Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Um estudo sobre os métodos de classificação hierárquica e medidas de avaliação

Jean Metz
Alex A. Freitas
Maria Carolina Monard

N° 364

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Maio/2011

Um estudo sobre os métodos de classificação hierárquica e medidas de avaliação* †

Jean Metz[1]
Alex A. Freitas[2]
Maria Carolina Monard[1]

[1]Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {metzz, mcmonard}@icmc.usp.br

[2]Kent University
School of Computing
Canterbury, Kent, United Kingdom
e-mail: A.A.Freitas@kent.ac.uk

Maio, 2011

*Trabalho desenvolvido com o apoio da CAPES e da FAPESP.
†Disponível on-line em: http://www.icmc.usp.br/~biblio/relatorios_tecnicos.php.

Resumo

Tarefas de classificação em Aprendizado de Máquina e Mineração de Dados são usualmente associadas a problemas cujos domínios possuem exemplos que participam de apenas uma classe sem qualquer relação estrutural ou hierárquica, denominados classificação plana. Entretanto, existem problemas de classificação mais complexos, nos quais cada exemplo pode estar associado a um subconjunto de classes estruturadas em uma hierarquia. A tarefa para esses problemas é predizer o subconjunto de classes para novos exemplos, de modo que as classes preditas respeitem a estrutura hierárquica e suas relações latentes. Essa tarefa é chamada de classificação hierárquica. O objetivo deste trabalho é apresentar os conceitos que envolvem essa tarefa, as principais abordagens e alguns dos algoritmos amplamente utilizados na literatura. São também apresentadas várias medidas utilizadas para avaliar a performance dos algoritmos de classificação hierárquica. Além disso, são propostos dois novos métodos que são extensões dos métodos tradicionais e fazem uso das boas características desses métodos para tentar minimizar alguns fatores que podem degradar o modelo final.

Palavras-Chave: Classificação hierárquica, Hierarquia de classes, Avaliação de classificação hierárquica

Este documento foi preparado com o formatador de textos LaTeX. O sistema de citações de referências bibliográficas utiliza o padrão *Apalike* do sistema bibTeX.

Sumário

Sumário v

Lista de Figuras vii

Lista de Tabelas ix

1 Introdução 1

2 Classificação Plana *versus* Hierárquica 2

3 Fundamentos e Definições para Classificação Hierárquica 3
- 3.1 Estrutura de Representação da Hierarquia de Classes 4
- 3.2 Restrição no Nível de Classificação 6
- 3.3 Uso da Estrutura de Classes 7
  - 3.3.1 Abordagem Plana 7
  - 3.3.2 Abordagem Global 8
  - 3.3.3 Abordagem Local 9
    - 3.3.3.1 Treinamento do modelo HBR 10
    - 3.3.3.2 Treinamento do modelo HMR 17
  - 3.3.4 Estratégia de Predição *Top-down* para Métodos Locais . . . 19

4 Algoritmos Propostos 21
- 4.1 HMR com Estratégia de Multiplicação 22
- 4.2 Modelo Híbrido de Cooperação: HBR + HMR 26

5 Medidas de Avaliação 29
- 5.1 Medidas de Avaliação da Classificação Plana 30
- 5.2 Medidas de Avaliação da Classificação Hierárquica 31
  - 5.2.1 Baseadas em Distância 34
    - 5.2.1.1 Dependente de Profundidade 34
    - 5.2.1.2 Independente de Profundidade 36
  - 5.2.2 Medidas Baseadas em Semântica 38
  - 5.2.3 Matriz de Custo de Erro de Classificação 39
  - 5.2.4 Medidas Baseadas na Hierarquia 40
- 5.3 Erro Majoritário Hierárquico 42

6 Considerações Finais 44

A Uso das heurísticas para seleção dos conjuntos de treinamento locais 51
- A.1 Exemplo completo com a heurística *S-Inclusive* 55

Lista de Figuras

1 Representação do conjunto de classes. 3
2 Estrutura de representação da hierarquia de classes. 5
3 Tipos de predição. 7
4 Abordagem plana para classificação hierárquica 8
5 Método HBR. 11
6 Ilustração das 6 heurísticas baseadas em operações de conjuntos. 13
7 Heurística *Best Global* 16
8 Heurística *Best* η *Locals* 17
9 Abordagem local com o método HMR. 18
10 Preparação do conjunto de treinamento para o método HMR. . . . 20
11 Classificação de um novo exemplo: HBR + *top-down* + MLN 21
12 Classificação de um novo exemplo: HBR + *top-down* + NMLN 22
13 Modelo HMR com estratégia de multiplicação 25
14 Modelo híbrido com classificadores locais. 27
15 Classificação com modelo Híbrido + NMLN 28
16 Classificação com modelo Híbrido + MLN 29
17 Erros de classificação hierárquica. 33
18 Exemplo de hierarquia com pesos definidos 36
19 Erro majoritário hierárquico para predição MLN. 43
20 Erro majoritário hierárquico para predição NMLN. 44

Lista de Tabelas

1 Tabela de símbolos. xi
2 Tabela atributo-valor 3
3 Heurísticas para preparação dos conjuntos de treinamento 11
4 Heurísticas consideradas em trabalhos relacionados 18
5 Exemplo da preparação do conjunto de treinamento 19
6 Matriz de confusão 30
7 Matriz de custo de erro para predição MLN 39
8 Matriz de custo de erro para predição NMLN 40
9 Heurística *Exclusive* 51
10 Heurística *L-Exclusive* 52
11 Heurística *S-Exclusive* 52
12 Heurística *Inclusive* 53
13 Heurística *L-Inclusive* 53
14 Heurística *S-Inclusive* 54
15 Exemplo da aplicação da heurística *S-Inclusive* 55

Notação

Tabela 1: Tabela de símbolos.

Tr	Conjunto com todos os exemplos de treinamento
Te	Conjunto com todos os exemplos de teste
$Tr^{-}_{y_j}$	Conjunto com todos os exemplos negativos para a classe y_j
$Tr^{+}_{y_j}$	Conjunto com todos os exemplos positivos para a classe y_j
L	Conjunto dos rótulos conhecidos para o domínio do problema. Esses são os rótulos distintos dos exemplos do conjunto de treinamento
y_j	Rótulo referente à classe j, $y_j \in L$
$\circledast(E_i)$	Classe mais específica de um exemplo
$\odot(Y_i)$	Classe mais específica de um conjunto de classes
$\Uparrow y_j$	Conjunto das classes ancestrais de y_j
$\uparrow y_j$	A classe mãe de y_j
$\leftrightarrow y_j$	Conjunto das classes irmãs de y_j
$\downarrow y_j$	Conjunto das classes filhas de y_j
$\Downarrow y_j$	Conjuntos das classes descendentes de y_j
M	Número de atributos dos exemplos
X_j	Rótulo dos atributos, $j = 1...M$
$\mathbf{x}_i$	Vetor de atributos do exemplo E_i no formato $(x_{i1}, x_{i2}, ..., x_{iM})$
x_{ij}	Valor do atributo j do exemplo E_i, $j = 1...M$
E_i	Exemplo no formato $(\mathbf{x}_i, Y_i)$, $i = 1, 2, ...\|Tr\|$
Y_i	Conjunto de rótulos verdadeiros associados ao exemplo E_i no formato $\{y_{i1}, y_{i2}, ..., y_{iq}\}$, onde $q \geq 1$. Quando $q = 1$ o problema é monorrótulo, enquanto que para valores de $q \geq 2$ o problema é considerado multirrótulo, $Y_i \subseteq L$. Por exemplo, quando se trata de problema monorrótulo $\mathrm{Y}_i = \{business\}$ indica que o exemplo E_i pertence à classe $business$, ao passo que para classificação multirrótulo, Y_i poderia ser igual a $\{business, economy\}$
Z_i	Conjunto de rótulos preditos para o exemplo i
z_j	*j-esimo* rótulo predito para um determinado exemplo, $z_j \in L$
C_{y_j}	Classificador plano associado a classe y_j
Γ	Valor que representa um limiar inferior de teste (*Threshold*)
MC	Matriz de confusão
ϵ	Custo de um erro de classificação
ζ_{y_j}	Centróide para a classe y_j, calculado a partir da Equação 1

1 Introdução

Em aprendizado de máquina e mineração de dados, a maior parte dos trabalhos relacionados ao tema de classificação envolvem problemas de *classificação plana*, nos quais cada exemplo pode ser classificado como *monorrótulo* ou *multirrótulo*, sem que exista qualquer relacionamento hierárquico ou estrutural entre as diferentes classes do problema. No primeiro caso, cada exemplo pode estar associado a apenas um rótulo, ao passo que na classificação plana multirrótulo mais de um rótulo pode ser atribuído a cada exemplo.

No entanto, existem problemas de classificação mais complexos para os quais o conceito meta (classes) faz parte de uma hierarquia (Freitas e de Carvalho, 2007), como a tarefa de sequenciamento de proteínas do MIPS (Munich Information Centre for Protein Sequences)[1], cujas classes são organizadas em uma taxonomia na qual o nível mais geral consiste das classes *metabolismo*, *energia*, *transcrição* e *síntese de proteínas*. Cada uma dessas classes é subdividida em outras classes mais específicas, as quais, por sua vez, são novamente subdivididas, e assim sucessivamente até que a hierarquia complete 4 níveis (Clare e King, 2001). Outro exemplo, chamado de catálogo funcional (FunCat - Functional Catalogue) (Ruepp et al., 2004), é um esquema de anotação da descrição funcional de proteínas de diversos organismos vivos. Suas 1362 categorias funcionais, na versão 2.1 do FunCat, são organizadas em uma estrutura de árvore com até 6 níveis de especificidade.

Conjuntos de dados hierárquicos para mineração de textos também estão disponíveis. O MeSH (Lipscomb, 2000), por exemplo, com descrições de artigos médicos e o conjunto da ACM[2] com artigos relacionados a Ciências da Computação. Além desses, o *Dmoz Open Directory*[3] mantém uma hierarquia de classes conceituais para classificação de páginas *web*, as quais podem ser rotuladas com mais de uma classe em diferentes níveis da hierarquia.

Para esses casos, métodos de classificação plana são ineficazes, pois não são capazes de utilizar as informações de relacionamentos e dependências latentes à hierarquia de classes. Consequentemente, faz-se necessária a elaboração de outros métodos e algoritmos que sejam capazes de utilizar essa informação para a construção de modelos de classificação mais adequados e eficazes para a predição das classes de novos exemplos desse domínio. Essa abordagem de classificação tem recebido bastante atenção nos últimos anos e é um importante tema de pesquisa, sendo aplicada em tarefas de *Mineração de Textos*, *Web Mining* e *Bioinformática*.

O objetivo deste trabalho é explorar o tema de classificação hierárquica,

[1] http://mips.gsf.de/
[2] http://www.acm.org/
[3] http://www.dmoz.org/

apresentando as principais abordagens para a solução desse problema, suas características e métodos de avaliação. Além disso, são apontadas algumas questões que ainda estão em aberto com relação a esse tema de pesquisa e, também, direcionamentos para trabalhos futuros.

Este trabalho está organizado da seguinte maneira: na Seção 2 são apresentadas as principais características e diferenças entre os métodos de classificação plana e classificação hierárquica. A fundamentação teórica sobre a classificação hierárquica é apresentada na Seção 3, na qual são apresentadas, entre outros, as principais abordagens para implementação de algoritmos de classificação hierárquica. Na Seção 4, são apresentados dois métodos por nós propostos para a solução do problema de classificação hierárquica. As medidas de avaliação dos algoritmos de classificação hierárquica são descritas na Seção 5, enquanto que as considerações finais deste trabalho são apresentadas na Seção 6.

2 Classificação Plana *versus* Hierárquica

A principal diferença conceitual entre os métodos de classificação plana e de classificação hierárquica é a maneira como as classes do problema estão organizadas. No caso de classificação plana, em geral não existe um interrelacionamento direto entre as classes (monorrótulo), ou quando há algum tipo de relacionamento (multirrótulo), esse relacionamento não representa relações de super ou subclasse. Na classificação hierárquica, por outro lado, uma determinada classe pode ser a super classe de um subconjunto de classes, constituindo assim relação de generalização e especialização entre classes e uma estrutura hierárquica com diversos níveis de especificidade ou generalidade. Na Figura 1 é ilustrada essa diferença entre os dois tipos de classificação. Observe que na Figura 1(A) os rótulos são apresentados sem qualquer relacionamento, ao passo que na Figura 1(B) arestas que conectam pares de classes representam as relações estruturais. Os nós mais próximos da raiz da hierarquia representam classes mais gerais e os nós mais próximos das folhas representam classes mais específicas. Neste trabalho usaremos os termos *raso* e *profundo* como referência à profundidade dos nós na taxonomia de classes. Assim, quanto mais geral for a classe, mais raso será o nó, e quanto mais específica for a classe, mais profundo será o nó.

Conjuntos de dados para classificação hierárquica são usualmente representados de maneira semelhante aos conjuntos de exemplos para classificação plana, por meio da tabela *atributo-valor* (Tabela 2). No entanto, na classificação hierárquica, Y_i é o conjunto de rótulos associados ao exemplo E_i para os quais existem relacionamentos de especialização e generalização entre os rótulos

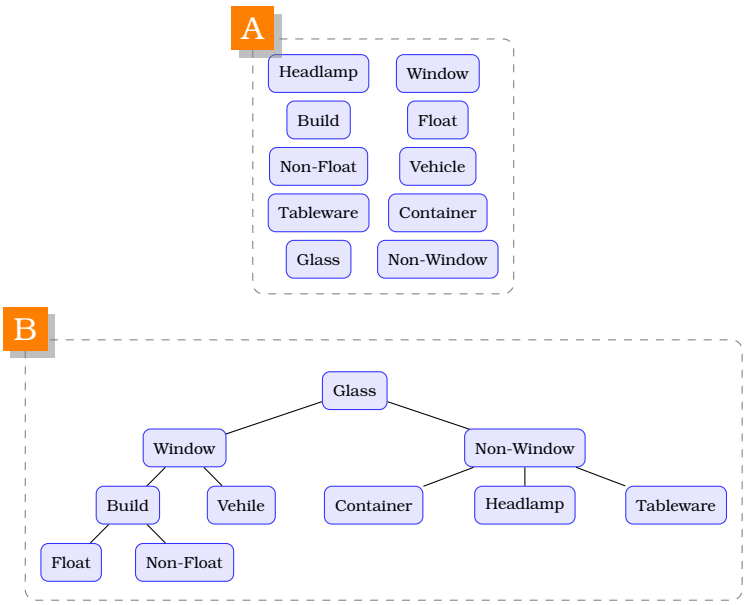


Figura 1: Representação do conjunto de classes. (A) classes sem qualquer tipo de organização estrutural e (B) classes hierarquicamente organizadas.

válidos de acordo com uma estrutura de classes previamente especificada (taxonomia de classes).

Considere como exemplo um conjunto de dados correspondente à classificação de exemplos em diferentes tipos de vidros, cuja hierarquia de classes corresponde à apresentada na Figura 1(B). Esse conjunto de dados está parcialmente representado na Tabela 2 e será utilizado para ilustrar os algoritmos e heurísticas apresentados nesse trabalho.

Tabela 2: Tabela atributo-valor com exemplos de classificação hierárquica.

	X_1	X_2	$\dots$	X_M	Y
E_1	x_{11}	x_{12}	$\dots$	x_{1M}	Glass→Window→Vehicle
E_2	x_{21}	x_{22}	$\dots$	x_{2M}	Glass→Non-Window→Container
E_3	x_{31}	x_{32}	$\dots$	x_{3M}	Glass→Non-Window→Headlamp
E_4	x_{41}	x_{42}	$\dots$	x_{4M}	Glass→Window→Build
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\dots$	x_{NM}	Glass→Window→Build→Float

3 Fundamentos e Definições para Classificação Hierárquica

Em alguns problemas de classificação hierárquica podem existir exemplos que tenham mais de um rótulo associado em cada nível da hierarquia de classes. Nesses casos, a tarefa é chamada de *classificação hierárquica multirrótulo*. Por outro lado, se cada exemplo pode estar associado a apenas um rótulo em cada nível da hierarquia, o problema é de *classificação hierárquica*

monorrótulo, ou simplesmente *classificação hierárquica*. Os métodos descritos neste trabalho classificam exemplos com apenas um rótulo por nível.

A organização das classes de maneira hierárquica implica que um exemplo não pode estar associado a uma classe y_j qualquer sem que esteja também associado com a classe mãe de y_j. Além disso, o conjunto de rótulos associados a cada exemplo deve formar um caminho com origem no nó raiz da hierarquia terminando no nó que representa a classe mais específica associada ao exemplo.

Desse modo, dada uma hierarquia de classes pré-definida, uma maneira bastante simples para construir um classificador hierárquico é transformar a tarefa em um ou mais problemas de classificação plana, construindo um ou mais classificadores planos em cada nível da hierarquia de classes. Outra alternativa é construir um classificador plano para cada nível da hierarquia. Na rotulação de um novo exemplo, esses classificadores são acionados seguindo uma abordagem do topo para baixo, *top-down*, e restringido a classificação em cada nível somente entre os classificadores filhos da classe que foi predita no nível anterior.

Outros métodos foram propostos para solucionar o problema de classificação hierárquica, alguns deles apresentados nas próximas seções, os quais podem ser categorizados considerando diversos aspectos (Freitas e de Carvalho, 2007; Sun e Lim, 2001):

a) a estrutura de dados utilizada para representar a hierarquia de classes e suas interrelações — Seção 3.1,

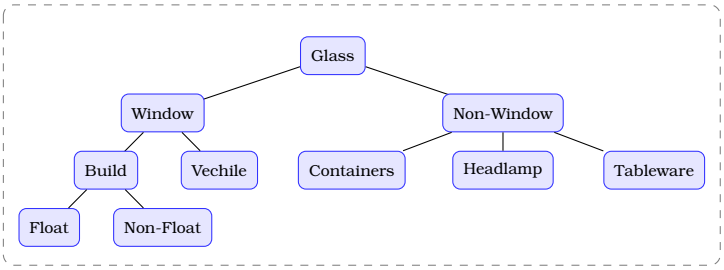
b) as restrições de nível para classificação — Seção 3.2,

c) as estratégias para o uso da informação/estrutura hierárquica de classes — Seção 3.3.

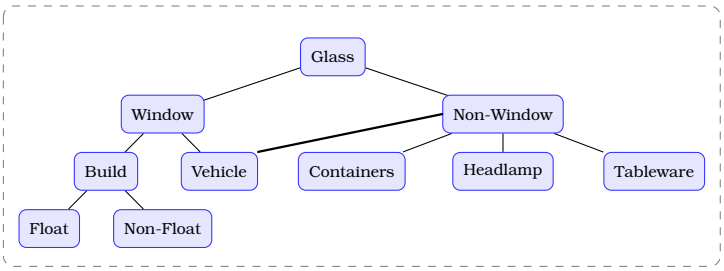
3.1 Estrutura de Representação da Hierarquia de Classes

A hierarquia de classes é representada usualmente por meio de estruturas de dados *árvores* ou de *grafos direcionados e acíclicos* (DAG — *Directed Acyclic Graph*), ilustradas na Figura 2. Em ambos os casos, os nós representam as classes a serem preditas enquanto que as arestas representam a relação de generalização e especialização das classes.

Neste trabalho restringimos o uso da representação a estruturas de árvores. Desse modo, a taxonomia de classes pode ser definida como uma árvore estruturada sobre um conjunto parcialmente ordenado $(L, \prec)$, onde L é um conjunto finito composto por todas as classes presentes na taxonomia e $\prec$ representa a relação *é uma* (*is-a*) assimétrica, anti-reflexiva e transitiva (Silla Jr e Freitas, 2011; Wu et al., 2005). Essas propriedades são definidas a seguir:



(a) Árvore



(b) Grafo

Figura 2: Estrutura de representação da hierarquia de classes.

- existe apenas uma classe ou nó chamado *raiz* da árvore e todas as outras classes são descendentes desta;
- assimétrica: $\forall y_i, y_j \in L, y_i \prec y_j \rightarrow y_j \not\prec y_i$;
- anti-reflexiva: $\forall y_i \in L, y_i \not\prec y_i$; e
- transitiva: $\forall y_i, y_j, y_k \in L, (y_i \prec y_j) \ \wedge \ (y_j \prec y_k) \ \rightarrow \ (y_i \prec y_k)$.

Além disso, existem relações entre nós em diferentes níveis da hierarquia, que são representadas por meio dos seguintes operadores:

- $\uparrow y_j$: classe mãe da classe y_j;
- $\Uparrow y_j$: conjunto de classes ancestrais da classe y_j;
- $\leftrightarrow y_j$: conjunto de classes irmãs da classe y_j;
- $\downarrow y_j$: conjunto de classes filhas de y_j;
- $\Downarrow y_j$: conjunto de classes descendentes da classe y_j, ou seja, todas as classes presentes na subárvore cuja raiz é y_j, exceto y_j.

