (100)	- <u>25,29</u> (90)	27,53 (90)	- <u>27,70</u> (90)	<u>29,08</u> (100)	- <u>29,08</u> (100)	22,47 (90)
Derise	<u>19,23</u> (10)	- 19,13 (90)	<u>19,97</u> (10)	- 19,08 (60)	20,17 (100)	- <u>20,41</u> (90)	18,44 (40)
Eisen	24,72 (100)	- <u>25,01</u> (60)	<u>28,28</u> (100)	- <u>28,28</u> (100)	<u>29,03</u> (70)	- 28,73 (100)	23,05 (90)
Expr	25,05 (100)	- <u>25,75</u> (80)	25,92 (100)	- <u>27,13</u> (80)	27,58 (100)	- <u>27,76</u> (90)	24,98 (100)
Gash1	28,84 (100)	- <u>29,08</u> (70)	29,16 (100)	- <u>29,68</u> (90)	30,40 (100)	- <u>30,98</u> (90)	23,09 (100)
Gash2	24,58 (100)	- <u>25,16</u> (80)	<u>24,72</u> (30)	- <u>24,72</u> (80)	25,89 (100)	- <u>26,29</u> (80)	21,92 (40)
Phenotype	<u>25,40</u> (10)	• 22,00 (10)	<u>25,23</u> (10)	- 24,94 (40)	25,77 (80)	- <u>25,84</u> (40)	25,99 (10)
Sequence	<u>23,71</u> (70)	- 22,43 (100)	<u>23,59</u> (100)	- <u>23,59</u> (100)	<u>24,05</u> (70)	- 23,64 (100)	21,65 (80)
SPO	<u>19,01</u> (90)	- 18,23 (70)	<u>19,56</u> (90)	- 19,14 (100)	<u>21,10</u> (90)	- 20,47 (100)	19,86 (100)
Total B	5	5	7	6	5	6	-

A Tabela 4.8 apresenta os resultados do classificador hierárquico *Per Parent Top-Down* com seleção de atributos *eager* e *lazy*. Observa-se, através das linhas de totais, que, para as bases do Grupo A, o classificador hierárquico PPTD com seleção de atributos *eager* obteve um desempenho melhor em relação à versão com seleção *lazy*. Além disso, nota-se que a seleção de atributos *lazy* não foi eficaz, pois muitos resultados foram encontrados com 100% dos atributos selecionados. Analisando as bases do Grupo B, observa-se, através das linhas de totais, um equilíbrio nos resultados encontrados. Considerando os melhores resultados encontrados por base de dados (sublinhados), o classificador PPTD com seleção *eager* encontrou 16, enquanto o classificador PPTD com seleção *lazy* obteve três desses resultados.

A Tabela 4.9 apresenta os resultados da estratégia *Sum of Weighted Votes* com seleção de atributos *eager* e *lazy*. Analisando esses resultados, observa-se, através das linhas de totais, que as versões *eager* e *lazy* da estratégia SW apresentam desempenho semelhantes. Entretanto, considerando os melhores resultados encontrados por base de dados (sublinhados), a estratégia SW com seleção *eager* encontrou oito, enquanto a estratégia SW com seleção *lazy* obteve 10 desses resultados.

A Tabela 4.10 apresenta os melhores resultados obtidos para cada um dos classificadores hierárquicos implementados com seleção de atributos *eager* e *lazy*. Consolidam-se