É importante, também, distinguir exemplos explicitamente e implicitamente rotulados:

- *explicitamente rotulado*: quando um exemplo está explicitamente rotulado com a classe y_j, significa que y_j é a classe mais específica (mais

profunda) de todas as classes associadas a esse exemplo. As outras classes associadas a esse exemplo fazem parte do conjunto $\Uparrow y_j$. Utilizaremos a notação $\circledast(E_i) = y_j$ sempre que for necessário dizer que o exemplo E_i está explicitamente rotulado com a classe y_j.

- *implicitamente rotulado*: caso a classe y_j não seja a classe mais específica associada ao exemplo E_i, ou seja, $\circledast(E_i) = y_k$ tal que $y_k \in \{\Downarrow y_j\}$, esse exemplo estará implicitamente rotulado com a classe y_j. Em outras palavras, a classe mais específica do exemplo E_i é descendente da classe y_j. Nesse caso, a notação $\circledcirc(E_i)$ é utilizada para representar a classificação implícita do exemplo E_i, *i.e.*, representa o conjunto de classes associadas ao exemplo E_i exceto a classe mais específica desse exemplo.

Para facilitar a visualização das imagens relacionadas ao conjunto de exemplos utilizados para apresentar os algoritmos e heurísticas tratados neste trabalho — Tabela 2 — ,a partir deste ponto utilizaremos apenas a primeira letra de cada rótulo (nome da classe) para identificar os nós da taxonomia referente aos diferentes tipos de vidro. Além disso, para evitar ambiguidade como nos casos de Window e Non-Window, ou Float e Non-Float, a letra N será utilizada como prefixo. Assim, a taxonomia apresentada na Figura 2(a) é representada pelo conjunto de rótulos $L = \{\mathrm{G}, \mathrm{W}, \mathrm{B}, \mathrm{F}, \mathrm{NF}, \mathrm{V}, \mathrm{NW}, \mathrm{C}, \mathrm{H}, \mathrm{T}\}$.

3.2 Restrição no Nível de Classificação

Os métodos de classificação hierárquica diferenciam-se, também, pela maneira como a predição de um novo exemplo é obtida. São duas estratégias:

1. a classe mais específica é obrigatoriamente uma classe-folha;

2. a classe mais específica pode ser uma classe interna da taxonomia.

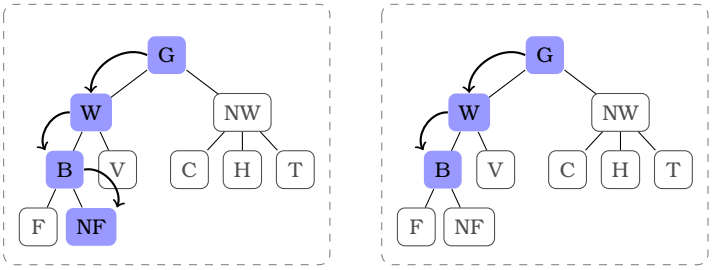
Muitas vezes a escolha de como é feita a classificação com relação a esse quesito (obrigatoriamente até as folhas ou não) depende das características do conjunto de dados.

Essas duas abordagem são definidas em (Sun et al., 2003a) e são nomeadas, respectivamente, *árvore (ou grafo) virtual de categoria*[4] e *árvore (ou grafo) real de categoria*[5]. No entanto, neste trabalho usaremos a nomenclatura proposta em (Freitas e de Carvalho, 2007), na qual os termos utilizados são, respectivamente, *predição obrigatória nos nós-folha* (MLN — *Mandatory Leaf Node-Prediction*) e *predição opicional nos nós-folha* (NMLN — *Non-mandatory*

[4] *Virtual category tree (or DAG).*
[5] *Real category tree (or DAG).*

Leaf Node Prediction). Na Figura 3 é ilustrada a diferença entre as duas estratégias para predição.

Para o caso MLN, ilustrado na Figura 3(a), o exemplo foi classificado no primeiro nível como classe W, no segundo nível como classe B e, finalmente, no terceiro nível como classe NF, a qual é representada por um nó-folha na hierarquia de classes. No caso NMLN, a classe predita mais específica é B, sendo que no nível anterior a classe W foi predita. Observe que nesse caso nenhuma das classes, F ou NF foi predita.



(a) MLN (b) NMLN

Figura 3: Tipos de predição.

É importante mencionar que predições feitas nos nós internos são, em geral, mais confiáveis que as predições nos nós-folha. Esse comportamento é devido à maior especificidade das classes-folha e, também, devido ao menor número de exemplos associados a estas classes em comparação com as classes internas na hierarquia. Por outro lado, a predição nas classes-folha são em geral mais úteis para o usuário (Freitas e de Carvalho, 2007).

3.3 Uso da Estrutura de Classes

O terceiro critério utilizado para diferenciar os métodos de classificação hierárquica é a maneira como a estrutura hierárquica é utilizada para a construção do classificador e para a classificação de novos exemplos:

- plana (*naïve*);
- global; e
- local.

As três abordagens são descritas em maiores detalhes a seguir.

3.3.1 Abordagem Plana

Uma maneira simples para resolver o problema de classificação hierárquica é transformá-lo em um problema de classificação plana monorrótulo e, então, aplicar algum dos diversos algoritmos disponíveis para essa tarefa.

Essa abordagem, referenciada como *abordagem direta* (*Direct Approach*) por Burred e Lerch (2003) e global por Xiao et al. (2007), ignora completamente a estrutura hierárquica do problema durante o treinamento, construindo um modelo de classificação que considera apenas as classes-folha da hierarquia, conforme ilustrado na Figura 4. Com a classe predita por meio desse modelo, obtêm-se as classes ancestrais seguindo o caminho a partir da classe predita até a raiz da estrutura hierárquica.

É importante lembrar que o termo global utilizado por Xiao et al. (2007) não deve ser confundido com o sentido de "global" em classificação verdadeiramente hierárquica, como discutido posteriormente na Seção 3.3.2.

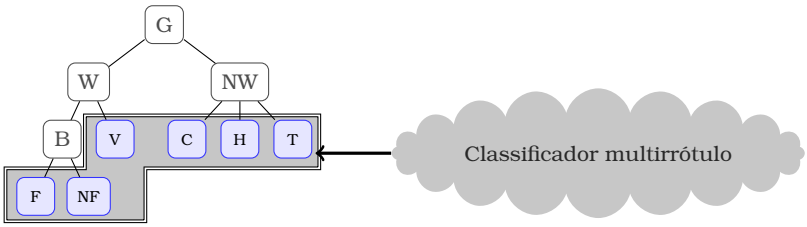


Figura 4: Classificação hierárquica transformada em um problema de classificação plana multiclasse.

Observe que nessa abordagem a estrutura hierárquica é sempre virtual, *i.e.*, a classe mais específica da predição dada a cada novo exemplo é obrigatoriamente um nó-folha. Barbedo e Lopes (2007) utilizaram o termo *bottom-up* para descrever essa abordagem, uma vez que um novo exemplo é classificado inicialmente de acordo com as classes "folhas" e as classes correspondentes aos nós internos da hierarquia são consequências dessa classificação inicial, seguindo uma abordagem de baixo para cima para a identificação das classes intermediárias (implicitamente anotadas).

Como desvantagem dessa abordagem observa-se o fato de não considerar as informações das relações entre as classes da hierarquia, e ainda assim, ter que discriminar entre um grande número de classes (todas as classes-folha).

Em outras palavras, utilizando essa abordagem, não há distinção entre o treinamento e teste da classificação hierárquica e a classificação plana monorrótulo. A única diferença é a maneira como a predição do modelo é utilizada, uma vez que infere-se indiretamente o conjunto de classes ancestrais a partir da classe predita.

3.3.2 Abordagem Global

Nessa abordagem, também conhecida como *Big-bang*, um único classificador é induzido e é capaz de rotular novos exemplos considerando toda a hierarquia de classes. O modelo construído por um algoritmo global é geralmente mais complexo que o gerado por outras abordagens. No entanto, apenas um

modelo é gerado, enquanto que em outras abordagens diversos modelos são construídos (ver abordagem local). Além disso, as dependências entre as diferentes classes da hierarquia, como as relações de generalização/especialização, são consideradas de maneira natural no processo de construção do modelo de classificação global (Blockeel et al., 2002).

No entanto, há ainda a necessidade de uma definição mais clara para diferenciar os algoritmos globais de outros tipos de algoritmos de classificação hierárquica, uma vez que não existe um consenso da comunidade científica a esse respeito. Segundo Silla Jr e Freitas (2011), essa distinção é feita usualmente por meio do critério de exclusão. Assim, qualquer algoritmo que não pertencer às abordagens *plana* ou *local*, é classificado como *global*. Além disso, os autores afirmam que embora aparentemente não existam características específicas que sejam compartilhadas por todos os algoritmos de classificação hierárquica global, em geral, esses algoritmos possuem duas características principais: (1) consideram toda a hierarquia de classes durante o treinamento do classificador; e (2) não há qualquer tipo de modularidade ou uso de informação local, como na abordagem *local*.

Por outro lado, é possível considerar que a principal diferença entre a abordagem global e as outras abordagens está relacionada à fase de treinamento, pois, como mencionado anteriormente, algoritmos globais consideram todas as relações estruturais da hierarquia de classe durante a fase de treinamento. Já durante o teste de novos exemplos, mesmo algoritmos globais podem fazer uso de estratégias de predição que não sejam globais, por exemplo a estratégia de predição *top-down*, descrita na Seção 3.3.4.

Outra característica marcante dessa abordagem é a pouca quantidade de trabalhos publicados em relação à abordagem *local*. Um fator determinante talvez seja a complexidade para a elaboração de algoritmos globais. Ainda assim, alguns algoritmos foram propostos, como os apresentados em (Cai e Hofmann, 2007; Vens et al., 2008; Qiu et al., 2009; Wang et al., 2009).

3.3.3 Abordagem Local

Assim como a abordagem plana, a abordagem local também transforma o problema de classificação hierárquica em problemas mais simples e modulares para os quais já existem soluções amplamente testadas e validadas. Contudo, ao contrário da abordagem plana, nesta abordagem a estrutura hierárquica das classes não é completamente ignorada, uma vez que, além das classes-folha, as outras classes também são utilizadas nas fases de treinamento e teste.

Nessa abordagem um ou mais classificadores independentes são construídos em cada nível da hierarquia de classes utilizando apenas a informação

local da taxonomia e do conjunto de exemplos de treinamento, como o conjunto de exemplos associados à classe que o classificador representa. Além disso, cada classificador local "não tem" conhecimento da existência dos outros classificadores locais.

A característica modular dessa abordagem é utilizada tanto na construção dos classificadores locais como na fase de classificação de novos exemplos, na qual, aplicando-se uma estratégia de predição *top-down* (ver Seção 3.3.4), a classificação de um exemplo em um determinado nível pode guiar a predição no nível seguinte. Em outras palavras, cada exemplo é apresentado apenas aos classificadores filhos da classe a qual foi associado em um nível anterior.

Em termos de nomenclatura, essa abordagem tem sido amplamente referenciada na literatura como abordagem *top-down* (Mladenić e Grobelnik, 1998; Sun e Lim, 2001; Sun et al., 2003b, 2004; Tikk et al., 2004; Barutcuoglu et al., 2006; Secker et al., 2007, 2010). Contudo, alguns autores consideram o termo *top-down* mais apropriado para a definição da estratégia de predição uma vez que esta é realizada nível a nível a partir da raiz até as folhas (Freitas e de Carvalho, 2007; Silla Jr e Freitas, 2011; Metz e Freitas, 2009). Em outras palavras, o termo *top-down* descreve basicamente como os classificadores *locais* são utilizados para classificar novos exemplos. Nesse sentido, neste trabalho o termo *local* é utilizado como referência a essa abordagem para construção de classificadores hierárquicos.

São basicamente três os métodos utilizados para a construção dos classificadores locais:

1. construir um classificador binário para cada classe da hierarquia (exceto o nó raiz). Neste trabalho utilizaremos o termo *Hierarchical Binary Relevance* (HBR) como referência a esse método, conforme proposto por Tsoumakas et al. (2010);

2. construir um classificador multiclasse para cada classe não-folha da hierarquia, chamado neste trabalho de *Hierarchical Multiclass Relevance* (HMR); e

3. construir um classificador multiclasse para cada nível da hierarquia, chamado neste trabalho de *Hierarchical Multiclass Per Level* (HMPL).

O modelo HMPL é mais suscetível a inconsistências e consequentemente menos utilizado. Devido a esse fato, neste trabalho nos concentraremos nos outros dois métodos, ambos apresentados nas próximas seções.

3.3.3.1 Treinamento do modelo HBR

Esse método, ilustrado na Figura 5, consiste em construir um classificador local binário C_{y_j} para cada classe y_j da taxonomia, com exceção do nó

raiz, usando um conjunto de exemplos positivos ($Tr^+_{y_j}$, os exemplos que representam a classe y_j) e um conjunto de exemplos negativos ($Tr^-_{y_j}$, os exemplos que não estão associados a classe y_j). A união desses dois conjuntos, $Tr^+_{y_j}$ e $Tr^-_{y_j}$, resulta no conjunto de treinamento local utilizado para a construção do classificador C_{y_j}, *i.e.*, $Tr_{y_j} = Tr^+_{y_j} \cup Tr^-_{y_j}$.

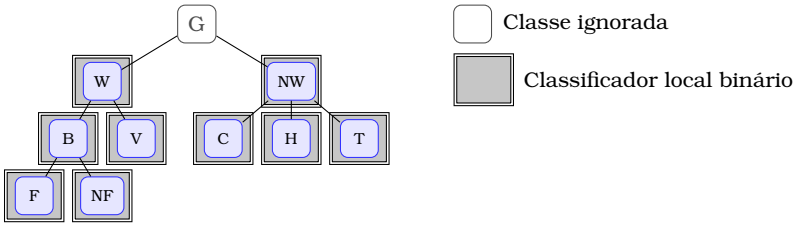


Figura 5: Método HBR.

Para encontrar os conjuntos de treinamento locais relativos a cada classe y_j a partir dos exemplos de treinamento global (Tr), oito heurísticas foram propostas, as quais são detalhadas na Tabela 3. Dessas heurísticas, seis são baseadas em *operações de conjuntos* aplicadas sobre subconjuntos de exemplos considerando a taxonomia de classes e o rótulo da classe mais específica atribuído a cada exemplo presente em Tr, *i.e.* $\circledast(E_i)$. Além disso, essas seis heurísticas fazem uso do operador $*$ que, nesse caso, é aplicado a um conjunto de classes e resulta os exemplos em Tr que têm como classes mais específica uma dentre as classes especificadas como parâmetro. As outras duas heurísticas (*Best Global* and *Best* η *Locals*) são baseadas em *distância* entre exemplos.

Tabela 3: Heurísticas para preparação do conjunto de treinamento dos classificadores locais binários.

Heurística	$Tr^+_{y_j}$	$Tr^-_{y_j}$	Referência
All (*L-Inclusive*)	$*(y_j \cup \Downarrow y_j)$	$Tr - Tr^+_{y_j}$	Mladenić e Grobelnik (1998)
Hierarchical training set (*S-Inclusive* §)	$*(y_j \cup \Downarrow y_j)$	$*(\leftrightarrow y_j \cup \Downarrow (\leftrightarrow y_j))$	Ceci e Malerba (2003)
Proper training set (*S-Exclusive* §)	$*(y_j)$	$*(\leftrightarrow y_j)$	Ceci e Malerba (2003)
Inclusive	$*(y_j \cup \Downarrow y_j)$	$Tr - (Tr^+_{y_j} \cup *(\Uparrow y_j))$	Eisner et al. (2005)
Exclusive	$*(y_j)$	$Tr - Tr^+_{y_j}$	Eisner et al. (2005)
L-Exclusive	$*(y_j)$	$Tr - *(y_j \cup \Downarrow y_j)$	Eisner et al. (2005)
Best Global	$*(y_j \cup \Downarrow y_j)$	$Rank^\eta(Tr - Tr^+_{y_j}, dist(E_i, \zeta_{y_j}))$	Fagni e Sebastiani (2007)
Best η *Locals*	$*(y_j \cup \Downarrow y_j)$	$Rank^\eta(Tr - Tr^+_{y_j}, distKnn(E_i))^6$	Fagni e Sebastiani (2007)

O símbolo § associado a duas heurísticas representa *siblings*, ou *irmãos* em Português.

De maneira intuitiva, a heurística *All* (Mladenić e Grobelnik, 1998) faz uso da informação hierárquica da taxonomia de classes para selecionar os exemplos para o conjunto de treinamento local Tr_{y_j}. Assim, exemplos explicitamente rotulados com a classe y_j ou qualquer das classes descendentes de y_j, *i.e.* todos os exemplos E_i tal que $\circledast(E_i) \in \{y_i \cup \Downarrow y_j\}$ são inclusos no conjunto $Tr^+_{y_j}$, uma vez que por definição $\forall y_k \in (\Downarrow y_j), y_k \prec y_j$. Todos os outros exemplos são utilizados como exemplos negativos, ou seja, colocados em $Tr^-_{y_j}$. Desse modo, $Tr^-_{y_j} = Tr - Tr^+_{y_j}$. Essa heurística foi também utilizada em (Eisner et al.,

2005), no qual foi chamada de *Less-Inclusive*, termo que abreviamos neste trabalho para *L-Inclusive*.

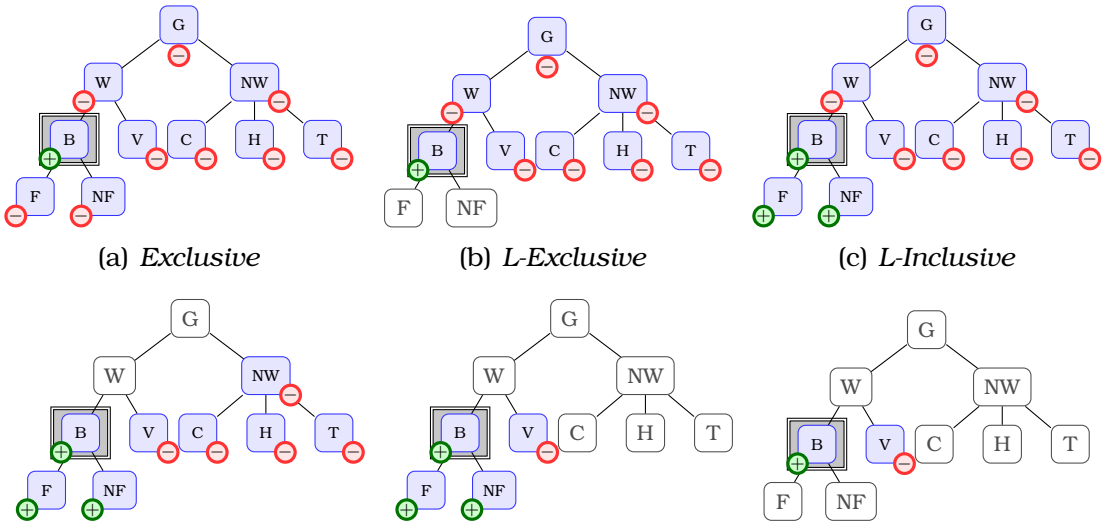
Outras maneiras de utilizar a informação estrutural da taxonomia de classes podem ser empregadas. Em (Eisner et al., 2005) são apresentados os conceitos de classificadores *exclusivos* e *inclusivos*. Esses conceitos podem ser utilizados para ajudar na interpretação das heurísticas baseadas em operações de conjunto. Segundo Eisner et al. (2005), um classificador é considerado *exclusivo* se, dado um novo exemplo que deveria ser explicitamente rotulado com a classe y_j, apenas o classificador local C_{y_j} dentre todos os classificadores locais construídos na hierarquia, prediz o novo exemplo como positivo. Assim, cada classificador C_{y_j} exclui (rejeita) todo exemplo que não seja explicitamente rotulado com a classe y_j. Nesse sentido, a classificação final de um exemplo para as classes ancestrais da classe mais específica é implicita, ou seja, encontrada a partir do caminho da classe explicitamente atribuída ao exemplo até a raiz da taxonomia de classes.

Por outro lado, quando um exemplo exclusivamente rotulado com a classe y_j é apresentado aos classificadores locais e todos os classificadores associados às classes $\{y_j \cup \Uparrow y_j\}$ predizem positivo para esse exemplo, esses classificadores são considerados *inclusivos*. Desse modo, a predição dos classificadores *ancestrais* é também considerada (inclusa) explicitamente na classificação final do exemplo apresentado.

A partir dos conceitos *exclusivo* e *inclusivo* podemos derivar heurísticas que consideram mais (inclusivas) ou menos (exclusivos) exemplos para a seleção dos conjuntos de treinamento locais.

As seis heurísticas baseadas em operações de conjunto são muito similares. Primeiramente os exemplos positivos ($Tr^+_{y_j}$) do conjunto de treinamento local são selecionados, considerando a classe mais específica de cada exemplo em Tr. Após, o conjunto $Tr^-_{y_j}$ é selecionado por meio da aplicação de operações matemáticas sobre os conjuntos Tr e $Tr^+_{y_j}$. A principal diferença entre essas seis heurísticas está relacionada ao tipo de operação utilizada e se essa operação é aplicada sobre todos os exemplos que não estão em $Tr^+_{y_j}$ ou a um subconjunto dele. Essas heurísticas são ilustradas na Figura 6, na qual a classe relevante para a construção dos conjuntos de treinamento locais é B, ou seja, o objetivo é selecionar o conjunto de treinamento para a construção do classificador C_{B}. Cada classe participa do processo assumindo diferentes papéis que dependem da heurística aplicada. Assim, uma classe pode determinar que o exemplo explicitamente associado a ela seja por vezes positivo ($\oplus$), negativo ($\ominus$) ou não considerado (nós com contorno pontilhado).

A seguir são apresentadas algumas observações com relação às heurísticas apresentadas.



(a) *Exclusive* (b) *L-Exclusive* (c) *L-Inclusive*

(d) *Inclusive* (e) *S-Inclusive* (f) *S-Exclusive*

Figura 6: Ilustração das 6 heurísticas baseadas em operações de conjuntos.

(a) não consideram completamente a hierarquia de classes. De fato, as heurísticas *Exclusive* e *S-Exclusive* ignoram completamente as relações de generalização e especialização de classes;

(b) a atribuição do rótulo negativo para os exemplos que fazem parte do conjunto $\Downarrow y_j$ (*Exclusive*) acarreta erro, uma vez que exemplos rotulados com qualquer uma das classes descendentes de y_j necessariamente fazem parte da classe y_j (relação *é-uma*);

(c) a atribuição do rótulo negativo para os exemplos que fazem parte do conjunto $\Uparrow y_j$ (*Exclusive* e *L-Exclusive*) também pode degradar a performance dos classificadores locais, pois diminui a capacidade de generalização dos mesmos;

(d) quando o conjunto de treinamento global contém apenas exemplos explicitamente rotulados com classes-folha ($Tr = \{E_i\}$, tal que $\circledast E_i = y_j$, sendo y_j uma classe-folha), os conjuntos de treinamento locais utilizados para a construção dos classificadores referente às classes mais rasas não terão exemplos positivos para o treinamento (*Exclusive*, *S-Exclusive* e *L-Exclusive*). Consequentemente, a performance do classificador final será degradada, uma vez que não será possível classificar de maneira fidedigna os novos exemplos nos níveis mais rasos da hierarquia de classes. No caso da heurística *S-Exclusive* pode ser ainda pior, pois nem mesmo exemplos negativos serão inclusos no conjunto de treinamento, não sendo possível a criação do classificador local;

(e) as heurísticas *inclusivas*, por outro lado, consideram um maior número de exemplos como positivos para cada classificador em comparação com as

heurísticas *Exclusive*, *L-Exclusive* e *S-Exclusive*, uma vez que não apenas os exemplos com classe mais específica igual a y_j são utilizados como positivos para o classificador C_{y_j}, mas também os exemplos com classe mais específica sendo uma das classes descendentes de y_j. Em outras palavras, faz uso das relações estruturais da taxonomia de classes. Consequentemente, mais informação hierárquica é utilizada no processo de treinamento dos classificadores locais;

(f) as heurísticas *Inclusive* e *S-Inclusive* não excluem exemplos das classes ancestrais, tornando o conjunto de treinamento local mais consistente com a taxonomia de classes; e

(g) as heurísticas que consideram apenas as sub-árvores irmãs para a seleção dos exemplos negativos (*S-Exclusive* e *S-Inclusive*), consideram que esses exemplos, apesar de não serem positivos, são os mais correlacionados com os exemplos positivos. Assim, a idéia básica é incluir em $Tr^-_{y_j}$ os exemplos que são considerados "quase positivos" para a classe y_j (Schapire et al., 1998). Esses seriam os exemplos que estão mais próximos da região de fronteira entre os exemplos positivos e não positivos e, por isso, são os mais informativos e podem ser utilizados como negativos para a construção do classificador local. Essa estratégia também é benéfica sob o ponto de vista de eficiência computacional, uma vez que um menor número de exemplos no conjunto de treinamento resulta em menor tempo de processamento para construir o classificador.

É importante observar que para todas as heurísticas apresentadas até então, há um grande risco de que os classificadores sejam treinados com conjuntos de exemplos bastante desbalanceados, uma vez que o número de exemplos positivos é, em geral, menor que o número de exemplos negativos. Para entender melhor, basta imaginarmos uma situação na qual existe um número igual de exemplos explicitamente rotulados com cada classe da taxonomia. Nesse contexto, o conjunto $Tr^+_{y_j}$ é selecionado a partir de um número menor de classes que o conjunto $Tr^-_{y_j}$. Consequentemente, o número resultante de exemplos positivos é menor que o de exemplos negativos. Esse risco de seleção de conjuntos de treinamentos locais desbalanceados pode ser bastante reduzido como uso da heurística *S-Exclusive*. No entanto, como observado anteriormente, as heurísticas *exclusivas* apresentam diversos problemas. Assim, um boa alternativa pode ser o uso da heurística *S-Inclusive*. Por exemplo, se o número de exemplos associados a cada classe é o mesmo, a distribuição de exemplos para um caso no qual existem apenas duas classes em um determinado nível e o número de classes descendentes também é o mesmo para cada sub-árvore, a distribuição de exemplos nessas classes será de 50% para cada,

resultando em um perfeito balanceamento.

Nas heurísticas baseadas em *distância* (*Best Global* and *Best* η *Locals*) (Fagni e Sebastiani, 2007) também está embutido o conceito de "quase positivo", mais especificamente a estratégia de seleção por *"query-zoning"* descrita em (Singhal et al., 1997). Segundo essa estratégia, em um modelo espacial, uma *query zone* pode ser representada pelo volume de um vetor de características. Assim, no contexto de classificação, uma *query zone* para um exemplo ou conjunto de exemplos, pode ser simulada por um conjunto de exemplos de treinamento que apresentam alguma similaridade com o exemplo de interesse. Essa similaridade pode, por sua vez, ser simulada pela distância entre exemplos. A partir dessa medida de "similaridade" pode-se identificar uma região de fronteira nesse espaço entre dois grupos de exemplos. No caso de classificação, podemos assumir que um grupo seja o conjunto de exemplos positivos para uma determinada classe y_j, enquanto o outro grupo pode ser o conjunto de exemplos negativos para essa classe. Nesse sentido, essa estratégia pode ser utilizada para seleção de exemplos negativos a partir do conjunto de exemplos não positivos com intuito de selecionar os exemplos mais informativos possíveis, ou seja, os exemplos mais próximos na região de fronteira entre as duas classes, que permitiriam a construção de classificadores com maior poder de predição.

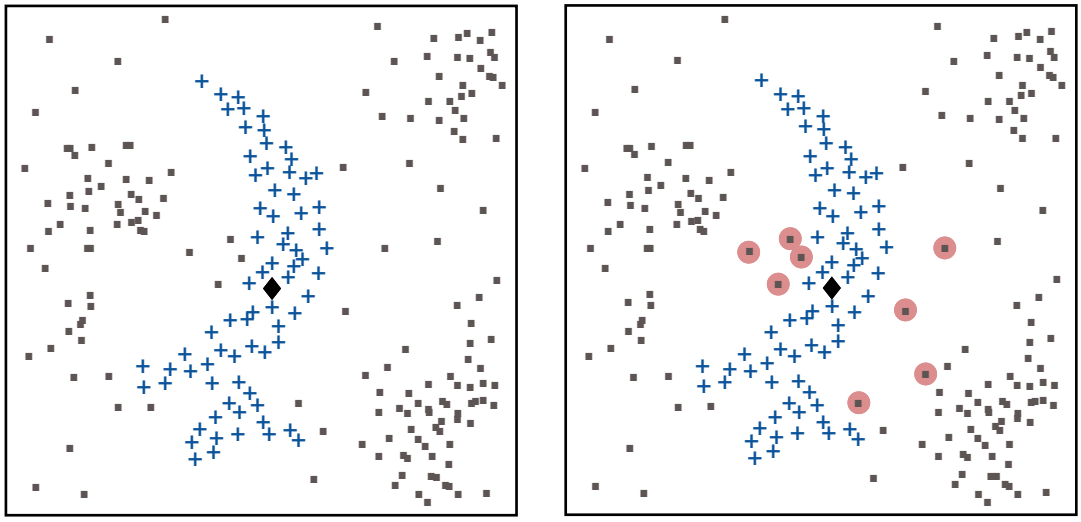
Para as heurísticas baseadas em *distância*, o conjunto $Tr^+_{y_j}$ é composto pelos exemplos com a classe mais específica sendo uma dentre as classes $\{y_j \ \cup (\Downarrow y_j)\}$. Já o conjunto $Tr^-_{y_j}$ depende da heurística aplicada.

A heurística *Best Global* inicialmente calcula o centróide do conjunto $Tr^+_{y_j}$ (Equação 1). Após, calcula a distância de cada exemplo não positivo até o centróide. Com esses valores de distância, essa heurística inclui em $Tr^-_{y_j}$ os η exemplos mais próximos do centróide.

$$\zeta_{y_j} = \frac{1}{|Tr^+_{y_j}|} \sum_{i=1}^{|Tr^+_{y_j}|} E_i; \ E_i \in \ Tr^+_{y_j} \tag{1}$$

Para ilustrar o uso dessa heurística, considere a Figura 7(a), na qual são representados no espaço bi-dimensional todos os exemplos no conjunto de treinamento global (Tr). Suponha que os exemplos positivos, selecionados por meio da regra $*(y_j \cup \Downarrow y_j)$, sejam os pontos representados pelo símbolo "+" e que o respectivo centróide está representado pelo símbolo ♦. Os exemplos não positivos são representados pelos outros pontos. Considerando $\eta = 8$ (8 exemplos negativos deverão ser selecionados), os 8 exemplos não positivos mais próximos do contróide selecionados para comporem o conjunto $Tr^-_{y_j}$ são mostrados em destaque na Figura 7(b).

Quando apenas um centróide é utilizado para selecionar os exemplos ne-



(a) Primeiro passo (b) Segundo passo

Figura 7: Ilustração da heurística *Best Global* para definição do conjunto de treinamento para um classificador local.

gativos, é possível que exemplos importantes não sejam selecionados e outros menos importantes sejam inclusos no conjunto de exemplos negativos. Se a classe y_j é linearmente separável das demais, a estratégia com o uso do centróide (*Best Global*) pode ser uma opção ótima. Porém, para classes não separáveis linearmente, a heurística *Best Global* resulta em um conjunto de exemplos negativos com exemplos que, apesar de próximos do centróide, podem estar de fato distantes da região de fronteira entre as classes.

O conceito de "quase positivo" também é usado na heurística *Best* η *Locals*. No entanto, nesse caso o objetivo é encontrar exemplos que estejam em uma região de fronteira mais complexa e que represente o "formato" da classe positiva. Para isso, não apenas um ponto de referência (centróide) é utilizado para selecionar os exemplo negativos, mas um pequeno subconjunto de k exemplos positivos.

Assim, para cada exemplo não positivo identifica-se inicialmente os k-vizinhos positivos mais próximos a partir dos exemplos positivos, *i.e.* $Knn(E_i, Tr^+_{y_j}, k)$, segundo alguma medida de distância pre-definida. Após, constrói-se um *ranking* com os exemplos não positivos organizados em ordem crescente em função da soma das distâncias entre eles e os seus respectivos k vizinhos positivos mais próximos — Equação 2. A partir desse *ranking*, seleciona-se os η primeiros exemplos — Equação 3.

$$distKnn(E_i) = \sum_{i=1}^{k} dist(E_i, E_j) \; \forall E_j \in Knn(E_i, Tr^+_{y_j}, k) \tag{2}$$

$$Rank^{\eta}(S, f(.)) \tag{3}$$

onde o método $Rank^{\eta}/2$ ordena o conjunto S de acordo com o resultado da função f aplicada a cada elemento do conjunto S. Após, retorna os η elementos melhores colocados no ranking. Para o caso da heurística *Best* η *Locals* $S = Tr - Tr^{+}_{y_j}$ e $f = distKnn(E_i)\ \forall E_i \in S$.

Como ilustração considere a Figura 8, na qual os valores $k = 3$ e $\eta = 15$ foram utilizados. Observe que os exemplos destacados com o círculo foram selecionados para compor o conjunto de exemplos negativos. Além disso, para um desses exemplos são indicados por meio de setas ($\rightarrow$) os seus k-vizinhos positivos mais próximos.

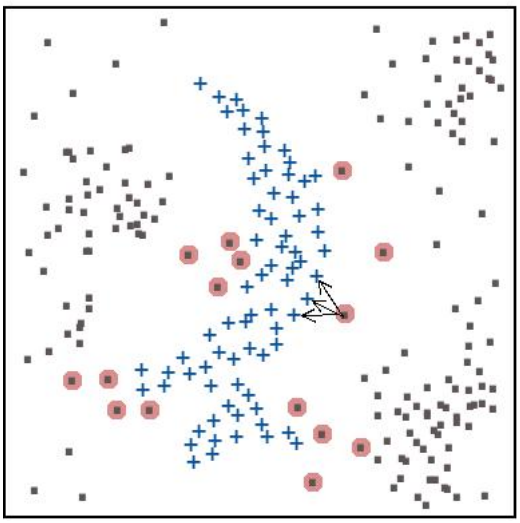


Figura 8: Ilustração da heurística *Best* η *Locals* para definição do conjunto de treinamento para um classificador.

Deve ser observado que a performance dos algoritmos pode ser bastante influenciada pela escolha da heurística, uma vez que há diferenças substanciais na maneira como cada heurística prepara os conjuntos de treinamento locais. Avaliações realizadas sobre o impacto que essas heurísticas causam no resultado final da classificação hierárquica, mostram que quanto mais inclusiva for a heurística, melhor será a performance do classificador em termos de qualidade da predição (Eisner et al., 2005).

Apesar de alguns pesquisadores reportarem resultados de análises do impacto dessas heurísticas na performance global de classificadores HBR, não é do nosso conhecimento algum estudo com resultados experimentais conclusivos comparando todas essas heurísticas em diversos domínios e/ou conjuntos de dados. Os resultados encontrados em alguns trabalhos são sumarizados na Tabela 4.

3.3.3.2 Treinamento do modelo HMR

Nesse modelo, referenciado como *top-down* em diversos trabalhos (Greiner et al., 1997; Sun et al., 2003a; Vens et al., 2008), para cada nó não-folha da

Tabela 4: Heurísticas consideradas em trabalhos relacionados (▲ melhor desempenho, e ▼ pior desempenho).

Referência	Heurística							
	Exclusive	*L-Exclusive*	*L-Inclusive*	*Inclusive*	*S-Inclusive*	*S-Exclusive*	*Best Global*	*Best η Locals*
Ceci e Malerba (2003)					▲	▼		
Eisner et al. (2005)	▼	▼	▲	▲				
Fagni e Sebastiani (2007)			▲		▲		▼	▼

hierarquia de classes, é construído um classificador multiclasse cuja tarefa é rotular novos exemplos de acordo com suas subclasses, *i.e.*, discriminar apenas entre suas classes filhas. Esse modelo é ilustrado na Figura 9, na qual o classificador associado à classe G, C_{G}, por exemplo, é treinado para predizer as classes W ou NW. O classificador C_{W}, por sua vez, classifica exemplos em B ou V. Os outros classificadores (C_{B} e C_{NW}), são utilizados de maneira análoga, sempre classificando novos exemplos em uma de suas respectivas classes filhas.

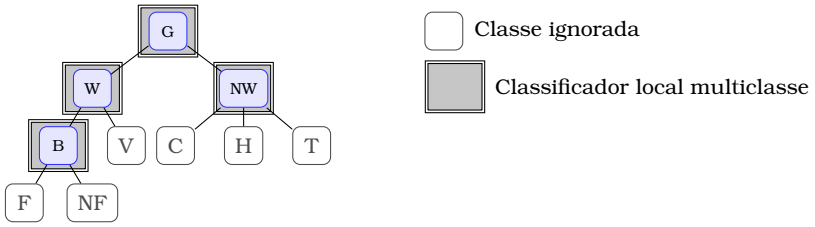


Figura 9: Abordagem local com o método HMR.

Assim como no modelo HBR, nesse modelo cada classificador local é construído a partir de um conjunto de exemplos de treinamento local. Uma possível heurística para a seleção dos exemplos que compõem esses conjuntos de treinamento pode ser: $Tr_{y_j} = *(\Downarrow y_j)$. Nesse caso os conjuntos de exemplos são multiclasse, uma vez que os classificadores a serem construídos são multiclasse.

Cada conjunto de treinamento local deve ser preparado de modo que os exemplos nele inclusos sejam rotulados apenas com as classes que esse classificador deverá discriminar. Desse modo, é necessário que o rótulo de cada exemplo inserido no conjunto de treinamento selecionado para a construção do classificador local C_{y_j} seja modificado de modo que apenas os rótulos referentes às classes filhas de y_j estejam presentes. Para isso, todos os exemplos previamente selecionados, cuja classe mais específica não está no conjunto $\downarrow y_j$, devem ter seu rótulo generalizado segundo a relação $\prec$. Em outras palavras, todo exemplo E_i tal que $\circledast E_i \in \{\Downarrow y_j\}$ previamente selecionado, passa por um processo de generalização de classe de modo que que $\circledast E_i \in \{\downarrow y_j\}$ seja verdade.

Considere como exemplo a preparação do conjunto de treinamento local para a construção do classificador C_{G}. Todos os exemplos $\circledast E_i \in \{\text{F}, \text{NF}, \text{B}, \text{V}\}$

serão generalizados de modo que sua classe mais específica seja W, ou seja, $\circledast E_i = \text{W}$.

Como ilustração desse processo, considere o conjunto de exemplos apresentado na Tabela 5, na qual são mostrados os exemplos selecionados no conjunto de treinamento para construir o classificador C_{W}. Na Figura 10 é apresentado um esquema gráfico de como as classes são consideradas para a criação dos conjuntos de treinamento para construir cada classificador local considerando a hierarquia apresentada na Figura 9.

Tabela 5: Exemplo da preparação do conjunto de treinamento para construir o classificador local C_{W}.

E_i	Rótulos verdadeiros	Selecionado para C_{W}	Rótulo generalizado
E_1	G →W		
E_2	G →W →B	✓	B
E_3	G →W →B	✓	B
E_4	G →W → V	✓	V
E_5	G →W → V	✓	V
E_6	G →W → V	✓	V
E_7	G →W → V	✓	V
E_8	G →W →B →F	✓	B
E_9	G →W →B →F	✓	B
E_{10}	G →W →B →F	✓	B
E_{11}	G →W →B →F	✓	B
E_{12}	G →W →B →NF	✓	B
E_{13}	G →W →B →NF	✓	B
E_{14}	G →W →B →NF	✓	B
E_{15}	G →W →B →NF	✓	B
E_{16}	G →NW		
E_{17}	G →NW →C		
E_{18}	G →NW →C		
E_{19}	G →NW →T		
E_{20}	G →NW →T		
E_{21}	G →NW →T		
E_{22}	G →NW →T		
E_{23}	G →NW →H		
E_{24}	G →NW →H		
E_{25}	G →NW →H		

3.3.4 Estratégia de Predição *Top-down* para Métodos Locais

Durante a etapa de classificação, métodos de classificação hierárquica que seguem a abordagem local usualmente realizam a predição de um novo exemplo utilizando uma estratégia da *raiz para as folhas*, frequentemente referenciada como *top-down* e também conhecida como *Pachinko Machine*. Esse é um procedimento recursivo que começa com a apresentação do exemplo ao classificador (ou classificadores) no nível mais raso da hierarquia. Após, com o resultado dessa predição, o processo se repete recursivamente inspecionando apenas os classificadores referentes às classes imediatamente abaixo da(s) classe(s) predita(s) no passo anterior. Esse processo recursivo termina quando uma classe-folha é encontrada. Além disso, é possível definir outros critérios de parada, como um limiar para o grau de confiança da classificação.