Tabela 4.8: hF do Classificador PPTD com seleção de atributos *Eager* e *Lazy*

Bases	1-NN		7-NN		9-NN		C4.5
	Eager	Lazy	Eager	Lazy	Eager	Lazy	Eager
GPCRpfam	70,32 (90) - 70,32 (100)	70,32 (90) - 70,32 (100)	69,09 (70) - 69,04 (100)	69,04 (100)	68,55 (70) - 68,47 (100)	68,47 (100)	68,85 (70)
GPCRprints	82,97 (80) - 82,97 (100)	82,97 (80) - 82,97 (100)	80,95 (70) - 80,89 (100)	80,89 (100)	80,41 (80) - 80,41 (100)	80,41 (80) - 80,41 (100)	79,22 (50)
GPCRprosite	69,26 (70) - 69,25 (100)	69,25 (100)	67,38 (70) - 67,31 (100)	67,31 (100)	66,57 (90) - 66,57 (100)	66,57 (90) - 66,57 (100)	67,67 (70)
GPCRinterpro	83,09 (90) - 83,09 (100)	83,09 (90) - 83,09 (100)	81,96 (80) - 81,95 (100)	81,95 (100)	81,29 (80) - 81,29 (100)	81,29 (80) - 81,29 (100)	81,54 (100)
ECpfam	98,77 (70) - 98,77 (100)	98,77 (70) - 98,77 (100)	98,16 (70) - 98,17 (70)	98,17 (70)	97,88 (70) - 97,88 (100)	97,88 (70) - 97,88 (100)	98,40 (60)
ECprints	98,19 (80) - 98,19 (100)	98,19 (80) - 98,19 (100)	97,37 (80) - 97,37 (100)	97,37 (80) - 97,37 (100)	97,05 (80) - 97,05 (100)	97,05 (80) - 97,05 (100)	97,35 (100)
ECprosite	98,81 (70) - 98,80 (100)	98,80 (100)	98,29 (70) - 98,29 (100)	98,29 (70) - 98,29 (100)	98,03 (80) - 98,03 (100)	98,03 (80) - 98,03 (100)	98,46 (70)
ECinterpro	99,07 (30) - 99,07 (10)	99,07 (30) - 99,07 (10)	98,62 (70) - 98,62 (100)	98,62 (70) - 98,62 (100)	98,32 (70) - 98,32 (100)	98,32 (70) - 98,32 (100)	98,73 (100)
Total A	8	6	7	4	8	7	-
Church	21,64 (10) - 20,76 (10)	20,76 (10)	23,06 (10) - 22,87 (10)	22,87 (10)	23,07 (10) - 23,43 (10)	23,43 (10) - 23,43 (10)	25,29 (10)
CellCycle	24,75 (40) - 24,98 (90)	24,98 (90) - 24,98 (90)	29,29 (50) - 28,38 (100)	28,38 (100)	29,91 (40) - 29,40 (100)	29,91 (40) - 29,91 (100)	22,93 (80)
Derise	20,09 (70) - 19,58 (80)	19,58 (80)	22,07 (70) - 21,70 (50)	21,70 (50)	22,71 (30) - 22,28 (50)	22,71 (30) - 22,71 (50)	22,82 (10)
Eisen	25,66 (50) - 28,67 (90)	28,67 (90) - 28,67 (90)	29,53 (70) - 30,88 (80)	30,88 (80) - 30,88 (80)	30,39 (50) - 30,59 (80)	30,59 (80) - 30,59 (80)	26,77 (50)
Expr	26,05 (10) - 26,78 (70)	26,78 (70) - 26,78 (70)	29,33 (70) - 30,41 (30)	30,41 (30) - 30,41 (30)	29,78 (70) - 30,49 (30)	30,49 (30) - 30,49 (30)	26,26 (30)
Gash1	28,29 (100) - 28,29 (100)	28,29 (100) - 28,29 (100)	31,37 (80) - 30,97 (100)	30,97 (100)	30,08 (100) - 30,67 (70)	30,67 (70) - 30,67 (70)	24,34 (90)
Gash2	25,23 (100) - 25,67 (60)	25,67 (60) - 25,67 (60)	27,89 (80) - 27,13 (30)	27,13 (30)	27,92 (70) - 26,85 (50)	27,92 (70) - 27,92 (70)	23,39 (40)
Phenotype	21,34 (20) - 22,14 (10)	22,14 (10) - 22,14 (10)	23,23 (20) - 22,72 (10)	22,72 (10)	22,95 (90) - 22,95 (90)	22,95 (90) - 22,95 (90)	21,57 (60)
Sequence	25,01 (90) - 24,02 (100)	24,02 (100)	24,88 (80) - 24,97 (10)	24,97 (10) - 24,97 (10)	25,63 (30) - 24,69 (10)	25,63 (30) - 25,63 (30)	25,26 (10)
SPO	21,70 (20) - 21,45 (50)	21,45 (50)	24,55 (30) - 24,57 (30)	24,57 (30) - 24,57 (30)	26,05 (30) - 24,49 (30)	26,05 (30) - 26,05 (30)	23,83 (10)
Total B	5	6	6	4	6	5	-

Tabela 4.9: hF da Estratégia SW com seleção de atributos *Eager* e *Lazy*

Bases	1-NN		7-NN		9-NN		C4.5
	Eager	Lazy	Eager	Lazy	Eager	Lazy	Eager
GPCRpfam	70,31 (90) - 70,31 (50)	70,31 (90) - 70,31 (50)	68,78 (50) - 68,82 (50)	68,82 (50) - 68,82 (50)	68,25 (40) - 68,29 (50)	68,29 (50) - 68,29 (50)	68,76 (50)
GPCRprints	83,00 (80) - 83,00 (80)	83,00 (80) - 83,00 (80)	80,97 (70) - 80,94 (70)	80,94 (70)	80,29 (70) - 80,28 (100)	80,29 (70) - 80,29 (100)	79,53 (50)
GPCRprosite	69,36 (40) - 69,41 (30)	69,41 (30) - 69,41 (30)	67,13 (40) - 67,12 (30)	67,12 (30)	66,14 (30) - 66,14 (30)	66,14 (30) - 66,14 (30)	67,13 (20)
GPCRinterpro	83,09 (90) - 83,09 (90)	83,09 (90) - 83,09 (90)	81,69 (80) - 81,68 (90)	81,68 (90)	81,31 (80) - 81,31 (80)	81,31 (80) - 81,31 (80)	81,98 (90)
ECpfam	98,77 (70) - 98,79 (20)	98,79 (20) - 98,79 (20)	98,30 (70) - 98,30 (70)	98,30 (70) - 98,30 (70)	98,19 (70) - 98,19 (70)	98,19 (70) - 98,19 (70)	98,43 (60)
ECprints	98,19 (80) - 98,21 (30)	98,21 (30) - 98,21 (30)	97,45 (80) - 97,45 (80)	97,45 (80) - 97,45 (80)	97,22 (70) - 97,22 (80)	97,22 (70) - 97,22 (80)	97,54 (80)
ECprosite	98,81 (70) - 98,80 (70)	98,80 (70)	98,35 (60) - 98,34 (70)	98,34 (70)	98,10 (70) - 98,08 (60)	98,10 (70) - 98,10 (70)	98,57 (70)
ECinterpro	99,08 (70) - 99,15 (20)	99,15 (20) - 99,15 (20)	98,82 (70) - 98,82 (70)	98,82 (70) - 98,82 (70)	98,70 (70) - 98,70 (70)	98,70 (70) - 98,70 (70)	98,79 (100)
Total A	4	7	7	4	7	5	-
Church	21,88 (10) - 21,96 (10)	21,96 (10) - 21,96 (10)	22,96 (10) - 23,59 (10)	23,59 (10) - 23,59 (10)	23,22 (10) - 24,16 (10)	24,16 (10) - 24,16 (10)	25,58 (10)
CellCycle	25,12 (40) - 26,01 (80)	26,01 (80) - 26,01 (80)	30,16 (40) - 29,38 (50)	29,38 (50)	30,96 (40) - 29,87 (50)	30,96 (40) - 30,96 (50)	24,86 (40)
Derise	20,26 (60) - 19,71 (60)	19,71 (60)	23,22 (10) - 22,55 (30)	22,55 (30)	23,52 (40) - 23,18 (20)	23,52 (40) - 23,52 (20)	22,44 (10)
Eisen	27,26 (50) - 27,51 (60)	27,51 (60) - 27,51 (60)	32,43 (50) - 31,32 (60)	31,32 (60)	32,51 (50) - 32,74 (30)	32,74 (30) - 32,74 (30)	28,20 (70)
Expr	26,18 (30) - 28,05 (20)	28,05 (20) - 28,05 (20)	29,85 (50) - 31,74 (40)	31,74 (40) - 31,74 (40)	30,42 (40) - 31,72 (30)	31,72 (30) - 31,72 (30)	28,92 (60)
Gash1	29,60 (90) - 28,98 (100)	28,98 (100)	31,67 (80) - 31,34 (70)	31,34 (70)	30,80 (100) - 31,95 (70)	31,95 (70) - 31,95 (70)	27,30 (90)
Gash2	26,46 (50) - 25,19 (80)	25,19 (80)	28,39 (50) - 26,83 (80)	26,83 (80)	28,93 (40) - 27,42 (40)	28,93 (40) - 28,93 (40)	24,62 (30)
Phenotype	26,25 (10) - 24,34 (10)	24,34 (10)	26,69 (10) - 27,05 (10)	27,05 (10) - 27,05 (10)	26,46 (10) - 27,08 (10)	27,08 (10) - 27,08 (10)	27,94 (10)
Sequence	23,99 (70) - 23,08 (100)	23,08 (100)	24,19 (40) - 24,00 (10)	24,00 (10)	25,48 (40) - 25,09 (10)	25,48 (40) - 25,48 (10)	26,18 (70)
SPO	21,95 (20) - 22,03 (20)	22,03 (20) - 22,03 (20)	24,98 (30) - 25,02 (10)	25,02 (10) - 25,02 (10)	25,64 (20) - 24,96 (40)	25,64 (20) - 25,64 (40)	23,38 (40)
Total B	5	5	6	4	5	5	-

para cada base de dados, os melhores resultados encontrados por cada um dos classificadores hierárquicos (FL, PPTD e SW) com a aplicação de seleção de atributos *eager* e *lazy*, apresentados nas tabelas anteriores. Para cada combinação de base de dados e classificador hierárquico analisado, são apresentados os valores da medida hF e, ao lado, têm-se o classificador plano adotado e a porcentagem de atributos que foram selecionados. Além disso, os melhores resultados encontrados por base de dados são sublinhados.