É importante observar que esse método, quando aplicado com o modelo HBR, é naturalmente multirrótulo (no sentido de mais de uma classe ser pre-

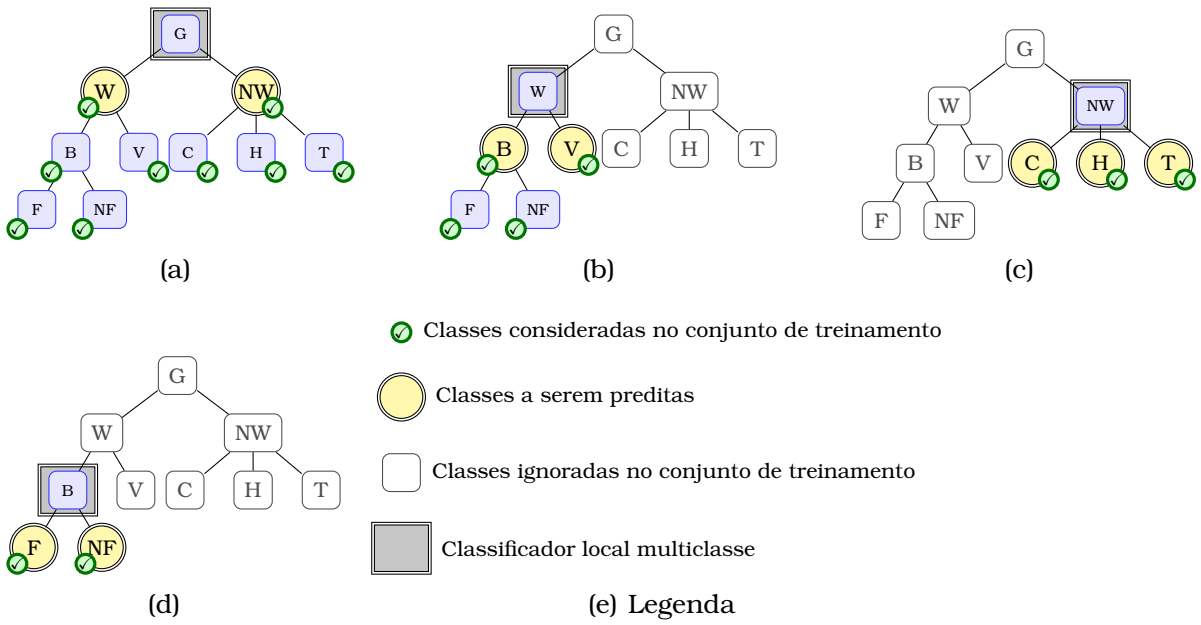


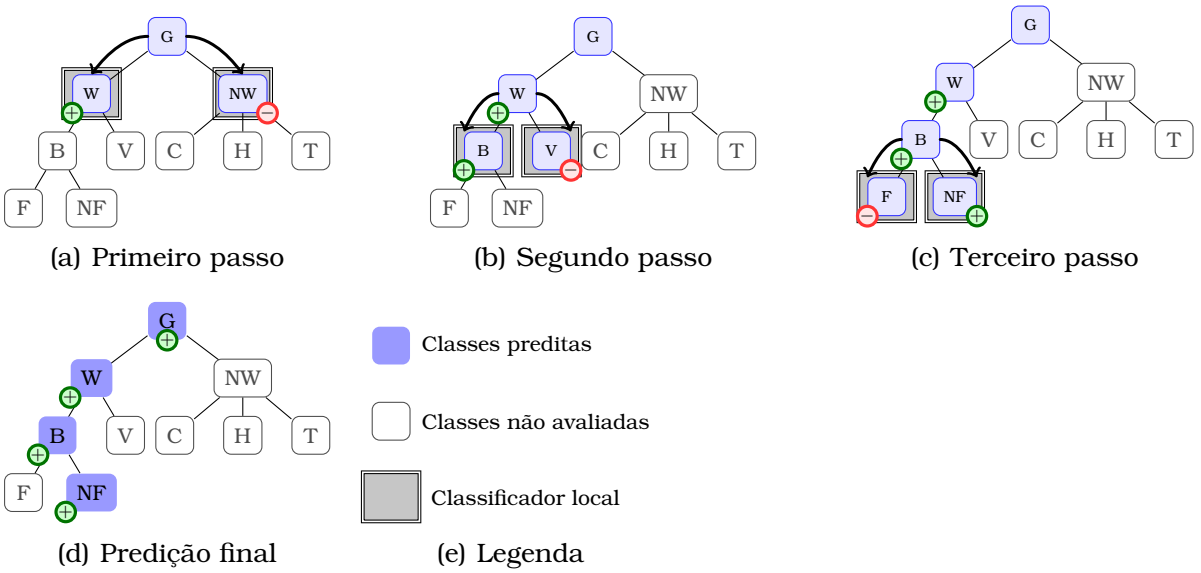
Figura 10: Preparação do conjunto de treinamento para o método HMR.

dita como positiva no mesmo nível da hierarquia). Essa característica pode ser facilmente eliminada se apenas a classe predita com maior grau de confiança em cada nível for atribuída ao exemplo. Com isso, apenas as classes descendentes dessa classe poderão ser preditas nos próximos níveis.

A classificação final de um novo exemplo a partir dessa estratégia, corresponde à união de todas as classes que foram preditas como positivas para esse exemplo, desde que respeitem a taxonomia de classes.

Segundo Freitas e de Carvalho (2007), existem dois tipos gerais de predição (ilustrados nas Figuras 11 e 12, respectivamente): predição obrigatória de classe-folha (*Mandatory leaf node prediction* – MLN), quando a classe mais específica predita deve obrigatoriamente ser uma classe-folha, e predição não obrigatória de classe-folha (*Non-mandatory leaf node prediction* – NMLN), quando a classe mais específica predita pode ser tanto uma classe interna quanto uma folha da taxonomia. Ambos tipos de predição podem ser utilizados tanto com o modelo HBR como HMR.

No caso da predição MLN, o algoritmo avalia inicialmente a predição dos classificadores no primeiro nível da hierarquia, para verificar se algum desses classificadores prediz o novo exemplo como positivo. Nesse caso, temos dois classificadores no primeiro nível, C_{W} e C_{NW}. Para o novo exemplo, o classificador C_{W} prediz a classe positiva enquanto que C_{NW} prediz negativo — Figura 11(a). No próximo passo, o exemplo é apresentado apenas aos classificadores filhos do classificador que o classificou como positivo no nível anterior. Então, os classificadores C_{B} e C_{V} classificam o novo exemplo como positivo e negativo, respectivamente, ao passo que os classificadores na subárvore com raiz NW são ignorados — Figura 11(b). Desse modo, o algoritmo segue pela subárvore



(a) Primeiro passo (b) Segundo passo (c) Terceiro passo

(d) Predição final (e) Legenda

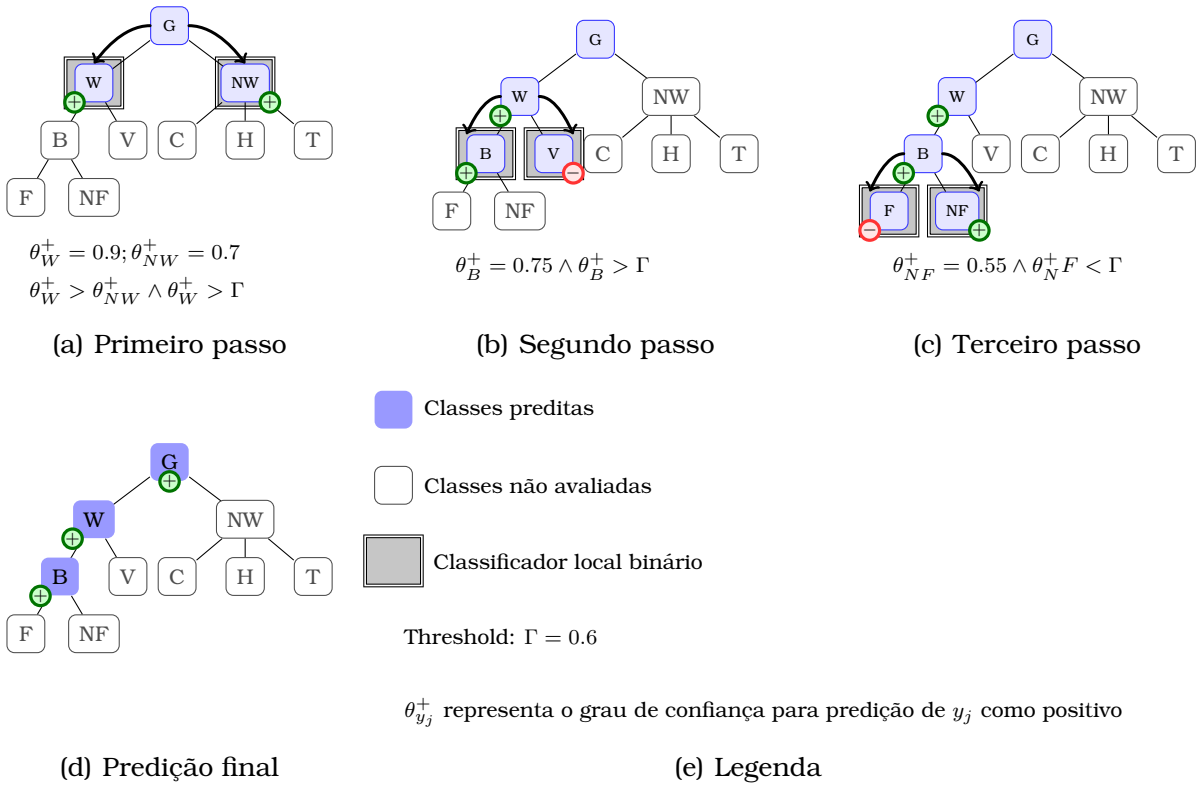
Figura 11: Classificação de um novo exemplo usando estratégia local com predição *top-down* obrigatória até as folhas.

com raiz em B, apresentado o novo exemplo aos classificadores C_{F} e C_{NF}, os quais classificam o novo exemplo como negativo e positivo, respectivamente — Figura 11(c). Assim, a classificação final do novo exemplo é G → W → B → NF, como indicado na Figura 11(d).

Para o caso da predição opcional nas folhas, NMLN, é necessário que, além de escolher o classificador com maior grau de confiança em cada nível, esse valor de confiança seja maior que um limiar pré-especificado, como ilustrado na Figura 12. O processo de classificação é interrompido se o grau de confiança de predição de todos os classificadores inspecionados em um determinado nível for menor que o limiar pré-especificado. Caso contrário continua até que uma classe-folha seja predita.

4 Algoritmos Propostos

Os algoritmos apresentados anteriormente, apesar de serem amplamente utilizados pela comunidade científica, apresentam alguns problemas no que diz respeito à correta classificação de novos exemplos. Em alguns casos, é possível que ocorram inconsistências na predição, ou mesmo que uma decisão equivocada em um nível mais raso degrade completamente o poder de predição final do classificador. Nesse contexto, propomos nessa seção dois novos métodos para classificação hierárquica utilizando a abordagem local. Esses métodos são projetados com o objetivo de minimizar os problemas presentes nos algoritmos apresentados anteriormente, e são descritos a seguir.



(a) Primeiro passo (b) Segundo passo (c) Terceiro passo

(d) Predição final (e) Legenda

Figura 12: Classificação de um novo exemplo usando estratégia *top-down* e predição opicional até as folhas.

4.1 HMR com Estratégia de Multiplicação

Os algoritmos de classificação hierárquica que seguem a abordagem local são dependentes, principalmente, do bom poder de predição dos classificadores locais que estão associados aos níveis mais rasos da hierarquia de classes. Com o modelo HMR com predição *top-down*, em caso de falha na classificação dada pelo classificador associado ao nó raiz, todo o processo de rotulamento é comprometido, pois a predição referente ao próximo nível depende dessa predição. O mesmo ocorre quando o modelo utilizado é HBR. Desse modo, é interessante desenvolver mecanismos para reforçar a correta predição desses classificadores locais evitando erros.

Em (Sun et al., 2004) foi proposto o método *Extended Multiplicative Method* (EMM) com o objetivo principal de diminuir o número de exemplos "bloqueados" nos níveis mais próximos ao nó raiz, permitindo que mais exemplos sejam apresentados aos classificadores mais profundos e, com isso, possibilitar a melhora do poder de predição desses classificadores. É importante observar que em (Sun et al., 2004) o tipo de predição utilizado é NMLN. Para isso, nesse método é calculada a multiplicação das probabilidades de predição obtidas por classificadores em diferentes níveis. Os valores resultantes da multiplicação são comparados a um valor de *threshold* previamente definido para cada nível a fim de determinar a aceitação ou rejeição de exemplos pelos classificadores

locais. Os valores de *threshold* devem ser decrescentes de acordo com o nível na hierarquia. Assim, considerando classificadores locais binários em cada nó, um novo exemplo E é rotulado como y_j (classificado como positivo por C_{y_j}) se $P(y_j|E) \times P(\uparrow y_j|E) \geq \Gamma_{(y_j,\uparrow y_j)}$, onde $\Gamma_{(y_j,\uparrow y_j)}$ é o limiar (*threshold*) mínimo de classificação associado ao nível da classe y_j na taxonomia.

Uma desvantagem do método EMM é a necessidade de definição de diferentes valores de *threshold*, sendo um para cada nível da hierarquia. Inspirados nesse método, porém preocupados com a questão de eficiência computacional e a diminuição de parâmetros para a execução do algoritmo, propusemos uma nova heurística, mais simples, a qual pode ser utilizada com classificadores multiclasse em cada nó da hierarquia, *i.e.*, HMR. Essa heurística também faz uso da predição de classificadores em diferentes níveis da hierarquia para a definição da classe mais provável de modo a reforçar a predição de cada classificador pai com auxílio dos classificadores filhos. Porém, diferentemente do método EMM, nós consideramos o grau de confiança da predição dos diferentes classificadores e não valores de probabilidades, fazendo com que uma gama maior de algoritmos de aprendizado possam ser utilizados para gerar os classificadores locais. Além disso, não há a necessidade de múltiplos valores de corte para a classificação.

Para motivar o uso dessa heurística, considere que um novo exemplo deve ser classificado por meio do modelo HMR de acordo com a hierarquia de classes Glass, apresentada anteriormente. Assim, se o grau de confiança para a classificação resultante do classificador C_{G} for igual 0.5 para cada uma das classes filhas, por exemplo, o rótulo associado ao novo exemplo para esse nível deverá ser selecionado aleatoriamente. Consequentemente, teríamos para esse exemplo 50% de chances de erro na classificação final, uma vez que a predição deve resultar em uma de duas classes possíveis.

No entanto, se utilizarmos os classificadores associados aos nós filhos de G, podemos verificar qual classe tem maior chance de ser a classe correta para esse exemplo. Com essa heurística, o grau de confiança do classificador primário (ou corrente) para cada classe filha é atualizado por meio da média geométrica entre o grau de confiança da predição inicial desse classificador e o máximo grau de confiança do classificador associado a cada classe filha. Assim, sendo $\Theta_{C_{y_j},E} = \left\{\theta^k_{C_{y_j},E}\right\}$ o conjunto com k valores de graus de confiança obtidos pelo classificador C_{y_j} para o exemplo E, e $k = 1..|\downarrow y_j|$, é possível modificar cada valor $\theta^k_{C_{y_j},E}$ considerando a predição dos classificadores filhos para melhor determinar a correta classificação para o nó corrente. A atualização é dada pela Equação 4, onde $\downarrow_k C_{y_j}$ representa o classificador associado à subclasse k de C_{y_j}.

$$\hat{\theta}^k_{C_{y_j},E} = \sqrt{\theta^k_{C_{y_j},E} \times f(\downarrow_k C_{y_j}, E)} \tag{4}$$

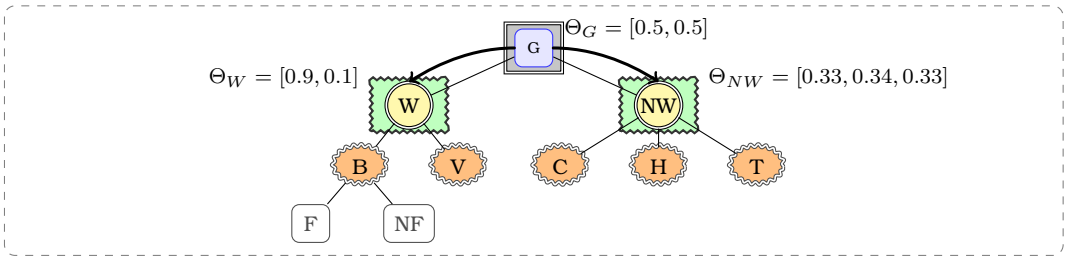
$$f(C_i, E) = \underset{\theta^k}{\operatorname{argmax}}\ \Theta_{C_i,E} \tag{5}$$

Como ilustração, considere que um novo exemplo é submetido a um classificador hierárquico HMR treinado exatamente como o modelo padrão descrito na Seção 3.3.3, considerando a hierarquia de classes Glass. Para rotular um novo exemplo, ele é inicialmente submetido ao classificador local C_{G}. Suponha que o grau de confiança para a classificação desse novo exemplo seja igual a 0.5 para cada uma das suas subclasses. Considere que antes de decidir a predição do classificador C_{G}, esse mesmo exemplo é apresentado aos classificadores C_{W} e C_{NW}, ambos filhos de C_{G}, e que o grau de confiança para classificação resultante desses classificadores seja $[\text{B}: 0.9, \text{V}: 0.1]$ e $[\text{C}: 0.33, \text{T}: 0.34, \text{H}: 0.33]$, respectivamente, conforme apresentado na Figura 13(a). Com essa informação adicional, selecionamos o maior valor para o grau de confiança resultante de cada classificador filho de C_{G} — Equação 5. Nesse caso, o maior grau de confiança resultante do C_{W} é 0.9 para a classificação como B, enquanto que o classificador C_{NW} resultou em 0.34 para a classe T — Figura 13(b).

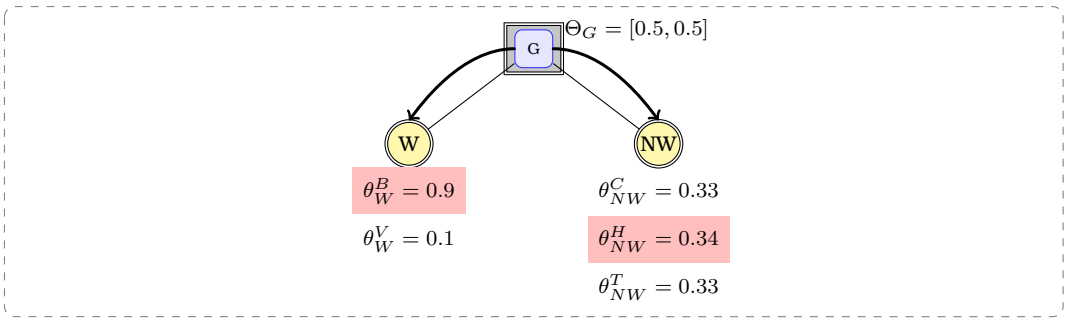
Para esse exemplo, aplicamos a média geométrica conforme Equação 4, resultando $C_{\text{G}} = [\text{W}: \sqrt{0.5 \times 0.9}, \text{NW}: \sqrt{0.5 \times 0.34}]$ — Figura 13(c). Como resultado da operação, teríamos $C_{\text{G}} = [\text{W}: 0.67, \text{NW}: 0.41]$ — Figura 13(d). Consequentemente, após essa atualização e devido à influência do nível imediatamente abaixo na hierarquia, a classe mais provável escolhida pelo classificador C_{G} seria W — Figura 13(d). Seguindo a estratégia de predição *top-down*, esse processo é repetido para a subárvore com raiz em C_{W}.

É importante lembrar que o treinamento dos classificadores locais é efetuado exatamente como no modelo padrão HMR, apresentando diferenças apenas na fase de classificação dos novos exemplos.