Observa-se, nessa tabela, através das linhas de totais, que o classificador hierárquico FL obteve melhor desempenho quando aplicado o método de seleção *lazy* para as bases da Grupo A. Para as bases do Grupo B, o classificador FL com o método de seleção *eager* atinge o melhor resultado em seis das 10 bases. No classificador hierárquico PPTD, a aplicação da seleção de atributos *eager* obtém melhores resultados em 16 das 18 bases de dados. Para os resultados do classificador hierárquico com a estratégia SW, percebe-se que para as bases do Grupo A, o método de seleção *lazy* obtém melhor desempenho, por outro lado, nas bases do grupo B, o método de seleção *eager* alcança melhores resultados. Conclui-se, também, analisando os melhores resultados encontrados por base de dados (sublinhados), que o classificador hierárquico com a estratégia SW, mostra-se competitivo, obtendo 13 dos 18 melhores resultados. O classificador FL obteve cinco e o PPTD apenas quatro dos melhores resultados.

Tabela 4.10: Influência da seleção de atributos *Eager* e *Lazy* nos classificadores hierárquicos

Bases	FL				PPTD				SW			
	Eager		Lazy		Eager		Lazy		Eager		Lazy	
GPCRpfam	70,31	1-NN(90)	<u>70,32</u>	1-NN(70)	<u>70,32</u>	1-NN(90)	<u>70,32</u>	1-NN(100)	70,30	1-NN(90)	<u>70,31</u>	1-NN(50)
GPCRprints	82,98	1-NN(80)	<u>82,99</u>	1-NN(70)	<u>82,97</u>	1-NN(80)	<u>82,97</u>	1-NN(100)	<u>83,00</u>	1-NN(80)	<u>83,00</u>	1-NN(80)
GPCRprosite	69,35	1-NN(50)	<u>69,42</u>	1-NN(30)	<u>69,26</u>	1-NN(70)	69,25	1-NN(100)	69,36	1-NN(40)	<u>69,41</u>	1-NN(30)
GPCRinterpro	83,09	1-NN(80)	<u>83,12</u>	1-NN(70)	<u>83,09</u>	1-NN(90)	<u>83,09</u>	1-NN(100)	<u>83,09</u>	1-NN(90)	<u>83,09</u>	1-NN(90)
ECpfam	98,77	1-NN(70)	<u>98,80</u>	1-NN(20)	<u>98,77</u>	1-NN(70)	<u>98,77</u>	1-NN(100)	98,77	1-NN(70)	<u>98,79</u>	1-NN(20)
ECprints	98,19	1-NN(80)	<u>98,21</u>	1-NN(20)	<u>98,19</u>	1-NN(80)	<u>98,19</u>	1-NN(100)	98,19	1-NN(80)	<u>98,21</u>	1-NN(30)
ECprosite	<u>98,80</u>	1-NN(70)	<u>98,80</u>	1-NN(80)	<u>98,81</u>	1-NN(70)	98,80	1-NN(100)	<u>98,81</u>	1-NN(70)	98,80	1-NN(70)
ECinterpro	99,08	1-NN(70)	<u>99,14</u>	1-NN(10)	<u>99,07</u>	1-NN(30)	<u>99,07</u>	1-NN(10)	99,07	1-NN(70)	<u>99,15</u>	1-NN(20)
Total A	1		8		8		6		3		7	
Church	<u>25,23</u>	1-NN(10)	24,30	9-NN(10)	<u>25,29</u>	C4.5(10)	23,43	9-NN(10)	<u>25,58</u>	C4.5(10)	24,16	9-NN(10)
CellCycle	<u>29,08</u>	9-NN(100)	<u>29,08</u>	9-NN(100)	<u>29,91</u>	9-NN(40)	29,40	9-NN(100)	<u>30,96</u>	9-NN(40)	29,87	9-NN(50)
Derise	20,17	9-NN(100)	<u>20,41</u>	9-NN(90)	<u>22,82</u>	C4.5(10)	22,28	9-NN(50)	<u>23,52</u>	9-NN(40)	23,18	9-NN(20)
Eisen	<u>29,03</u>	9-NN(70)	28,73	9-NN(100)	30,39	9-NN(50)	<u>30,88</u>	7-NN(80)	32,51	9-NN(50)	<u>32,74</u>	9-NN(30)
Expr	27,58	9-NN(100)	<u>27,76</u>	9-NN(90)	29,78	9-NN(70)	<u>30,49</u>	9-NN(30)	30,42	9-NN(40)	<u>31,74</u>	7-NN(40)
Gash1	30,40	9-NN(100)	<u>30,98</u>	9-NN(90)	<u>31,37</u>	7-NN(80)	30,97	7-NN(100)	31,67	7-NN(80)	<u>31,95</u>	9-NN(70)
Gash2	25,89	9-NN(100)	<u>26,29</u>	9-NN(80)	<u>27,92</u>	9-NN(70)	27,13	7-NN(30)	<u>28,93</u>	9-NN(40)	27,42	9-NN(40)
Phenotype	<u>25,99</u>	C4.5(10)	25,84	9-NN(40)	<u>23,23</u>	7-NN(20)	22,95	9-NN(90)	<u>27,94</u>	C4.5(10)	27,08	9-NN(10)
Sequence	<u>24,05</u>	9-NN(70)	23,64	9-NN(100)	<u>25,63</u>	9-NN(30)	24,97	7-NN(10)	<u>26,18</u>	C4.5(70)	25,71	C4.5(100)
SPO	<u>21,10</u>	9-NN(90)	20,47	9-NN(100)	<u>26,05</u>	9-NN(30)	24,57	7-NN(30)	<u>25,64</u>	9-NN(20)	25,02	7-NN(10)
Total B	6		5		8		2		7		3	

Na Tabela 4.11, consolidam-se todos os melhores resultados para cada um dos classificadores hierárquicos implementados. Para cada combinação de base de dados e classificador hierárquico analisado (FL, PPTD e SW), são apresentados os melhores valores obtidos

da medida hF e, ao lado, têm-se a combinação de classificador plano, o método de seleção utilizado (*eager* ou *lazy*) e porcentagem de atributos que levou ao melhor valor. Além disso, os melhores resultados por base de dados entre os classificadores FL, PPTD e SW estão marcados em negrito. Exemplificando, para a base *GPCRpfam*, o maior valor de hF atingido foi de 70,32, obtido pelo classificador hierárquico FL, utilizando o classificador plano 1-NN, seleção de atributos *lazy*, com seleção de 70% dos atributos .

Analisando essa tabela, pode-se concluir que:

- O classificador hierárquico com a estratégia SW, proposta neste trabalho, apresentou um desempenho satisfatório. Obteve quatro dos oito melhores resultados para as bases do Grupo A e nove dos 10 melhores resultados para as bases do Grupo B. O classificador FL obteve cinco melhores resultados das bases do Grupo A e nenhum do Grupo B. O classificador PPTD atingiu dois melhores resultados das bases do Grupo A e apenas um melhor resultado para as bases do Grupo B.
- Entre os classificadores hierárquicos tradicionais FL e PPTD, percebe-se um melhor desempenho do classificador FL nas bases do Grupo A, porém, para as bases do Grupo B, o classificador PPTD alcança sempre melhores resultados.
- Os melhores resultados por base (negrito), no Grupo A, foram obtidos quando adotado o classificador 1-NN e, no Grupo B, em grande parte das bases, quando utilizado o classificador 9-NN.
- Métodos de seleção de atributos aplicados a classificadores hierárquicos, em grande parte dos casos, conseguiram aumentar o desempenho desses classificadores. Em apenas seis dos 54 casos, a utilização de todos os atributos (100%) levou ao melhor resultado.
- O método de seleção de atributos *lazy* conseguiu melhorar ainda mais, alguns dos resultados encontrados com seleção de atributos *eager*. Em 31, dos 54 casos explorados, a estratégia *lazy* conseguiu atingir o melhor resultado.

A Tabela 4.12 apresenta, para cada base de dados, os melhores resultados obtidos e as estratégias que os atingiram. Cada estratégia está representada pelo classificador hierárquico aplicado, classificador plano utilizado, tipo de seleção de atributos (*eager* ou *lazy*) e a porcentagem de atributos que foram selecionados. Mais uma vez, observa-se a importância da utilização de alguma estratégia de seleção de atributos (*eager* ou *lazy*) uma