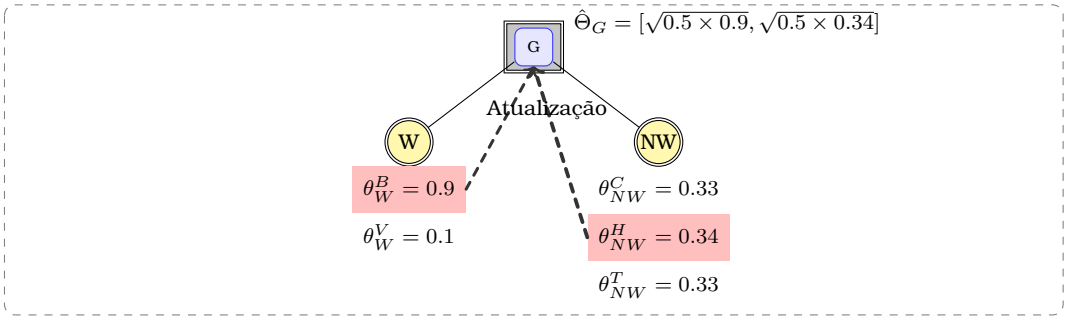
Uma vantagem desse modelo com relação ao modelo padrão de HMR com predição opcional nos níveis folha, é que os nós filhos influenciam na classificação no nível anterior, fazendo com que cada classificador local considere um passo a frente a cada decisão. Porém, esse modelo ainda não elimina a necessidade de classificadores precisos, principalmente nos níveis mais acima da hierarquia. Quanto mais precisos forem os classificadores locais, melhor será o aproveitamento da influência dos nós filhos e, consequentemente, maiores serão as chances de acerto para a predição. Além disso, esse modelo por si só não trata o bloqueio de exemplos nos primeiros níveis da hierarquia. Para esse problema, é necessária a utilização de outras estratégias, como valores



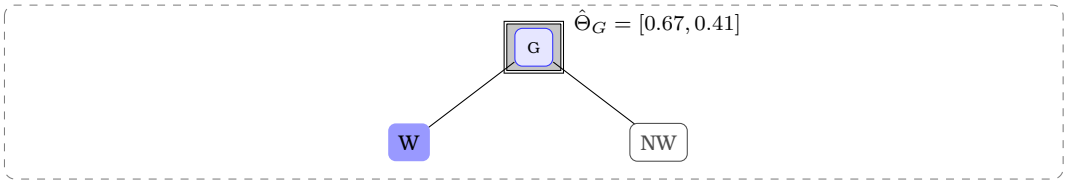
(a) Classificação do novo exemplo pelos classificadores primário e secundário



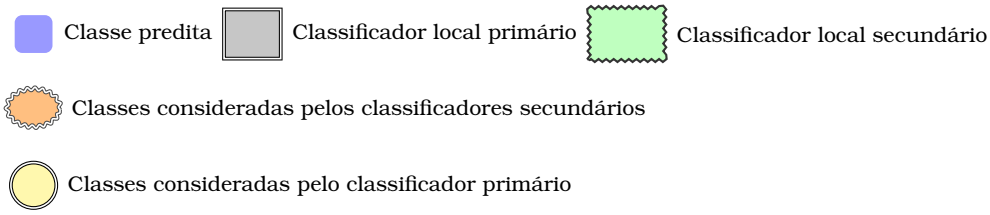
(b) Identificação das maiores confianças



(c) Atualização dos valores de confiança do classificador primário



(d) Predição obtida pelo classificador primário depois da atualização



(e) Legenda

Figura 13: Classificação por meio do modelo HMR com estratégia de multiplicação.

decrescentes de *threshold* para cada classificador local ou para cada nível da hierarquia.

4.2 Modelo Híbrido de Cooperação: HBR + HMR

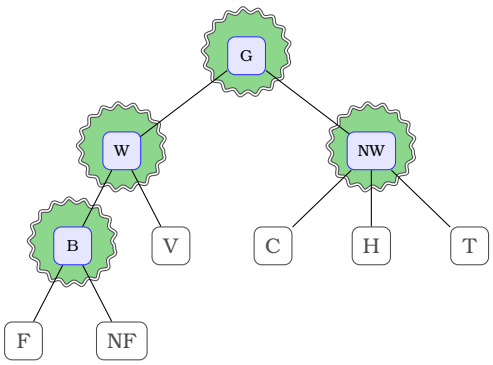
Quando o modelo HMR é aplicado para a construção de um classificador hierárquico, cada classificador local obrigatoriamente rotula um novo exemplo com uma dentre suas subclasses. Entretanto, mesmo que o exemplo não pertença a nenhuma dessas classes, ele será avaliado ou por ter sido erroneamente classificado no passo anterior ou porque sua classe mais específica é a classe do nó corrente. Assim, propomos também um modelo para tentar evitar que esse erro seja cometido e impedir que o mesmo seja propagado para os demais níveis da hierarquia.

Como ilustração, considere novamente a hierarquia de classes Glass. Suponha que um novo exemplo com informações de um vidro de um porta-retratos seja submetido para a classificação por meio do modelo HMR. Nesse caso, o novo exemplo faz parte da classe NW no primeiro nível. Porém, não se encaixa em nenhuma das subclasses C, T ou H. Nesse sentido, a classe mais específica para esse exemplo, considerando essa hierarquia de classes conhecida, é NW. Porém, mesmo utilizando a predição opcional nos nós folha, é possível que ele seja erroneamente rotulado com uma das subclasses de NW, uma vez que o classificador multiclasse relacionado à classe NW sempre selecionará uma de suas subclasses para rotular um novo exemplo.

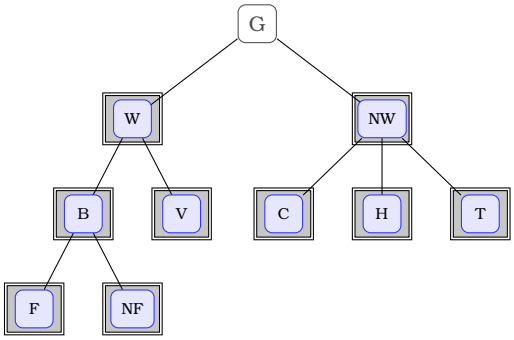
Para evitar esse tipo de erro e, ainda, reforçar as predições corretas do classificador multiclasse (aqui referenciado como classificador primário) por meio de uma segunda perspectiva, propomos a utilização de um modelo híbrido que usa, além dos classificadores multiclasse em cada nó não-folha, classificadores locais binários associados a todos os nós, exceto ao nó raiz. Em outras palavras, estamos propondo a utilização dos modelos HMR e HBR simultaneamente para a construção do classificador hierárquico, conforme ilustrado na Figura 14.

A construção dos modelos HMR e HBR e feita de maneira independente. Porém, no modelo híbrido os classificadores locais multiclasse (HMR) são considerados os classificadores primários e os classificadores binários (HBR) são considerados secundários, utilizados apenas para reforçar a predição dos classificadores multiclasse e evitar a propagação de erros.

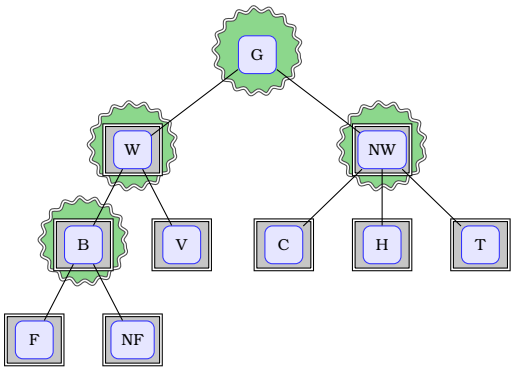
Durante o rotulamento de um novo exemplo por meio da estratégia de predição *top-down*, o classificador primário associado ao nó raiz classifica o exemplo resultando os valores de grau de confiança da classificação em cada uma das suas subclasses. Em seguida, os classificadores secundários associados a cada uma das subclasses do classificador primário são utilizados



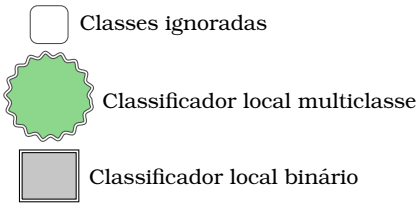
(a) Modelo HMR: um classificador local multiclasse por nó pai



(b) Modelo HBR com um classificador local binário por nó, exceto a raiz



(c) Modelo Híbrido com classificadores locais binários e locais multiclasse



(d) Legenda

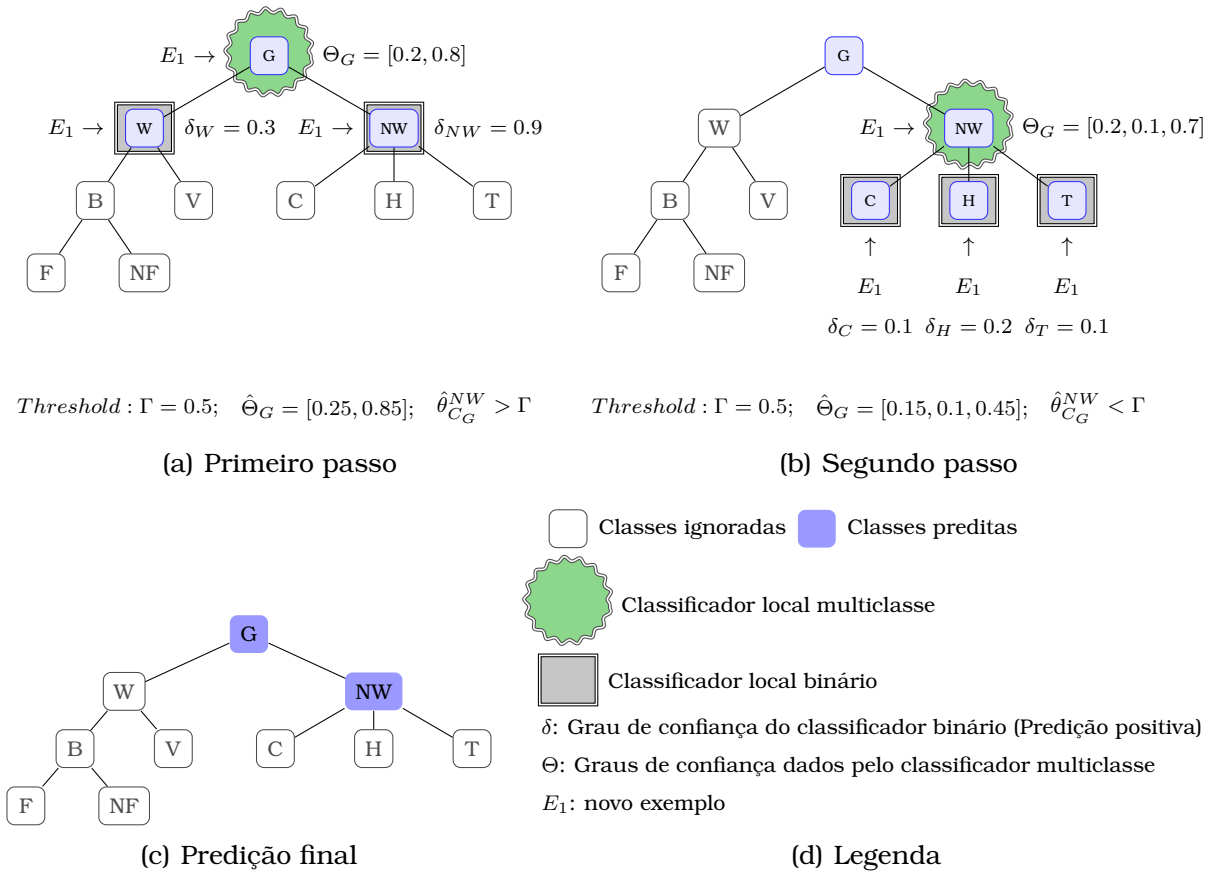
Figura 14: Ilustração do modelo híbrido com classificadores locais.

para classificar o novo exemplo, resultando também seus respectivos graus de confiança para predição como positivo. Então, calcula-se a média dos valores dos graus de confiança para cada uma das subclasses do classificador primário, obtidos tanto pelo classificador primário quanto pelos classificadores secundários, conforme a Equação 6, onde $\theta^k_{C_{Prim,E}}$ representa o grau de confiança para a classificação obtido pelo classificador primário referente à subclasse k considerando o exemplo E como entrada, e $\delta_{C^k_{sec,E}}$ representa o grau de confiança na predição para o exemplo E como positivo obtido por meio do classificador secundário (binário) associado a subclasse k do classificador primário.

$$\hat{\theta}^k_{C_{Prim,E}} = \frac{(\theta^k_{C_{Prim,E}} + \delta_{C^k_{sec,E}})}{2} \tag{6}$$

A rotulação de um novo exemplo por meio desse modelo é ilustrada na Figura 15 com classificação opcional até as classes-folha e, na Figura 16, com predição obrigatória até as classes-folha. Em ambos os casos, considere que a classe verdadeira dos dois novos exemplos é G →NW.

A principal desvantagem desse método com relação aos descritos anteriormente é a sua complexidade de tempo computacional, uma vez que dois



(a) Primeiro passo

(b) Segundo passo

(c) Predição final

(d) Legenda

Figura 15: Exemplo de classificação por meio do modelo híbrido com estratégia de predição opcional até o nível das classes-folha.

modelos são treinados ao mesmo tempo. Assim, a complexidade total desse modelo é igual à soma das complexidades do modelo HMR e do modelo HBR. Se considerarmos a complexidade do modelo HBR como $(|L| - 1) \times O(Bin)$, onde Bin representa o classificador base binário, e do modelo HMR como $(|L - folhas|) \times O(Mult)$, onde $Mult$ representa o classificador base multiclasse, o modelo híbrido terá complexidade total de $|L| \times (O(Bin) + O(Mult)) - |folhas| \times (O(Mult) - O(Bin))$.

Por outro lado, esse modelo apresenta maiores garantias "teóricas" com relação à capacidade de classificar corretamente novos exemplos.

Considerando a Figura 15 como ilustração para a classificação de um novo exemplo, cuja classe verdadeira é G $\rightarrow$NW, o classificador C_{NW} o rotularia como H com grau de confiança 0.7 caso não houvesse a influência dos classificadores secundários. Porém, com a avaliação das predições dadas pelos classificadores secundários C_{C}, C_{T} e C_{H}, os quais predizem o novo exemplo como negativo, a predição do classificador primário é modificada e, com isso, a classe correta é predita.

Para o segundo exemplo de teste, classificado pelo modelo híbrido com estratégia de predição obrigatória até o nível das classes-folha, apesar de o classificador primário C_{NW} predizer a classe H para esse exemplo, mais uma

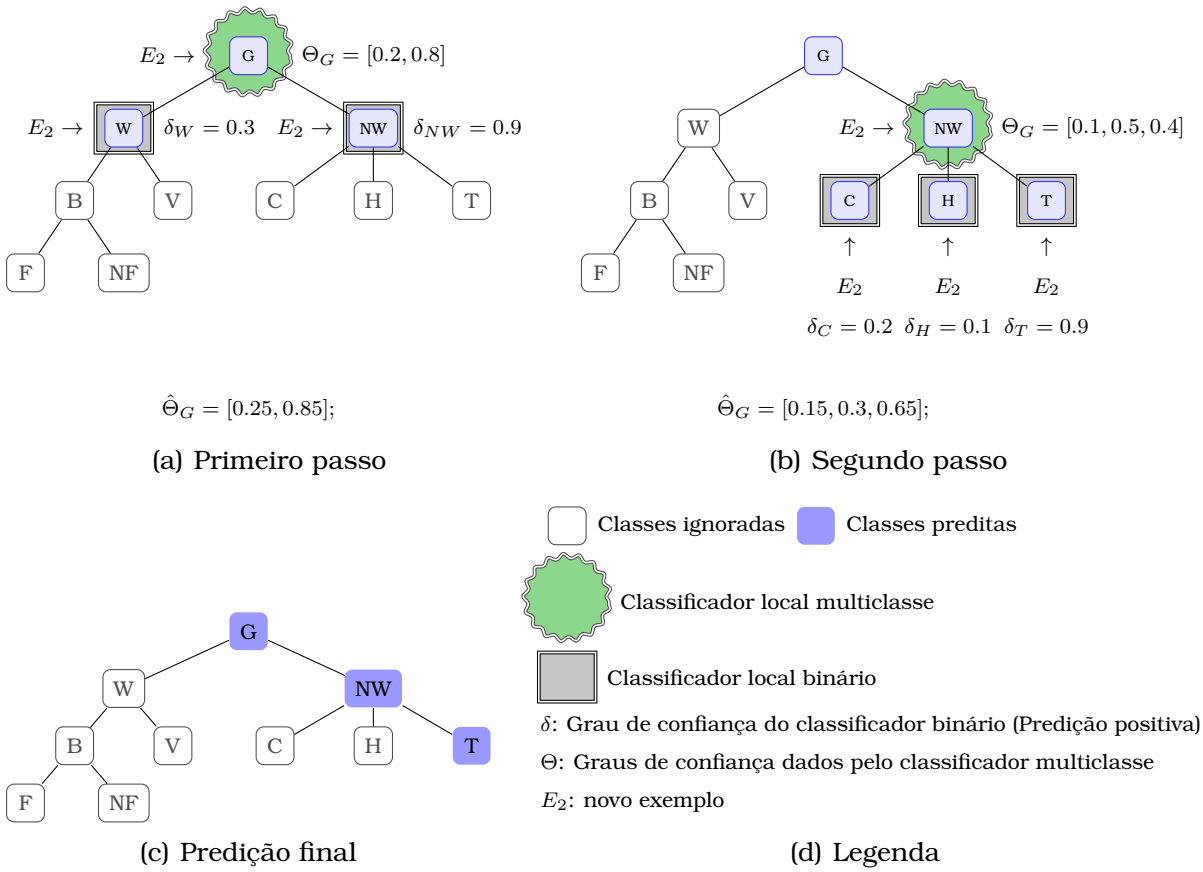


Figura 16: Exemplo de classificação por meio do modelo híbrido com estratégia de predição obrigatória até o nível das classes-folha.

vez os classificadores secundários influenciaram na classificação final, pois os dois classificadores secundários C e H não classificam o novo exemplo como positivo, ao passo que o classificador secundário T o predisse como positivo e com alto grau de confiança. Com a média dos graus de confiança entre os classificadores primários e secundários, a predição final é G →NW →T.

5 Medidas de Avaliação

São muitas as soluções ou algoritmos propostos para o problema de classificação. Também são abundantes as medidas propostas para avaliar o poder de predição dos algoritmos desenvolvidos para essa tarefa. No entanto, a maioria dessas medidas foram projetadas para avaliação da classificação plana. Assim, não consideram relações estruturais entre as classes, como generalização e especialização, presentes em uma estrutura hierárquica de classes. Um estudo a respeito das medidas de avaliação da classificação plana pode ser encontrado em (Monard e Baranauskas, 2003). A seguir é apresentada uma breve descrição de algumas das medidas de classificação plana que servem como base para o entendimento das medidas propostas para avaliação da classificação hierárquica.

5.1 Medidas de Avaliação da Classificação Plana

Uma das maneiras de avaliar a predição de um classificador plano, seja ele binário ou multiclasse, é por meio da matriz de confusão (MC). Essa matriz representa o número de classificações corretas e erradas obtidas pelo classificador sobre um conjunto de exemplos de teste. Os resultados são totalizados em duas dimensões: *classe verdadeira* e *classe predita*. Cada célula $MC(y_i, y_j)$ da matriz representa o número de exemplos que realmente pertencem a classe y_i mas que foram classificados como pertencentes a classe y_j. Na Tabela 6 é mostrada a matriz de confusão de um classificador plano com duas classes, $+$ e $-$.

Tabela 6: Matriz de confusão. VP: verdadeiro positivo; VN: verdadeiro negativo; FP: falso positivo; FN: falso negativo.

		Classe Predita	
		$+$	$-$
Classe Verdadeira	$+$	VP	FN
	$-$	FP	VN

A partir dessas informações são definidas diversas medidas de performance, tais como *acurácia* (Acc), *erro* (Err), *precisão* (Pr) e *recall* (Re), apresentadas respectivamente nas Equações 7 a 10.

A acurácia, também conhecida como precisão total, é uma medida simples que calcula a fração de exemplos corretamente classificados. O erro por outro lado, é o complemento da acurácia.

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \tag{7}$$

$$Err = \frac{FP + FN}{VP + VN + FP + FN} = 1 - Acc \tag{8}$$

A precisão, também chamada de *positive reliability* ou *positive predictive value*, é calculada como uma medida de fidelidade, no sentido de que seu valor representa a fração de exemplos realmente positivos dentre os exemplos que foram preditos como positivos.

$$Pr = \frac{VP}{VP + FP} \tag{9}$$

O *recall* (*sensitivity* ou *true positive rate*), por outro lado, representa uma medida de perfeição no sentido de que seu valor representa o número de exemplos corretamente preditos como positivos pelo número total de exemplos cuja classe verdadeira é positivo.

$$Re = \frac{VP}{VP + FN} \tag{10}$$

Outra maneira bastante usual para a avaliação de classificadores é considerar a média harmônica entre duas medidas, por exemplo precisão e *recall*, por meio da medida F (*F-Measure*) apresentada na Equação 11, na qual o parâmetro β indica o peso dado a uma das medidas. Quando o valor de β é igual a 1 as duas medidas tem o mesmo peso no cálculo da média final. Com $\beta = 2$ a medida de *recall* recebe o dobro do peso dado a precisão, ao passo que para $\beta = 0.5$ ocorre a situação inversa, ou seja, a precisão recebe o dobro do peso que o *recall*.

$$f_\beta = (1 + \beta^2) \times \frac{Pr \times Re}{\beta^2 \times Pr + Re} \tag{11}$$

Além dessas medidas de performance, o erro majoritário também é uma informação importante, uma vez que é obtido a partir da estratégia mais *naive* possível para a classificação, simplesmente rotulando todos os exemplos com a classe majoritária. Com isso, pode ser utilizado como base de comparação para avaliação da performance de algoritmos de aprendizado. A taxa de erro majoritário é apresentada na Equação 36:

$$Error_{majority} = 1 - \max_{j=1..|L|} P(y_j) \tag{12}$$

onde $P(y_j)$ representa a probabilidade da classe y_j, usualmente aproximada por meio da frequência relativa dessa classe no conjunto de treinamento.