Tabela 4.11: Melhores resultados dos classificadores hierárquicos FL, PPTD e SW

Bases	FL		PPTD		SW	
	hF	Estratégia	hF	Estratégia	hF	Estratégia
GPCRpfam	70,32	1-NN-LAZY(70)	70,32	1-NN-EAGER(90) 1-NN-LAZY(100)	70,31	1-NN-LAZY(50)
GPCRprints	82,99	1-NN-LAZY(70)	82,97	1-NN-EAGER(80) 1-NN-LAZY(100)	83,00	1-NN-EAGER(80) 1-NN-LAZY(80)
GPCRprosite	69,42	1-NN-LAZY(30)	69,26	1-NN-EAGER(70)	69,41	1-NN-LAZY(30)
GPCRinterpro	83,12	1-NN-LAZY(70)	83,09	1-NN-EAGER(90) 1-NN-LAZY(100)	83,09	1-NN-EAGER(90) 1-NN-LAZY(90)
ECpfam	98,80	1-NN-LAZY(20)	98,77	1-NN-EAGER(70) 1-NN-LAZY(100)	98,79	1-NN-LAZY(20)
ECprints	98,21	1-NN-LAZY(20)	98,19	1-NN-EAGER(80) 1-NN-LAZY(100)	98,21	1-NN-LAZY(30)
ECprosite	98,80	1-NN-EAGER(70) 1-NN-LAZY(80)	98,81	1-NN-EAGER(70)	98,81	1-NN-EAGER(70)
ECinterpro	99,14	1-NN-LAZY(10)	99,07	1-NN-EAGER(30) 1-NN-LAZY(10)	99,15	1-NN-LAZY(20)
TOTAL A		5		2		4
Church	25,23	1-NN-EAGER(10)	25,29	C4.5-EAGER(10)	25,58	C4.5-EAGER(10)
CellCycle	29,08	9-NN-EAGER(100) 9-NN-LAZY(100)	29,91	9-NN-EAGER(40)	30,96	9-NN-EAGER(40)
Derisi	20,41	9-NN-LAZY(90)	22,82	C4.5-EAGER(10)	23,52	9-NN-EAGER(40)
Eisen	29,03	9-NN-EAGER(70)	30,88	7-NN-LAZY(80)	32,74	9-NN-LAZY(30)
Expr	27,76	9-NN-LAZY(90)	30,49	9-NN-LAZY(30)	31,74	7-NN-LAZY(40)
Gasch1	30,98	9-NN-LAZY(90)	31,37	7-NN-EAGER(80)	31,95	9-NN-LAZY(70)
Gasch2	26,29	9-NN-LAZY(80)	27,92	9-NN-EAGER(70)	28,93	9-NN-EAGER(40)
Phenotype	25,99	C4.5-EAGER(10)	23,23	7-NN-EAGER(20)	27,94	C4.5-EAGER(10)
Sequence	24,05	9-NN-EAGER(70)	25,63	9-NN-EAGER(30)	26,18	C4.5-EAGER(70)
SPO	21,10	9-NN-EAGER(90)	26,05	9-NN-EAGER(30)	25,64	9-NN-EAGER(20)
TOTAL B		0		1		9

vez que, apenas para a base GPCRpfam, a base inteira (100% dos atributos) deu origem ao melhor resultado. Percebe-se também a importância de cada estratégia de seleção de atributos, isoladamente, pois cada uma obteve individualmente o melhor resultado para subconjuntos distintos de oito bases de dados.

Destaca-se ainda o bom desempenho da estratégia proposta – *Sum of Weighted Votes* (SW) – que conseguiu obter melhor resultado em 13 das 18 bases, sendo em 11 delas de forma isolada.

Tabela 4.12: Melhores resultados encontrados por base de dados

Grupo	Bases	hF	Estratégia(s)
A	GPCRpfam	70,32	FL/1-NN-LAZY(90), PPTD/1-NN-EAGER(90) e PPTD/1-NN-LAZY(100)
	GPCRprints	83,00	SW/1-NN-EAGER(80) e SW/1-NN-LAZY(80)
	GPCRprosite	69,42	FL/1-NN-LAZY(30)
	GPCRinterpro	83,12	FL/1-NN-LAZY(70)
	ECpfam	98,80	FL/1-NN-LAZY(20)
	ECprints	98,21	FL/1-NN-LAZY(20) e SW/1-NN-LAZY(30)
	ECprosite	98,81	PPTD/1-NN-EAGER(70) e SW/1-NN-EAGER(70)
	ECinterpro	99,15	SW/1-NN-LAZY(20)
B	Church	25,58	SW/C4.5-EAGER(10)
	CellCycle	30,96	SW/9-NN-EAGER(40)
	Derisi	23,52	SW/9-NN-EAGER(40)
	Eisen	32,74	SW/9-NN-LAZY(30)
	Expr	31,74	SW/7-NN-LAZY(40)
	Gasch1	31,95	SW/9-NN-LAZY(70)
	Gasch2	28,93	SW/9-NN-EAGER(40)
	Phenotype	27,94	SW/C4.5-EAGER(10)
	Sequence	26,18	SW/C4.5-EAGER(70)
	SPO	26,05	PPTD/9-NN-EAGER(30)

Capítulo 5

Conclusões

Problemas de classificação hierárquica têm sido explorados utilizando-se diferentes abordagens na construção dos classificadores. A abordagem por classificadores locais é uma das mais utilizadas e eficazes para classificação hierárquica, na qual a estratégia explora a hierarquia de classes através de uma perspectiva local, combinando classificadores que consideram, isoladamente, diferentes partes da hierarquia.

A abordagem de classificação local por nível, apesar de necessitar do treinamento de um número reduzido de classificadores locais, um por nível hierárquico, ainda não possui muitos trabalhos e pesquisas relacionadas na literatura. Provavelmente, devido à ocorrência de inconsistências nos resultados obtidos pelos classificadores associados aos diferentes níveis da hierarquia.