5.2 Medidas de Avaliação da Classificação Hierárquica

As medidas para avaliação da classificação plana apresentadas podem ser utilizadas também para avaliar a performance da predição de algoritmos de classificação hierárquica. Porém, essas medidas são inadequadas para esse tipo de problema, pois não diferenciam os erros de classificação que podem ocorrer em cenários com relacionamentos estruturais entre as classes (Kiritchenko et al., 2005), como é o caso da classificação hierárquica.

Pode ser observado que as medidas de avaliação propostas para o problema de classificação plana, consideram que todos os erros de classificação tem a mesma importância. No entanto, para problemas hierárquicos, existem erros como de generalização e/ou especialização que devem receber pesos diferenciados, uma vez que um erro de predição cometido nos níveis mais rasos da hierarquia (classes mais gerais) são mais importantes que os erros cometidos nos níveis mais profundos, nos quais as classes são mais específicas (Freitas e de Carvalho, 2007). Em outras palavras, na classificação hierárquica é importante distinguir os seguintes tipos de erros:

- *erro de generalização*: o percentual de exemplos classificados erronea-

mente em uma classe ancestral da classe correta mais específica.

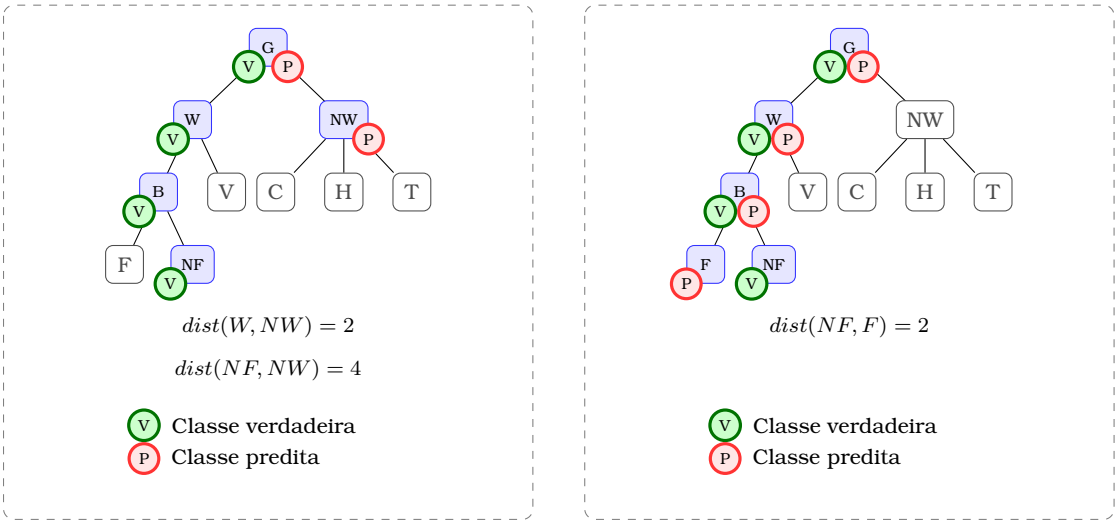
- *erro de especialização*: o percentual de exemplos classificados erroneamente em uma das classes descendentes da classe correta mais específica.

Uma medida bastante utilizada no contexto da classificação hierárquica é baseada em conceitos de distância entre classes (número de arestas no caminho entre as duas classes na hierarquia). Mas essa medida também apresenta algumas desvantagens, como a difícil adaptação para hierarquias definidas sobre estruturas de grafos (DAG), nos quais existem múltiplos caminhos entre duas categorias. Além disso, esse tipo de medida considera que erros em diferentes níveis da hierarquia têm o mesmo peso. Por exemplo, na Figura 17, a predição de uma classe irmã da classe verdadeira no primeiro nível da hierarquia (predição de W ao invés de NW, acarreta distância 2, pois o caminho tem apenas duas arestas) teria o mesmo peso que a predição de uma classe irmã no nível 3 (predição de F ao invés de NF). Porém, um erro no terceiro nível da hierarquia representa um erro de refinamento, em geral mais específico, e não deveria ser penalizado da mesma maneira que o erro nos níveis mais rasos da hierarquia, os quais propagam o erro para níveis mais profundos da hierarquia.

Mesmo sabendo que as medidas de classificação plana e as hierárquicas baseadas em distância podem não representar corretamente o poder de predição dos classificadores hierárquicos, em muitas das pesquisas publicadas os resultados são avaliados por meio dessas medidas (Clare e King, 2001; Toutanova et al., 2001; Struyf et al., 2005). Ainda assim, os pesquisadores reconhecem que essas medidas não são apropriadas para a classificação hierárquica, pois, como mencionado, os diferentes tipos de erros recebem o mesmo grau de importância.

Outros pesquisadores reportam os resultados das suas pesquisas considerando como medida de performance apenas a contagem de exemplos corretamente (ou erroneamente) classificados considerando como acerto apenas os casos para os quais todos os rótulos foram corretamente preditos (Labrou, 1999; Sasaki e Kita, 1998). Contudo, muitos algoritmos permitem classificação parcial. Assim, o cálculo da performance desses algoritmos considerando apenas a predição perfeita no conjunto de teste não é apropriado, pois não representa o real poder de predição dos algoritmos.

Nesse contexto, com objetivo de definir medidas capazes de avaliar a performance dos classificadores hierárquicos de maneira fidedigna, Kiritchenko et al. (2005) formularam três pré-requisitos que devem ser satisfeitos pelas medidas propostas para esse fim:



(a) Erro da predição de uma classe mais geral

(b) Erro da predição de uma classe mais específica

Figura 17: Erros de classificação hierárquica.

1. A medida deve ser capaz de *avaliar predições parciais*: por exemplo, o erro gerado pela predição da classe B (considerando apenas a predição da classe mais específica) dado que a classe verdadeira é NF, deve ser menos penalizado que a predição de NW, uma vez que B faz parte do caminho correto entre a classe verdadeira mais específica e a raiz, ao passo que NW não faz parte desse caminho.

2. A medida deve dar *maior peso para erros envolvendo classes mais distantes na hierarquia de classes*: dois casos devem ser considerados.

   (a) a medida resulta em uma melhor avaliação para predições mais específicas corretas, mesmo que não sejam tão específicas quanto a classe verdadeira. Por exemplo, considerando a classe NF como sendo a classe verdadeira mais específica, é preferível predizer B a W como classe mais específica, pois B está mais perto da classe verdadeira.

   (b) a medida resulta em pior avaliação para predições mais específicas erradas, comparando com predições erradas menos específicas. Por exemplo, predizer a classe F é pior que predizer B quando a classe correta é W, uma vez que F está ainda mais distante que B da classe verdadeira.

3. A medida deve resultar em *pior avaliação para erros de predição mais gerais*: por exemplo, quando a classe mais específica verdadeira é NF, o erro dado à predição de um novo exemplo como sendo F sua classe mais específica é menos severo que a predição de W como sua classe mais

especifica.

Diversas medidas para avaliar a classificação hierárquica têm sido propostas, mas nem todas respeitam os pré-requisitos apresentados Kiritchenko et al. (2005). Essas medidas podem ser agrupadas em 4 tipos gerais (Freitas e de Carvalho, 2007; Costa et al., 2007):

1. distância,
2. similaridade semântica,
3. custo de erro de classificação, e
4. baseadas na hierarquia.

Segue a descrição de algumas das medidas propostas em cada um desses grupos.

5.2.1 Baseadas em Distância

Essa categoria de medidas de performance de predição consiste em atribuir para cada erro de classificação hierárquica um custo que é proporcional à distância entre a classe mais específica predita e a classe mais específica verdadeira de cada exemplo. Essa categoria pode ser subdividida ainda em duas subcategorias (Freitas e de Carvalho, 2007):

1. dependente de profundidade (*depth-dependent*); e
2. independente de profundidade (*depth-independent*).

descritas a seguir.

5.2.1.1 Dependente de Profundidade

Nessa categoria de medidas, proposta por Blockeel et al. (2002), a distância entre duas classes é obtida por meio de uma função que considera dois fatores: (a) o número de arestas entre as classes predita e verdadeira (mais específicas) e (b) a profundidade dessas duas classes na hierarquia.

Uma maneira de definir essa função é atribuir pesos (custo) a cada aresta da hierarquia. A partir desses pesos, o erro de classificação associado à predição é dado pela soma dos pesos de todas as arestas no caminho entre as duas classes. No entanto, para implementar a idéia de que erros de predições incorretas em nós mais rasos (mais próximos ao nó raiz) tenham avaliações piores que as predições erradas em nós mais profundos, os pesos das arestas devem ser definidos considerando o nível onde ocorrem.

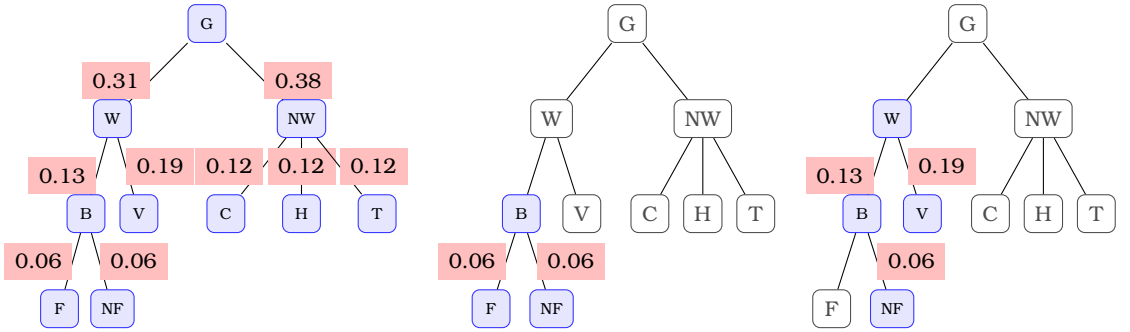
Para solucionar esse problema, Holden e Freitas (2006) e Blockeel et al. (2002) propõem a definição dos pesos de modo exponencialmente decrescente de acordo com o aumento da profundidade dos nós. Porém, essa proposta também têm desvantagens. A primeira ocorre quando a árvore que representa a hierarquia de classes é fortemente desbalanceada, no sentido de que diferentes nós-folha aparecem em diferentes níveis da hierarquia. Para esse caso, um erro envolvendo duas classes-folha rasas (verdadeira e predita) será menos penalizado que um erro envolvendo um par de classes-folha que estão em um nível mais profundo, simplesmente porque no último caso haverá mais arestas associadas ao caminho entre as duas classes. Essa menor penalização é discutível, uma vez que um erro na predição envolvendo classes rasas impede que os níveis mais profundos sejam avaliados.

Um outro problema identificado por Lord et al. (2003) está relacionado à variação da profundidade dos nós-folha, uma vez que o fato de uma classe folha ser mais profunda que outra não significa necessariamente que a primeira é mais informativa que a segunda. Por exemplo, uma classe no terceiro nível da árvore pode representar informação tão específica quanto uma classe no oitavo nível da estrutura hierárquica, se as duas classes estiverem em diferentes subárvores do nó raiz. Então, a atribuição de pesos considerando apenas a profundidade das arestas e não a informação associada com as classes pode ser um problema.

Uma proposta para minimizar esses problemas é usar não apenas medidas que consideram a profundidade das classes, mas também que resultem no mesmo peso total para cada subárvore sendo comparada, como por exemplo, a medida *Normalized Weighted Depth-Dependent Distance-Based Accuracy* proposta por Holden e Freitas (2009) e posteriormente utilizada em (Metz e Freitas, 2009), a qual define os pesos das arestas de modo que o caminho partindo da raiz até um nó-folha tenha peso total igual a 0.5, e o peso das arestas desse caminho depende da profundidade da aresta.

Com essa medida, a distância entre duas folhas que não fazem parte da mesma subárvore, *i.e.*, que tenham como ancestral comum mais profundo o nó raiz da hierarquia, resultará sempre em 1, valor esse que representa o erro máximo para a classificação de um exemplo. Observe que se um erro de predição ocorre em um nível bastante profundo onde a classe predita é irmã da classe verdadeira, a avaliação dessa predição resulta em um valor bastante baixo se comparado com um erro que tenha sido cometido nos primeiros níveis da hierarquia. Considere como exemplo a Figura 18, para a qual os pesos das arestas foram definidos de acordo com essa estratégia. Assim, se a classe mais específica de um determinado exemplo E é F e a predição for NF, o erro associado é 0.12 (Figura 18(b)), ao passo que o erro associado à predição como

sendo V seria 0.38 (Figura 18(c)).



(a) Pesos definidos (b) Distância entre F e NF (c) Distância entre V e NF

Figura 18: Exemplo de hierarquia com pesos definidos para o cálculo da distâncias entre nós.

Dados os pesos das arestas da estrutura hierárquica de classes, a acurácia baseada na distância normalizada para um exemplo de teste E_i é definida pela Equação 13,

$$Acc_{E_i}^{Norm} = 1 - dist^w(\odot(Y_i), \odot(Z_i)) \tag{13}$$

tal que Y_i é o conjunto de classes verdadeiras do exemplo E_i e Z_i é o conjunto de classes preditas para esse exemplo. O operador $\odot$ retorna a classe mais específica contida no conjunto de classes especificado, enquanto que $dist_w(y, z)$ é a distância entre os nós representados pelas classes y e z considerando os pesos nas arestas que compõem o caminho entre essas duas classes.

É importante observar que essa medida de acurácia varia entre 0 (erro de predição em todos os níveis) e 1 (predição perfeita), e a acurácia global do classificador é simplesmente a média obtida sobre todo o conjunto de teste, conforme a Equação 14.

$$Acc_{global}^{Norm} = \frac{1}{|Te|} \times \sum_{i=1}^{|Te|} Acc_{E_i}^{Norm} \tag{14}$$

5.2.1.2 Independente de Profundidade

Nesse caso, a distância entre duas classes é medida por meio do número de arestas que compõem o menor caminho entre duas classes. Foi inicialmente proposta por Wang et al. (1999) e posteriormente utilizada por Sun e Lim (2001) no contexto de classificação hierárquica de documentos com múltiplos caminhos na hierarquia, como uma extensão das medidas de taxa de *precisão*, *recall*, *acurácia* e *erro*. Em (Costa et al., 2007), essas medidas foram adaptadas de modo que apenas um caminho na hierarquia seja considerado. Para isso, inicialmente a contribuição de cada falso positivo é calculada (Equação 15). Esse valor de contribuição é posteriormente normalizado entre

$[-1, 1]$ (Equação 16). Após, para cada classe y_j esse valor normalizado de cada falso positivo é somado (Equação 17). De maneira análoga, a contribuição das predições de falsos negativos também é calculada, conforme a Equação 18.

$$Con(Y_i, Z_i) = 1.0 - \frac{Dist(\odot(Z_i), \odot(Y_i))}{Dist_\sigma} \tag{15}$$

$$RCon(Y_i, Z_i) = min(1, max(-1, Con(Y_i, Z_i))) \tag{16}$$

$$FpCon_{y_j} = \sum_{E_i \in FP_{y_j}} RCon(Y_i, Z_i) \tag{17}$$

$$FnCon_{y_j} = \sum_{E_i \in FN_{y_j}} RCon(Y_i, Z_i) \tag{18}$$

Para o cálculo desses valores de contribuição de falsos positivos e negativos, um valor de distância $Dist_\sigma$ maior que zero deve ser especificado para a normalização.

A partir dos valores obtidos por meio das Equações 15 a 18, as medidas de *precisão*, *recall*, *acurácia* e *erro* usualmente utilizadas na avaliação da classificação plana, podem ser estendidas para a avaliação da classificação hierárquica. As extensões dessas medidas são apresentadas nas Equações 19 a 22, respectivamente. Observe que esses valores são definidos para cada classe y_j na hierarquia.

$$Pr_{y_j} = \frac{max(0, |VP_{y_j}| + FpCon_{y_j} + FnCon_{y_j})}{|VP_{y_j}| + |FP_{y_j}| + FnCon_{y_j}} \tag{19}$$

$$Re_{y_j} = \frac{max(0, |VP_{y_j}| + FpCon_{y_j} + FnCon_{y_j})}{|VP_{y_j}| + |FN_{y_j}| + FpCon_{y_j}} \tag{20}$$

$$Acc_{y_j} = \frac{|VP_{y_j}| + |VN_{y_j}| + FpCon_{y_j} + FnCon_{y_j}}{|VP_{y_j}| + |VN_{y_j}| + |FP_{y_j}| + |FN_{y_j}|} \tag{21}$$

$$Er_{y_j} = \frac{|FP_{y_j}| + |FN_{y_j}| + FpCon_{y_j} + FnCon_{y_j}}{|VP_{y_j}| + |VN_{y_j}| + |FP_{y_j}| + |FN_{y_j}|} \tag{22}$$

Duas maneiras de avaliar a performance global dos classificadores hierárquicos é por meio do cálculo das médias micro (*micro-average*) e macro (*macro-average*) dessas medidas, calculadas conforme as Equações 23 a 26, onde o sobrescrito μ indica *micro-average* e M indica *macro-average*.

$$Pr^\mu = \frac{\sum_{j=1}^{|L|}(max(0, |VP_{y_j}| + FpCon_{y_j} + FnCon_{y_j})}{\sum_{j=1}^{|L|}(|VP_{y_j}| + |FP_{y_j}| + FnCon_{y_j})} \tag{23}$$

$$Re^{\mu} = \frac{\sum_{j=1}^{|L|}(max(0, |VP_{y_j}| + FpCon_{y_j} + FnCon_{y_j}))}{\sum_{j=1}^{|L|}(|VP_{y_j}| + |FN_{y_j}| + FpCon_{y_j})} \tag{24}$$

$$Pr^{M} = \frac{\sum_{j=1}^{|L|} Pr_{y_j}}{|L|} \tag{25}$$

$$Re^{M} = \frac{\sum_{j=1}^{|L|} Re_{y_j}}{|L|} \tag{26}$$

Essas medidas de performance têm como principal vantagem a sua simplicidade, tanto conceitual quanto de implementação. Porém, tem como desvantagem o fato de não diferenciar erro de generalização ou especialização. Assim, não satisfazem os pré-requisitos descritos na Seção 5.2.

5.2.2 Medidas Baseadas em Semântica

As medidas de performance dessa categoria avaliam a qualidade da predição de um classificador hierárquico considerando a similaridade semântica entre a classe predita e a classe verdadeira. Lord et al. (2003) propuseram uma medida projetada para a avaliação da classificação hierárquica no contexto de Ontologias Genética (GO). Essa medida considera a especificidade de um termo da GO, que é estimada a partir da probabilidade de utilização desse termo em anotações de genes. A similaridade de dois termos é calculada como a probabilidade mínima de seus ancestrais comuns. Uma vez que muitos pares de termos podem apresentar o mesmo conjunto de nós ancestrais e, então, ter a mesma similaridade semântica, essa medida não tem um bom poder discriminativo para ser usada como uma medida de avaliação da classificação hierárquica (Kiritchenko et al., 2005).

Outra medida, proposta em (Sun e Lim, 2001), utiliza o centróide de cada classe y_j, ζ_{y_j}, para o cálculo das Equações 27 a 29, que representam a similaridade semântica entre classes, a qual é posteriormente utilizada para a definição das taxas de *precisão*, *recall*, *acurácia* e *erro*, como no caso das medidas baseadas em distância independente de profundidade, também propostas por Sun e Lim (2001) e descritas na Seção 5.2.1.2.

Assim, a similaridade entre as classes y_i e y_j, é calculada por meio da Equação 27.

$$CS(y_i, y_j) = \frac{\zeta_{y_i} \times \zeta_{y_j}}{\sqrt{\zeta_{y_i}^2 \times \zeta_{y_j}^2}} \tag{27}$$

Com os valores de similaridade entre todas as classes, calcula-se a similaridade média entre classes (ACS) — Equação 28. Então, para cada predição é calculada a contribuição para falsos positivos utilizando a Equação 29.

$$ACS = \frac{2 \times \sum_{i=1}^{|L|} \sum_{j=i+1}^{|L|} CS(y_i, y_j)}{|L| \times (|L| - 1)} \tag{28}$$

$$FpCon(Y_i, Z_i) = \frac{CS(\odot(Z_i), \odot(Y_i)) - ACS}{1 - ACS} \tag{29}$$

Uma desvantagem dessas medidas é que, em muitos problemas, a hierarquia de classes por si já representa as relações de similaridade semântica por meio dos relacionamentos de superclasse e subclasse (Freitas e de Carvalho, 2007). Assim, a classe F é mais semelhante a NF que a classe C, pois F e NF são irmãs, ao passo que C está em um ramo distinto da árvore — Figura 18 página 36. Observe que essa noção de similaridade está implicitamente representada na estrutura hierárquica de classes, uma vez que o número de arestas entre F e NF é menor que o número de arestas entre NF e C. Nesse sentido, uma classe que está próxima de outra na hierarquia é também semanticamente mais similar a essa classe.