Neste trabalho, foram propostas duas estratégias hierárquicas com tratamento posterior de inconsistência *Sum of Votes* (SV) e *Sum of Weighted Votes* (SW) aplicadas em classificadores hierárquicos locais por nível. Utilizando 18 bases hierárquicas, foram avaliadas ambas as estratégias e concluiu-se que a estratégia SW atingiu melhores acurácias preditivas. No segundo experimento, comparou-se o classificador hierárquico local por nível usando a estratégias SW com duas abordagens tradicionais de classificação hierárquica: abordagem de classificação plana (*Flat Classification Approach*) e abordagem de classificação local por nó pai (*Local Classification Per Parent Approach*). Os resultados obtidos mostraram que a estratégia *Sum of Weighted Votes* apresentou novamente um desempenho competitivo e robusto em termos de acurácia preditiva, evidenciando a importância da abordagem de classificação local por nível. Esses resultados estão publicados em [25].

Outra contribuição deste trabalho foi a introdução de métodos de seleção de atributos

nos classificadores hierárquicos explorados. Foram aplicados dois métodos de seleção de atributos: o método tradicional, denominado *eager*, no qual a seleção de atributos é realizada uma única vez, anteriormente a fase de classificação das instâncias e um método de seleção de atributos recentemente proposto [26], considerado *lazy*, pois adia a seleção de atributos até o momento da classificação de novas instâncias. Usando as mesmas 18 bases hierárquicas, inicialmente, avaliou-se o impacto do método *eager* no desempenho dos classificadores hierárquicos tradicionais (*Flat On Leaves* e *Per Parent Top-Down*) e no desempenho do classificador hierárquico local por nível com o tratamento de inconsistência proposto (*Sum of Weighted Votes*). Concluiu-se que, para todas as 18 bases de dados, o melhor resultado foi obtido por uma estratégia hierárquica quando utilizada a seleção de atributos *eager*. Apenas em uma base, um dos classificadores sem seleção de atributos também atingiu o melhor resultado. No segundo experimento, foi avaliado o impacto do método de seleção *lazy* nos mesmos classificadores hierárquicos utilizados anteriormente. Concluiu-se que o método de seleção de atributos *lazy* conseguiu melhorar ainda mais alguns dos resultados encontrados com seleção de atributos *eager*.

Como trabalho futuro, considera-se avaliar extensões do método proposto *Sum of Weighted Votes* (SW), visando: (a) priorizar votos referentes a classes em níveis mais próximos à raiz da hierarquia, onde as previsões tendem a ter mais chances de acerto, e (b) normalizar os pesos dos votos, considerando o tamanho dos ramos, de forma que, por exemplo, dois votos em um ramo de apenas duas classes tenha maior relevância do que três votos em um ramo de cinco classes. Além disso, visando avaliar a aplicabilidade das técnicas propostas, considera-se realizar novos experimentos utilizando-se outras bases de dados de outros domínios de aplicação, com diferentes características, principalmente no que se refere à hierarquia de classes.

Referências

- [1] BARBEDO, J. G. A., LOPES, A. Automatic genre classification of musical signals. *EURASIP Journal on Applied Signal Processing 2007*, 1 (2007), p. 157–157.
- [2] BARUTCUOGLU, Z., DECORO, C. Hierarchical shape classification using Bayesian aggregation. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications* (2006), p. 44–44.
- [3] BURRED, J. J., LERCH, A. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th International Conference on Digital Audio Effects* (2003), p. 8–11.
- [4] CECI, M., MALERBA, D. Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems 28*, 1 (2007), p. 37–78.
- [5] CLARE, A., KING, R. D. Predicting gene function in *Saccharomyces cerevisiae*. In *Proceedings of the European Conference on Computational Biology* (2003), p. 42–49.
- [6] COSTA, E. P., LORENA, A. C., CARVALHO, FREITAS, A. A. A review of performance evaluation measures for hierarchical classifiers. In *Evaluation Methods for Machine Learning II: papers from the 2007 AAI Workshop* (2007), AAAI Press, p. 1–6.
- [7] COSTA, E. P., LORENA, A. C., CARVALHO, A. C. P. L. F., FREITAS, A. A. Top-down hierarchical ensembles of classifiers for predicting G-Protein-Coupled-Receptor functions. In *Proceedings of the 3rd Brazilian Symposium on Bioinformatics, Lecture Notes in Bioinformatics 5167* (2008), p. 35–46.
- [8] COSTA, E. P., LORENA, A. C., CARVALHO, A. C. P. L. F., FREITAS, A. A., HOLDEN, H. Comparing several approaches for hierarchical classification of proteins with decision trees. In *Proceedings of the 2nd Brazilian Symposium on Bioinformatics, Lecture Notes in Bioinformatics 4643* (2007), p. 126–137.
- [9] D’ALESSIO, S., MURRAY, K., SCHIAFFINO, R., KERSHENBAUM, A. The effect of using hierarchical classifiers in text categorization. In *Proceedings of 6th International Conference Recherche d’Information Assistee par Ordinateur* (2000), p. 302–313.
- [10] DUMAIS, S., CHEN, H. Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2000), p. 256–263.

-
- [11] EISNER, R., POULIN, B., SZAFRON, D., LU, P., GREINER, R. Improving protein function prediction using the hierarchical structure of the Gene Ontology. In *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* (2005), p. 354–363.
- [12] FAGNI, T., SEBASTIANI, F. On the selection of negative examples for hierarchical text categorization. In *Proceedings of the 3rd Language & Technology Conference* (2007), p. 24–28.
- [13] FREITAS, A. A., CARVALHO, A. C. P. F. L. A tutorial on hierarchical classification with applications in bioinformatics. In *Research and Trends in Data Mining Technologies and Applications*, D. Taniar, Ed. Idea Group, 2007, p. 175–208.
- [14] GUYON, I., ELISSEEFF, A. An introduction to feature extraction. In *Feature Extraction, Foundations and Applications* (2006), Springer, p. 1–24.
- [15] HALL, M. A. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th International Conference on Machine Learning* (2000), Morgan Kaufmann Publishers Inc., p. 359–366.
- [16] HAN, J., KAMBER, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [17] HOLDEN, N., FREITAS, A. A. A hybrid pso/aco algorithm for classification. In *Proceedings of the GECCO Conference Companion on Genetic and Evolutionary Computation* (2007), ACM, p. 2745–2750.
- [18] HOLDEN, N., FREITAS, A. A. Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation. *Soft Computing* 13 (2008), p. 259–272.
- [19] HOLDEN, N., FREITAS, A. A. Improving the performance of hierarchical classification with swarm intelligence. In *Proceedings of the 6th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, LNCS 4973* (2008), p. 48–60.
- [20] JAIN, R. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991.
- [21] KIRITCHENKO, S., MATWIN, S., FAMILI, A. F. Functional annotation of genes using hierarchical text categorization. In *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology* (2005).
- [22] KOLLER, D., SAHAMI, M. Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning* (1997), p. 170–178.
- [23] LABROU, Y., FININ, T. Yahoo! as an ontology: using yahoo! categories to describe documents. In *Proceedings of the 8th International Conference on Information and Knowledge Management* (1999), p. 180–187.

- [24] LIU, H., SETIONO, R. A probabilistic approach to feature selection - a filter solution. In *Proceedings of the 13th International Conference on Machine Learning* (1996), Morgan Kaufmann, p. 319–327.
- [25] PAES, B. C., PLASTINO, A., FREITAS, A. A. Improving local per level hierarchical classification. *Journal of Information and Data Management (to appear)* (2012).
- [26] PEREIRA, R., PLASTINO, A., ZADROZNY, B., MERSCHMANN, L., FREITAS, A. Lazy attribute selection: Choosing attributes at classification time. *Intelligent Data Analysis* 15, 5 (2011), p. 715–732.
- [27] ROCCHIO, J. J. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton, Ed. Prentice-Hall, 1971, p. 313–323.
- [28] SECKER, A., DAVIES, M. N., FREITAS, A. A., CLARK, E. B., TIMMIS, J., FLOWER, D. R. Hierarchical classification of G-Protein-Coupled Receptors with data-driven selection of attributes and classifiers. *International Journal of Data Mining and Bioinformatics* 4, 2 (2010), p. 191–210.
- [29] SECKER, A., DAVIES, M. N., FREITAS, A. A., TIMMIS, J., MENDAO, M., FLOWER, D. R. An experimental comparison of classification algorithms for the hierarchical prediction of protein function. In *Proceedings of the 3rd UK Data Mining and Knowledge Discovery Symposium* (2007), p. 13–18.
- [30] SILLA, C., FREITAS, A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22, 1-2 (2011), p. 31–72.
- [31] SILLA, C. N., FREITAS, A. A global-model naive bayes approach to the hierarchical prediction of protein functions. In *Proceedings of the 9th IEEE International Conference on Data Mining* (2009), p. 992–997.
- [32] SUN, A., LIM, E.-P. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining* (2001), IEEE Computer Society, p. 521–528.
- [33] VALENTINI, G. True path rule hierarchical ensembles. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems* (2009), p. 232–241.
- [34] WITTEN, I. H., FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Third Edition, 2011.
- [35] WU, F., ZHANG, J., HONAVAR, V. Learning classifiers using hierarchically structured class taxonomies. In *Proceedings of the Symposium on Abstraction, Reformulation, and Approximation* (2005), p. 313–320.
- [36] XIAO, Z., DELLANDREA, E., DOU, W., CHEN, L. Automatic hierarchical classification of emotional speech. In *Proceedings of the 9th IEEE International Symposium on Multimedia Workshops* (2007), p. 291–296.
- [37] YANG, Y., PEDERSEN, J. O. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning* (1997), Morgan Kaufmann Publishers Inc., p. 412–420.