5.2.3 Matriz de Custo de Erro de Classificação

Essa abordagem é uma generalização das matrizes de custo utilizadas para avaliação da classificação plana. Nessa matriz, cada célula armazena o custo pré-definido para cada erro de classificação possível. Uma desvantagem dessa abordagem é justamente a definição dos custos, o que pode ser uma tarefa bastante subjetiva. Além disso, para problemas de classificação com grande número de classes, um cenário frequente em tarefas de classificação hierárquica, as dimensões dessa matriz tornam-se muito grandes.

Quando a predição é obrigatória até um nó-folha, essa matriz pode ser representada pela Tabela 7, na qual os custos de erros de classificação são especificados apenas para o nível das classes-folha. Cada célula dessa matriz recebe um valor, ϵ_{ij}, que representa o custo associado ao erro cometido quando a predição da classe associada à linha i é dada a um exemplo que pertence à classe representada pela coluna j. Para as células da diagonal principal ($i = j$), o custo associado é zero, uma vez que essas células representam as predições corretas.

Tabela 7: Matriz de custo de erro de classificação hierárquica com predição obrigatória das classes-folha.

			Classe verdadeira					
			W			NW		
			B		V	C	T	H
Classe predita			F	NF				
W	B	F	0	ϵ_{12}	ϵ_{13}	ϵ_{14}	ϵ_{15}	ϵ_{16}
		NF	ϵ_{21}	0	ϵ_{23}	ϵ_{24}	ϵ_{25}	ϵ_{26}
	V		ϵ_{31}	ϵ_{32}	0	ϵ_{34}	ϵ_{35}	ϵ_{36}
NW	C		ϵ_{41}	ϵ_{42}	ϵ_{43}	0	ϵ_{45}	ϵ_{46}
	T		ϵ_{51}	ϵ_{52}	ϵ_{53}	ϵ_{54}	0	ϵ_{56}
	H		ϵ_{61}	ϵ_{62}	ϵ_{63}	ϵ_{64}	ϵ_{65}	0

Tabela 8: Matriz de custo de erro de classificação hierárquica com predição opcional das classes-folha.

Classe predita			Classe verdadeira								
			W					NW			
				B			V		C	T	H
				F	NF						
W			0	ϵ_{12}	ϵ_{13}	ϵ_{14}	ϵ_{15}	ϵ_{16}	ϵ_{17}	ϵ_{18}	ϵ_{19}
	B		ϵ_{21}	0	ϵ_{23}	ϵ_{24}	ϵ_{25}	ϵ_{26}	ϵ_{27}	ϵ_{28}	ϵ_{29}
		F	ϵ_{31}	ϵ_{32}	0	ϵ_{34}	ϵ_{35}	ϵ_{36}	ϵ_{37}	ϵ_{38}	ϵ_{39}
		NF	ϵ_{41}	ϵ_{42}	ϵ_{43}	0	ϵ_{45}	ϵ_{46}	ϵ_{47}	ϵ_{48}	ϵ_{49}
	V		ϵ_{51}	ϵ_{52}	ϵ_{53}	ϵ_{54}	0	ϵ_{56}	ϵ_{57}	ϵ_{58}	ϵ_{59}
NW			ϵ_{61}	ϵ_{62}	ϵ_{63}	ϵ_{64}	ϵ_{65}	0	ϵ_{66}	ϵ_{67}	ϵ_{68}
	C		ϵ_{71}	ϵ_{72}	ϵ_{73}	ϵ_{74}	ϵ_{75}	ϵ_{76}	0	ϵ_{78}	ϵ_{79}
	T		ϵ_{81}	ϵ_{82}	ϵ_{83}	ϵ_{84}	ϵ_{85}	ϵ_{86}	ϵ_{87}	0	ϵ_{89}
	H		ϵ_{91}	ϵ_{92}	ϵ_{93}	ϵ_{94}	ϵ_{95}	ϵ_{96}	ϵ_{97}	ϵ_{98}	0

Se o problema de classificação permite predição opcional nas classes-folha, a matriz de custos deve ser ajustada de modo que os custos associados aos nós internos também estejam presentes na matriz — Tabela 8.

A partir dessa matriz de custos, é relativamente trivial calcular as medidas anteriormente apresentadas. Isso pode ser feito de diversas maneiras, tais como:

- custo de erro de classificação uniforme: ϵ_{ij} tem o mesmo valor para todo i e j, com $i \neq j$
- erro de classificação para medidas baseadas em distância: cada custo é calculado a partir do número de arestas entre a classe predita e a classe verdadeira, de modo que o erro entre classes irmãs seja igual a 2.
- custo do erro de classificação baseado em semântica: cada valor deve ser especificado de acordo com a distância semântica entre cada par de classes.

5.2.4 Medidas Baseadas na Hierarquia

As medidas apresentadas até agora não satisfazem todos os pré-requisitos elaborados por Kiritchenko et al. (2005, 2006) (ver Seção 5.2). Em virtude desse fato, esses autores propuseram uma nova maneira de avaliar a performance de classificadores hierárquicos. Esse método de avaliação consiste em calcular as medidas de precisão e *recall* que respeitam os pré-requisitos por eles propostos e a hierarquia de classe. Essas medidas, chamadas de *hierarchical precision* (hP) e *hierarchical recall* (hR), consideram as relações de generalização e especialização da estrutura hierárquica, pois utilizam informações de nós ancestrais das classes verdadeira e predita para o cálculo.

Como pontos positivos dessas medidas, pode-se citar o fato de serem relativamente simples de implementar, respeitarem os pré-requisitos, serem independentes de parâmetros e formuladas de modo que podem ser utilizadas

também para problemas hierárquicos multirrórulo nos quais cada exemplo pode ser classificado com mais de uma classe em cada nível (com múltiplos caminhos na hierarquia) e também com estrutura de grafos. Outra vantagem de extrema importância é o fato de serem extensões de medidas já muito testadas e validadas pela comunidade científica em problemas de classificação plana, o que garante maior consistência na análise da performance de classificadores hierárquicos por meio dessas medidas.

Considerando tarefas de classificação hierárquica que permitem a predição de apenas uma classe por nível (apenas um caminho), o valor de precisão da predição para cada exemplo é calculado por meio da divisão do número de ancestrais comuns das classes predita e verdadeira. Utilizando a notação de conjuntos, pode-se calcular esse valor de precisão por meio da Equação 30.

$$hP_{E_i} = \frac{|Z_i \ \cap \ Y_i|}{|Z_i|} \tag{30}$$

De maneira análoga, o *recall* pode ser calculado conforme a Equação 31

$$hR_{E_i} = \frac{|Z_i \ \cap \ Y_i|}{|Y_i|} \tag{31}$$

onde Z_i é o conjunto de classes preditas para o exemplo de teste i, enquanto que Y_i é o conjunto de classes verdadeiras desse exemplo. É importante lembrar que o nó raiz não é considerado para o cálculo dessas medidas, uma vez que, por definição, é comum a todos os exemplos.

Para se ter uma avaliação global da performance do classificador é necessário calcular a média desses valores de precisão e *recall*. Contudo, é importante observar que o valor médio calculado com base nas Equações 30 e 31 é baseado na predição individual de cada exemplo.

Existem outros métodos para avaliar a performance global dos classificadores, por exemplo as médias micro e macro. Para se obter esses valores, é necessário primeiramente calcular os valores de precisão e *recall* locais, *i.e.*, para cada classe da taxonomia. Uma possível abordagem para isso é construir uma matriz de confusão para cada classe (ver Seção 5.1) e, a partir dessas matrizes locais, calcular as médias micro e macro.

Considerando uma matriz de confusão para cada classe y_j, as médias micro para as medidas de precisão e *recall* são calculadas, respectivamente, por meio das Equações 32 e 33.

$$Pr^{\mu} = \frac{\sum_{y_j \in L} VP_{y_j}}{\sum_{y_j \in L} VP_{y_j} + FP_{y_j}} \tag{32}$$

$$Re^{\mu} = \frac{\sum_{y_j \in L} VP_{y_j}}{\sum_{y_j \in L} VP_{y_j} + FN_{y_j}} \tag{33}$$

Os valores de *macro-average*, por outro lado, são calculados a partir da média dos valores de cada medida obtida individualmente para cada classe do problema, conforme as Equações 34 e 35.

$$Pr^M = \frac{1}{|L|} \times \sum_{y_j \in L} Pr_{y_j} \tag{34}$$

$$Re^M = \frac{1}{|L|} \times \sum_{y_j \in L} Re_{y_j} \tag{35}$$

5.3 Erro Majoritário Hierárquico

Pesquisadores que trabalham com o tema de classificação hierárquica, frequentemente reportam os resultados de suas pesquisas confrontando suas propostas com métodos de classificação plana, aplicados a problemas de classificação hierárquica, *i.e.*, os algoritmos de classificação plana são utilizados como base de comparação, limitando a compreensão e percepção do real ganho dos classificadores hierárquicos com relação aos classificadores planos. Outro fator importante é que a comparação com métodos de classificação plana não mostram a evolução dos métodos hierárquicos. Nesse contexto, seria mais interessante que fossem utilizados como base de comparação outras medidas, como alguma medida de erro majoritário ou mesmo o classificador hierárquico mais simples possível.

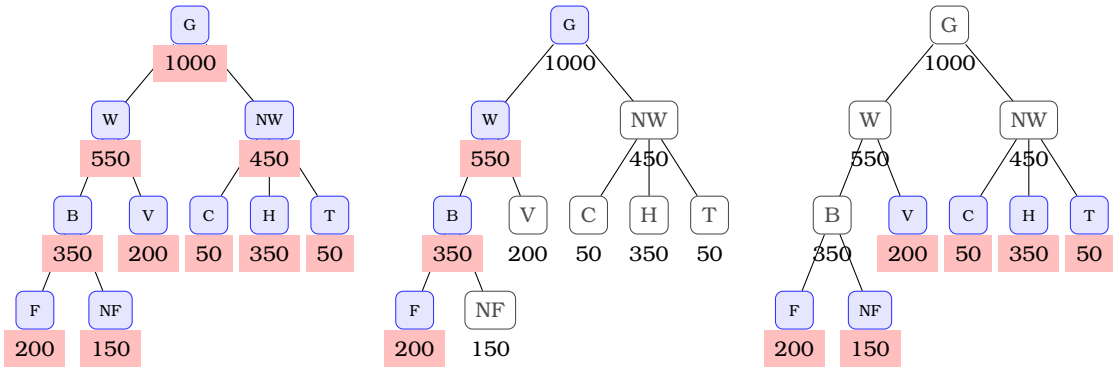
Ainda com relação ao métodos base para comparação, o primeiro ponto a discutir é como calcular a taxa de erro majoritário no contexto de classes hierárquicas. Uma abordagem para esse cálculo é considerar uma estratégia *top-down*, obtendo-se primeiramente o valor no primeiro nível e, então, seguir ao próximo nível considerando apenas a subárvore com raiz na classe majoritária do nível analisado.

Para ilustrar esse método, considere a Figura 19, na qual o número de exemplos associados a cada classe está apresentado diretamente abaixo do nó da classe. A partir dessa hierarquia, obtém-se a classe F como a classe majoritária seguindo a estratégia acima descrita — Figura 19(b). A taxa de erro pode ser calculada por meio da divisão da quantidade de exemplos associados à classe majoritária pelo número total de exemplos conforme a Equação 36,

$$Err_{major} = 1 - P(y_{major}) \tag{36}$$

onde $P(y_{major})$ é a probabilidade da classe majoritária, aproximada pela frequência da classe no conjunto de exemplos.

Entretanto, a classe majoritária pode ser definida de outras maneiras, por exemplo considerando apenas as classes-folha — Figura 19(c) — ou, então, por



(a) Distribuição dos exemplos

(b) Classe majoritária: F

(c) Classe majoritária: H

Figura 19: Erro majoritário hierárquico para predição MLN.

meio de alguma outra heurística pré-definida. Observe que para o mesmo conjunto de dados e hierarquia de classes, a maneira como a classe majoritária é identificada pode resultar em diferentes valores para o erro majoritário. Por exemplo, se utilizarmos apenas a frequência das classes-folha para definir a classe majoritária, considerando a Figura 19, teremos como classe majoritária a classe H e $Err_{major} = 65\%$, enquanto que considerando F como a classe majoritária (200/1000) o erro majoritário é $Err_{major} = 80\%$.

Outro detalhe em relação ao cálculo do erro majoritário, é como identificar a classe majoritária quando o problema admite exemplos com classe mais específica interna, *i.e.*, predição opcional nas folhas (NMLN). Considere uma outra distribuição dos exemplos na hierarquia ilustrada na Figura 20, na qual os números apresentados em cada nó da hierarquia representam *o número de exemplos explicitamente anotados com essa classe*, ao passo que os valores nas arestas representam *o número de exemplos anotados com alguma das classes na subárvore abaixo da aresta*. Por exemplo, ao todo 200 exemplos são associados à classe B, F ou NF, dos quais 100 estão anotados explicitamente com a classe B e os outros 100 distribuídos entre as classes F e NF. Observe que, nesse caso, muitos exemplos não devem ser classificados nos nós-folha. Assim, diferentemente do caso ilustrado na Figura 19, as classes internas podem ter mais exemplos a elas atribuídas do que a soma dos exemplos associados às suas respectivas classes filhas. Nesse caso, a classe majoritária poderia ser escolhida de acordo com o número de exemplos associados exclusivamente a cada classe com exceção da raiz. Para esse exemplo (Figura 20), a classe majoritária seria W ou NW (300/1000), uma vez que as duas possuem o mesmo número de exemplos exclusivos. O erro majoritário para esse exemplo seria de 70%.

O fator mais importante a ser observado é a necessidade de uma base de comparação bem definida, e que permita que os valores de qualidade da

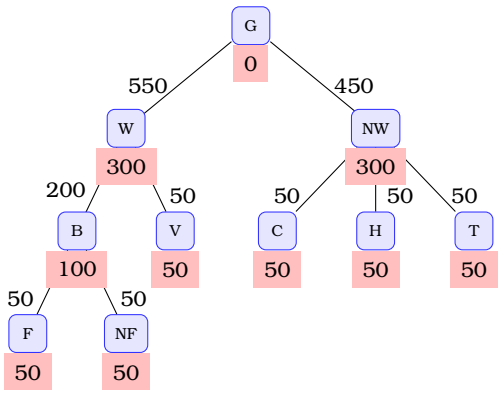


Figura 20: Erro majoritário hierárquico para predição NMLN.

predição hierárquica obtidos por meio das medidas de avaliação possam ser facilmente interpretados, *i.e.*, deve ser possível intuir o ganho real obtido pelo algoritmo com relação ao erro majoritário do problema. Assim, existe a necessidade de uma medida capaz de capturar essa taxa de erro nos diferentes cenários de classificação hierárquica.

6 Considerações Finais

Neste trabalho foram apresentados diversos conceitos relacionados à tarefa de classificação hierárquica e métodos de avaliação de algoritmos para essa tarefa. Alguns aspectos importantes foram descritos, como as principais abordagens para solucionar o problema de classificação hierárquica e dois novos métodos que propomos neste trabalho. Além disso, no caso de algoritmos que fazem uso de classificadores locais binários, um fator muito importante observado é a preparação do conjunto de treinamento para a construção de cada classificador local. Algumas heurísticas para realizar essa tarefa foram apresentadas neste trabalho.

Como mencionado, as abordagens para solução do problema de classificação hierárquica são três:

(a) plana (*naive*), quando o problema é transformado em um problema de classificação plana e um classificador plano é utilizado para predizer uma (ou mais) classe(s)-folha da estrutura hierárquica de classes, a partir da(s) qual(is) inferem-se as classes ancestrais;

(b) global, quando um algoritmo é projetado especificamente para resolver o problema de classificação hierárquica considerando toda a hierarquia de classes sem qualquer tipo de transformação ou decomposição do problema; e

(c) local, quando o problema de classificação hierárquica é transformado em

diversos problemas de classificação plana e a predição hierárquica final é o resultado da composição da predição de cada um dos problemas locais de classificação plana. Nesse caso, usualmente é utilizado *um classificador binário por nó, exceto a raiz* (HBR) ou *um classificador multiclasse em cada nó não-folha da hierarquia* (HMR).

Também é importante ressaltar que neste trabalho foram propostos dois algoritmos que fazem parte da abordagem local para classificadores hierárquicos. No primeiro caso, a idéia básica é construir um modelo com *um classificador local por nó pai com estratégia de multiplicação* e utilizar a predição dos classificadores filhos para auxiliar ou reforçar a predição dos classificadores pais e, com isso, tentar minimizar erros de classificação. No segundo método, *modelo híbrido de cooperação entre classificadores binários e multiclasse*, por outro lado, a idéia é evitar que uma classe seja predita por falta de opção, uma vez que um classificador multiclasse (nesse caso denominado primário) sempre vai predizer uma das suas classes alvo, mesmo que o exemplo não pertença a nenhuma delas. Assim, com o auxílio de um classificador secundário (nesse caso binário) a predição do classificador primário é validada e reforçada ou corrigida dependendo da predição dos classificadores secundários.

Para a utilização do segundo modelo proposto neste trabalho, assim como todos os modelos que fazem uso de classificadores binários locais, é necessário definir quais exemplos serão utilizados como exemplos positivos e negativos para a construção de cada classificador local. Para isso, diversas heurísticas foram discutidas. Dentre as heurísticas apresentadas neste trabalho, seis podem ser categorizadas como *baseadas em conjuntos* e duas como *baseadas em distâncias*. Porém, outras heurísticas podem ser desenvolvidas. Não é de nosso conhecimento nenhum trabalho que tenha avaliado o impacto da utilização de cada uma dessas heurísticas em diferentes domínios. Apenas dois trabalhos foram publicados com análises parciais e resultados inconclusivos (Eisner et al., 2005; Fagni e Sebastiani, 2007). Nesse sentido, como uma das atividades futuras pretendemos elaborar outras heurísticas e avaliá-las em conjunto com as heurísticas apresentadas neste trabalho.

No que diz respeito à avaliação dos métodos de classificação hierárquica, muitas medidas foram propostas na literatura. Porém, nenhuma delas é completamente aceita pela comunidade científica. O que se observa é que em muitos trabalhos os pesquisadores propõem novas medidas que apenas eles utilizam. Além disso, muitas das medidas de avaliação propostas não satisfazem os pré-requisitos básicos para avaliação da classificação hierárquica. Mais recentemente, algumas pesquisas têm mostrado uma tendência para adoção das medidas de avaliação propostas por Kiritchenko et al. (2005) —

Hierarchical Precision (hP) e *Hierarchical Recall (hR)*. Como outra atividade futura, pretendemos avaliar o poder de predição de diversos algoritmos de classificação hierárquica considerando as diferentes medidas apresentadas neste trabalho. O objetivo dessa atividade é observar e analisar as diferenças nos resultados obtidos. Além disso, um método base para comparação dos algoritmos, bem definido e robusto, deve ser utilizado. Para as futuras análises desses métodos e medidas de avaliação pretendemos utilizar o *erro majoritário hierárquico* com intuito de validar o ganho (ou degradação) real obtido pelos métodos hierárquicos.

Agradecimentos

Agradecemos aos colegas Thaís Rachel Lucca e Everton Alvares Cherman pela valiosa ajuda e revisão do texto deste trabalho.

Referências

Barbedo, J. G. A. e Lopes, A. (2007). Automatic genre classification of musical signals. *EURASIP J. Appl. Signal Process.*, 2007(1):157–157. Citado na página 8.

Barutcuoglu, Z., Schapire, R. E., e G.Troyanskaya, O. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836. Citado na página 10.

Blockeel, H., Bruynooghe, M., Dzeroski, S., Ramon, J., e Struyf, J. (2002). Hierarchical multi-classification. Em *Proceedings of the First SIGKDD Workshop on Multi-relational Data Mining (MRDM-2002)*, páginas 21–35. Citado nas páginas 9, 34, and 35.

Burred, J. J. e Lerch, A. (2003). A hierarchical approach to automatic musical genre classification. Em *Proceedings of The 6th International Conference on Digital Audio Effects (DAFx)*, páginas 8–11. Citado na página 8.

Cai, L. e Hofmann, T. (2007). Exploiting known taxonomies in learning overlapping concepts. Em *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical intelligence*, páginas 714–719, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Citado na página 9.

Ceci, M. e Malerba, D. (2003). Hierarchical classification of html documents with webclassii. Em *Proceedings of the 25th European conference on IR research*, ECIR'03, páginas 57–72, Berlin, Heidelberg. Springer-Verlag. Citado nas páginas 11 and 18.

Clare, A. e King, R. D. (2001). Knowledge discovery in multi-label phenotype data. Em *PKDD '01: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, páginas 42–53, London, UK. Springer-Verlag. Citado nas páginas 1 and 32.

Costa, E. P., Lorena, A. C., Carvalho, e Freitas, A. A. (2007). A review of performance evaluation measures for hierarchical classifiers. Em *Evaluation Methods for Machine Learning II: papers from the 2007 AAAI Workshop*, páginas 1–6. AAAI Press. Citado nas páginas 34 and 36.

Eisner, R., Poulin, B., Szafron, D., Lu, P., e Greiner, R. (2005). Improving protein function prediction using the hierarchical structure of the gene ontology. Em *In Proc. IEEE CIBCB*, páginas 1–10. Citado nas páginas 11, 12, 17, 18, and 45.

Fagni, T. e Sebastiani, F. (2007). On the selection of negative examples for hierarchical text categorization. Em *Proceedings of The 3rd Language Technology Conference*, páginas 24–28. Citado nas páginas 11, 15, 18, and 45.

Freitas, A. e de Carvalho, A. C. (2007). A tutorial on hierarchical classification with applications in bioinformatics. Em Taniar, D., editor, *Research and Trends in Data Mining Technologies and Applications*, chapter 7, páginas 175–208. IGI Global. Citado nas páginas 1, 4, 6, 7, 10, 20, 31, 34, and 39.

Greiner, R., Grove, A., e Schuurmans, D. (1997). On learning hierarchical classifications. Em *ResearchIndex; The NECI Scientifc Literature Digital Library (Online)*. `http://citeseer.nj.nec.com/article/greiner97learning.html`. Citado na página 17.

Holden, N. e Freitas, A. (2009). Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation. *Soft Computing Journal*, 13(3):259–272. Citado na página 35.

Holden, N. e Freitas, A. A. (2006). Hierarchical classification of g-protein-coupled receptors with a pso/aco algorithm. Em *Proc. IEEE Swarm Intelligence Symposium (SIS-06)*, páginas 77–84, Piscataway, NJ. IEEE Press. Citado na página 35.

Kiritchenko, S., Matwin, S., e Famili, A. F. (2005). Functional annotation of genes using hierarchical text categorization. Em *Proceedings of The BioLINK SIG: Linking Literature, Information and Knowledge for Biology*, páginas 1–4. Citado nas páginas 31, 32, 34, 38, 40, and 45.

Kiritchenko, S., Matwin, S., Nock, R., e Famili, A. (2006). Learning and evaluation in the presence of class hierarchies: Application to text categorization. Em Lamontagne, L. e Marchand, M., editors, *Advances in Artificial Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, páginas 395–406. Springer Berlin / Heidelberg. Citado na página 40.

Labrou, Y. K. (1999). Yahoo as an ontology - using yahoo categories to describe documents. Em *Proceedings of the 1999 ACM Conference on Information and Knowledge Management (CIKM'99)*. Citado na página 32.

Lipscomb, C. E. (2000). Medical subject headings (MeSH). *Bull Med Libr Assoc*, 88(3):265–266. Citado na página 1.

Lord, P. W., Stevens, R. D., Brass, A., e Goble, C. A. (2003). Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283. Citado nas páginas 35 and 38.

Metz, J. e Freitas, A. A. (2009). Extending hierarchical classification with semi-supervised learning. Em *Proceedings of the UK Workshop on Computational Intelligence*, páginas 1–6. Citado nas páginas 10 and 35.

Mladenić, D. e Grobelnik, M. (1998). Feature selection for classification based on text hierarchy. Em *Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, páginas 1–6. Citado nas páginas 10 and 11.

Monard, M. C. e Baranauskas, J. A. (2003). *Conceitos sobre aprendizado de máquina*, volume 1 of *1*, chapter 4, páginas 89–114. Manole, Barueri, SP, Brasil, 1ª ediÁ„o. Citado na página 29.

Qiu, X., Gao, W., e Huang, X. (2009). Hierarchical multi-class text categorization with global margin maximization. Em *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, páginas 165–168, Morristown, NJ, USA. Association for Computational Linguistics. Citado na página 9.

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., e Mewes, H. W. (2004). The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res*, 32(18):5539–5545. Citado na página 1.

Sasaki, M. e Kita, K. (1998). Rule-based text categorization using hierarchical categories. Em *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, páginas 2827–2830. Citado na página 32.

Schapire, R. E., Singer, Y., e Singhal, A. (1998). Boosting and Rocchio applied to text filtering. Em *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, páginas 215–223, New York, NY, USA. ACM. Citado na página 14.

Secker, A., Davies, M., Freitas, A., Timmis, J., Mendao, M., e Flower, D. (2007). An experimental comparison of classification algorithms for the hierarchical prediction of protein function. *Expert Update*, 9(3):17–22. Citado na página 10.

Secker, A., Davies, M. N., Freitas, A. A., Clark, E. B., Timmis, J., e Flower, D. R. (2010). Hierarchical classification of g-protein coupled receptors with data driven selection of attributes and classifiers. *International Journal of Data Mining and Bioinformatics*, 4(2):191–210. Citado na página 10.

Silla Jr, C. e Freitas, A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 1:1–42. Citado nas páginas 4, 9, and 10.

Singhal, A., Mitra, M., e Buckley, C. (1997). Learning routing queries in a query zone. Em *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, páginas 25–32, New York, NY, USA. ACM. Citado na página 15.

Struyf, J., Dzeroski, S., Blockeel, H., e Clare, A. (2005). Hierarchical multi-classification with predictive clustering trees in functional genomics. Em Bento, C., Cardoso, A., e Dias, G., editors, *Progress in Artificial Intelligence,*

12th Portuguese Conference on Artificial Intelligence, EPIA 2005, volume 3808 of *Lecture Notes in Computer Science*, páginas 272–283. Springer. Citado na página 32.

Sun, A., Lim, E., e Ng, W. (2003a). Performance measurement framework for hierarchical text classification. *Journal of the American Society for Information Science and Technology*, 54:1014–1028. Citado nas páginas 6 and 17.

Sun, A. e Lim, E.-P. (2001). Hierarchical text classification and evaluation. Em *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, páginas 521–528, Washington, DC, USA. IEEE Computer Society. Citado nas páginas 4, 10, 36, and 38.

Sun, A., Lim, E.-P., e Ng, W.-K. (2003b). Hierarchical text classification methods and their specification. *Kluwer International Series In Engineering And Computer Science*, 1(729):236–256. Citado na página 10.

Sun, A., Lim, E.-P., Ng, W.-K., e Srivastava, J. (2004). Blocking reduction strategies in hierarchical text classification. *IEEE Trans. on Knowl. and Data Eng.*, 16(10):1305–1308. Citado nas páginas 10 and 22.

Tikk, D., Biró, G., e Yang, J. D. (2004). A hierarchical text categorization approach and its application to frt expansion. *Australian Journal of Intelligent Information Processing Systems*, 8(3):123–131. Citado na página 10.

Toutanova, K., Chen, F., Popat, K., e Hofmann, T. (2001). Text classification in a hierarchical mixture model for small training sets. Em *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, páginas 105–113, New York, NY, USA. ACM. Citado na página 32.

Tsoumakas, G., Katakis, I., e Vlahavas, I. (2010). Mining multi-label data. Em Maimon, O. e Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, páginas 667–685. Springer US. Citado na página 10.

Vens, C., Struyf, J., Schietgat, L., Džeroski, S., e Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214. Citado nas páginas 9 and 17.

Wang, J., Shen, X., e Pan, W. (2009). On large margin hierarchical classification with multiple paths. *Journal of the American Statistical Association*, 104(487):1213–1223. Citado na página 9.

Wang, K., Zhou, S., e Liew, S. C. (1999). Building hierarchical classifiers using class proximity. Em *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, páginas 363–374, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Citado na página 36.

Wu, F., Zhang, J., e Honavar, V. (2005). Learning classifiers using hierarchically structured class taxonomies. Em Zucker, J.-D. e Saitta, L., editors, *Abstraction, Reformulation and Approximation*, volume 3607 of *Lecture Notes in Computer Science*, páginas 313–320. Springer Berlin / Heidelberg. Citado na página 4.

Xiao, Z., Dellandrea, E., Dou, W., e Chen, L. (2007). Automatic hierarchical classification of emotional speech. Em *Multimedia Workshops, 2007. ISMW '07. Ninth IEEE International Symposium on*, páginas 291–296. Citado na página 8.

A Uso das heurísticas para seleção dos conjuntos de treinamento locais

Nesta seção são apresentados exemplos de uso das heurísticas baseadas em operações de conjunto. A definição formal de cada uma dessas heurísticas é apresentada na Tabela 3 e as ilustrações de cada heurística são apresentadas na Figura 6, ambos na Seção 3.3.3.

Para ilustração de cada uma das heurísticas, considere um conjunto de treinamento com 25 exemplos que fazem parte da hierarquia de classes Glass (Figura 5 na página 11). Para cada heurística é apresentada uma tabela com a respectiva seleção dos exemplos para compor conjunto de treinamento local para a construção de cada classificador binário C_{y_j}. Por questão de simplicidade e facilidade para visualização, apenas o rótulo mais específico de cada exemplo é apresentado. Os rótulos intermediários podem ser inferidos de maneira trivial seguindo o caminho a partir do rótulo mais específico até a raiz da hierarquia de classes. Os atributos descritores dos exemplos também foram suprimidos, uma vez que o objetivo é ilustrar a utilização da heurística para definir o conjunto de treinamento para cada classificador local, para a qual os atributos descritores não são necessários. Além disso, para cada classe disposta nas colunas da Tabela 9, são associados os exemplos que fazem parte do seu conjunto de treinamento, como: $+$ para exemplos positivos e $-$ para exemplos negativos. Alguns exemplos podem ser ignorados por algumas das heurísticas. Nesses casos não há nenhum símbolo associado ao exemplo.

Tabela 9: Conjuntos de treinamento locais aplicando a heurística *Exclusive*.

E_i	Rótulo mais específico	Classes que compõem a hierarquia								
		W	NW	B	V	C	T	H	F	NF
E_1	W	$+$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
E_2	B	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$	$-$
E_3	B	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$	$-$
E_4	V	$-$	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$
E_5	V	$-$	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$
E_6	V	$-$	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$
E_7	V	$-$	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$
E_8	F	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$
E_9	F	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$
E_{10}	F	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$
E_{11}	F	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$
E_{12}	NF	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$
E_{13}	NF	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$
E_{14}	NF	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$
E_{15}	NF	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+$
E_{16}	NW	$-$	$+$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
E_{17}	C	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$	$-$
E_{18}	C	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$	$-$
E_{19}	T	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$
E_{20}	T	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$
E_{21}	T	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$
E_{22}	T	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$
E_{23}	H	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$
E_{24}	H	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$
E_{25}	H	$-$	$-$	$-$	$-$	$-$	$-$	$+$	$-$	$-$

Tabela 10: Conjuntos de treinamento locais aplicando a heurística *L-Exclusive*.

E_i	Rótulo mais específico	Classes que compõem a hierarquia								
		W	NW	B	V	C	T	H	F	NF
E_1	W	+	−	−	−	−	−	−	−	−
E_2	B		−	+	−	−	−	−	−	−
E_3	B		−	+	−	−	−	−	−	−
E_4	V		−	−	+	−	−	−	−	−
E_5	V		−	−	+	−	−	−	−	−
E_6	V		−	−	+	−	−	−	−	−
E_7	V		−	−	+	−	−	−	−	−
E_8	F		−		−	−	−	−	+	−
E_9	F		−		−	−	−	−	+	−
E_{10}	F		−		−	−	−	−	+	−
E_{11}	F		−		−	−	−	−	+	−
E_{12}	NF		−		−	−	−	−	−	+
E_{13}	NF		−		−	−	−	−	−	+
E_{14}	NF		−		−	−	−	−	−	+
E_{15}	NF		−		−	−	−	−	−	+
E_{16}	NW	−	+	−	−	−	−	−	−	−
E_{17}	C	−		−	−	+	−	−	−	−
E_{18}	C	−		−	−	+	−	−	−	−
E_{19}	T	−		−	−	−	+	−	−	−
E_{20}	T	−		−	−	−	+	−	−	−
E_{21}	T	−		−	−	−	+	−	−	−
E_{22}	T	−		−	−	−	+	−	−	−
E_{23}	H	−		−	−	−	−	+	−	−
E_{24}	H	−		−	−	−	−	+	−	−
E_{25}	H	−		−	−	−	−	+	−	−

Tabela 11: Conjuntos de treinamento locais aplicando a heurística *S-Exclusive*.

E_i	Rótulo mais específico	Classes que compõem a hierarquia								
		W	NW	B	V	C	T	H	F	NF
E_1	W	+	−							
E_2	B			+	−					
E_3	B			+	−					
E_4	V			−	+					
E_5	V			−	+					
E_6	V			−	+					
E_7	V			−	+					
E_8	F								+	−
E_9	F								+	−
E_{10}	F								+	−
E_{11}	F								+	−
E_{12}	NF								−	+
E_{13}	NF								−	+
E_{14}	NF								−	+
E_{15}	NF								−	+
E_{16}	NW	−	+							
E_{17}	C					+	−	−		
E_{18}	C					+	−	−		
E_{19}	T					−	+	−		
E_{20}	T					−	+	−		
E_{21}	T					−	+	−		
E_{22}	T					−	+	−		
E_{23}	H					−	−	+		
E_{24}	H					−	−	+		
E_{25}	H					−	−	+		

Tabela 12: Conjuntos de treinamento locais aplicando a heurística *Inclusive*.

E_i	Rótulo mais específico	Classes que compõem a hierarquia								
		W	NW	B	V	C	T	H	F	NF
E_1	W	+	−			−	−	−		
E_2	B	+	−	+	−	−	−	−		
E_3	B	+	−	+	−	−	−	−		
E_4	V	+	−	−	+	−	−	−	−	−
E_5	V	+	−	−	+	−	−	−	−	−
E_6	V	+	−	−	+	−	−	−	−	−
E_7	V	+	−	−	+	−	−	−	−	−
E_8	F	+	−	+	−	−	−	−	+	−
E_9	F	+	−	+	−	−	−	−	+	−
E_{10}	F	+	−	+	−	−	−	−	+	−
E_{11}	F	+	−	+	−	−	−	−	+	−
E_{12}	NF	+	−	+	−	−	−	−	−	+
E_{13}	NF	+	−	+	−	−	−	−	−	+
E_{14}	NF	+	−	+	−	−	−	−	−	+
E_{15}	NF	+	−	+	−	−	−	−	−	+
E_{16}	NW	−	+	−	−				−	−
E_{17}	C	−	+	−	−	+	−	−	−	−
E_{18}	C	−	+	−	−	+	−	−	−	−
E_{19}	T	−	+	−	−	−	+	−	−	−
E_{20}	T	−	+	−	−	−	+	−	−	−
E_{21}	T	−	+	−	−	−	+	−	−	−
E_{22}	T	−	+	−	−	−	+	−	−	−
E_{23}	H	−	+	−	−	−	−	+	−	−
E_{24}	H	−	+	−	−	−	−	+	−	−
E_{25}	H	−	+	−	−	−	−	+	−	−

Tabela 13: Conjuntos de treinamento locais aplicando a heurística *L-Inclusive*.

E_i	Rótulo mais específico	Classes que compõem a hierarquia								
		W	NW	B	V	C	T	H	F	NF
E_1	W	+	−	−	−	−	−	−	−	−
E_2	B	+	−	+	−	−	−	−	−	−
E_3	B	+	−	+	−	−	−	−	−	−
E_4	V	+	−	−	+	−	−	−	−	−
E_5	V	+	−	−	+	−	−	−	−	−
E_6	V	+	−	−	+	−	−	−	−	−
E_7	V	+	−	−	+	−	−	−	−	−
E_8	F	+	−	+	−	−	−	−	+	−
E_9	F	+	−	+	−	−	−	−	+	−
E_{10}	F	+	−	+	−	−	−	−	+	−
E_{11}	F	+	−	+	−	−	−	−	+	−
E_{12}	NF	+	−	+	−	−	−	−	−	+
E_{13}	NF	+	−	+	−	−	−	−	−	+
E_{14}	NF	+	−	+	−	−	−	−	−	+
E_{15}	NF	+	−	+	−	−	−	−	−	+
E_{16}	NW	−	+	−	−	−	−	−	−	−
E_{17}	C	−	+	−	−	+	−	−	−	−
E_{18}	C	−	+	−	−	+	−	−	−	−
E_{19}	T	−	+	−	−	−	+	−	−	−
E_{20}	T	−	+	−	−	−	+	−	−	−
E_{21}	T	−	+	−	−	−	+	−	−	−
E_{22}	T	−	+	−	−	−	+	−	−	−
E_{23}	H	−	+	−	−	−	−	+	−	−
E_{24}	H	−	+	−	−	−	−	+	−	−
E_{25}	H	−	+	−	−	−	−	+	−	−

Tabela 14: Conjuntos de treinamento locais aplicando a heurística *S-Inclusive*.

E_i	Rótulo mais específico	Classes que compõem a hierarquia								
		W	NW	B	V	C	T	H	F	NF
E_1	W	+	−							
E_2	B	+	−	+	−					
E_3	B	+	−	+	−					
E_4	V	+	−	−	+					
E_5	V	+	−	−	+					
E_6	V	+	−	−	+					
E_7	V	+	−	−	+					
E_8	F	+	−	+	−				+	−
E_9	F	+	−	+	−				+	−
E_{10}	F	+	−	+	−				+	−
E_{11}	F	+	−	+	−				+	−
E_{12}	NF	+	−	+	−				−	+
E_{13}	NF	+	−	+	−				−	+
E_{14}	NF	+	−	+	−				−	+
E_{15}	NF	+	−	+	−				−	+
E_{16}	NW	−	+							
E_{17}	C	−	+			+	−	−		
E_{18}	C	−	+			+	−	−		
E_{19}	T	−	+			−	+	−		
E_{20}	T	−	+			−	+	−		
E_{21}	T	−	+			−	+	−		
E_{22}	T	−	+			−	+	−		
E_{23}	H	−	+			−	−	+		
E_{24}	H	−	+			−	−	+		
E_{25}	H	−	+			−	−	+		

A.1 Exemplo completo com a heurística *S-Inclusive*

Após a escolha da heurística para seleção dos conjuntos de treinamentos locais, são construídos os classificadores locais que compõem o classificador hierárquico. A construção desses classificadores pode ser realizada em paralelo, um vez que cada classificador é construído de maneira completamente independente dos outros e atua apenas com informações locais referentes ao seu respectivo nó da hierarquia de classes. Nesse sentido, a ordem com que os classificadores locais são construídos não interfere no resultado final da classificação.

É importante lembrar que qualquer algoritmo de aprendizado supervisionado (binário) disponível pode ser utilizado para a construir os classificadores locais, por exemplo, árvores de decisão, *Suport Vector Machines*, *Naïve Bayes* ou modelos *lazy* como o k-NN.

Considere como exemplo a hierarquia de classes Glass e a heurística *S-Inclusive* para seleção dos conjuntos de treinamentos locais, os quais são apresentados na Tabela 15. Considere, também, a Figura 11 como ilustração do processo de classificação de um novo exemplo, na qual o tipo de predição utilizado é MLN.

Tabela 15: Exemplo da aplicação da heurística *S-Inclusive*: conjuntos de treinamentos para todos os classificadores locais binários.

C_{W}	
$+$	$-$
E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}, E_{13}, E_{14} e E_{15}	E_{16}, E_{17}, E_{18}, E_{19}, E_{20}, E_{21}, E_{22}, E_{23}, E_{24} e E_{25}

$C_{Non-\text{W}}$	
$+$	$-$
E_{16}, E_{17}, E_{18}, E_{19}, E_{20}, E_{21}, E_{22}, E_{23}, E_{24} e E_{25}	E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}, E_{13}, E_{14} e E_{15}

C_{B}	
$+$	$-$
E_2, E_3, E_8, E_9, E_{10}, E_{11}, E_{12}, E_{13}, E_{14} e E_{15}	E_4, E_5, E_6 e E_7

C_{V}	
$+$	$-$
E_4, E_5, E_6 e E_7	E_2, E_3, E_8, E_9, E_{10}, E_{11}, E_{12}, E_{13}, E_{14} e E_{15}

C_{F}	
$+$	$-$
E_8, E_9, E_{10} e E_{11}	E_{12}, E_{13}, E_{14} e E_{15}

$C_{Non-\text{F}}$	
$+$	$-$
E_{12}, E_{13}, E_{14} e E_{15}	E_8, E_9, E_{10} e E_{11}

C_{C}	
$+$	$-$
E_{17} e E_{18}	E_{19}, E_{20}, E_{21}, E_{22}, E_{23}, E_{24} e E_{25}

C_{T}	
$+$	$-$
E_{19}, E_{20}, E_{21} e E_{22}	E_{17}, E_{18}, E_{23}, E_{24} e E_{25}

$C_{Headamps}$	
$+$	$-$
E_{23}, E_{24} e E_{25}	E_{17}, E_{18}, E_{19}, E_{20}, E_{21} e E_{22}