UNIVERSIDADE FEDERAL DE OURO PRETO

HCAIM: Um Método de Discretização Supervisionado para o Contexto de Classificação Hierárquica

Valter Hugo Guandaline
Universidade Federal de Ouro Preto

Orientador: Prof. Dr. Luiz Henrique de Campos Merschmann

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do título de Mestre.

Ouro Preto, Fevereiro de 2016

HCAIM: Um Método de Discretização Supervisionado para o Contexto de Classificação Hierárquica

Valter Hugo Guandaline
Universidade Federal de Ouro Preto

Orientador: Prof. Dr. Luiz Henrique de Campos Merschmann

UFOP

Universidade Federal
de Ouro Preto

G913h Guandaline, Valter Hugo.
HCAIM [manuscrito]: um método de discretização supervisionado para o contexto de classificação hierárquica / Valter Hugo Guandaline. - 2016.
80f.: il.: tabs.

Orientador: Prof. Dr. Luiz Henrique de Campos Merschmann.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-graduação em Ciências da Computação.
Área de Concentração: Recuperação e Tratamento da Informação.

1. Sistemas de recuperação da informação. 2. Classificação. 3. Processamento de listas (Computadores). I. Merschmann, Luiz Henrique de Campos. II. Universidade Federal de Ouro Preto. III. Titulo.

CDU: 004.252

Catalogação: www.sisbin.ufop.br

Ata da Defesa Pública de Dissertação de Mestrado

Aos 15 dias do mês de fevereiro de 2016, às 09 horas na Sala de Seminários do DECOM no Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Luiz Henrique de Campos Merschmann (presidente e orientador), Prof. Dr. David Menotti Gomes e Prof. Dr. Ricardo Cerri**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem o mestrando **Valter Hugo Guandaline**, com o título **"HCAIM: Um Método de Discretização Supervisionado para o Contexto de Classificação Hierárquica"**. Aberta a sessão pelo presidente, coube ao candidato, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

(X) Aprovada sem recomendações

( ) Reprovada

( ) Aprovada com recomendações: _____

Banca Examinadora:

_____
Prof. Dr. Luiz Henrique de Campos Merschmann

_____
Prof. Dr. David Menotti Gomes

_____
Prof. Dr. Ricardo Cerri

_____
Prof. Dr. Luiz Henrique de Campos Merschmann
Coordenador do Programa de Pós-Graduação em Ciência da Computação
DECOM/ICEB/UFOP

Ouro Preto, 15 de fevereiro de 2016.

Dedico este trabalho à minha mãe, que não mediu esforços para que eu pudesse realizar mais esse sonho. À minha tia Carmem que sempre me incentivou. À minha namorada Danielle que sempre esteve ao meu lado.

HCAIM: Um Método de Discretização Supervisionado para o Contexto de Classificação Hierárquica

Resumo

A discretização de dados, como uma etapa da fase de pré-processamento, tem sido alvo de pesquisas em diversos trabalhos no contexto de classificação plana. Apesar da importância dos métodos de discretização para a tarefa de classificação, até onde se tem conhecimento, para problemas de classificação hierárquica, não existem na literatura propostas de métodos de discretização supervisionados que possam ser utilizados em conjunto com classificadores hierárquicos globais.

Desse modo, neste trabalho é proposto um método de discretização supervisionado para o contexto de classificação hierárquica. Este método, denominado HCAIM (*Hierarchical* CAIM), corresponde a uma adaptação do método de discretização CAIM proposto para o contexto de classificação plana.

A avaliação do método proposto foi realizada utilizando-se o método de classificação hierárquica *Global Model Naive Bayes – GMNB*. Os experimentos computacionais realizados com 8 bases de dados de bioinformática mostraram que o método HCAIM, para a maioria das bases, permitiu ao GMNB alcançar desempenho preditivo superior àqueles alcançados quando a base de dados foi pré-processada pelos métodos não supervisionados *EqualWidth* e *EqualFrequency*.

Palavras-chaves: discretização, classificação hierárquica, CAIM.

HCAIM: A Discretization Supervised Method for Context Hierarchical Classification

Abstract

The discretization of data such as a stage of pre-processing stage has been the subject of research in various studies in the context of flat classification. Despite the importance of discretization methods for the classification task, as far as is known, for hierarchical classification problems, there are proposals in the literature of supervised discretization methods that can be used in conjunction with global hierarchical classifiers.

Thus, in this paper we propose a discretization method for supervised hierarchical classification context. This method, called HCAIM (Hierarchical CAIM), corresponding to an adaptation of CAIM discretization method proposed for the flat classification context.

The evaluation of the proposed method was performed using the hierarchical classification method Global Model Naive Bayes - GMNB. Computational experiments with 8 bioinformatics databases showed that HCAIM method for most bases allowed GMNB achieve superior predictive performance to those obtained when the database is preprocessed by the methods Unsupervised *EqualWidth* and *EqualFrequency*.

Keywords: discretization, hierarchical classification, CAIM.

Declaração

Esta dissertação é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para outra qualificação nesta nem em outra universidade.

Valter Hugo Guandaline

Agradecimentos

Primeiramente agradeço a Deus, pois sem ele jamais chegaria onde estou.

A meus familiares e amigos, por todo apoio e carinho que me dedicam e principalmente pela compreensão e paciência de todos.

Agradeço aos meus pais Valter e Rufina por todo apoio e carinho que me deram e acima de tudo agradeço por todo o sacrifício que fizeram para que eu pudesse chegar até aqui.

A minha tia Carmem que sempre foi uma mãe para mim, agradeço por todo apoio, carinho e por vir de tão longe para me ver sempre que pôde.

Aos meus amigos Guilherme, Evandro, Vinicius, Eduardo e Marcio agradeço pela compreensão que tiveram, a força que me deram e por me mostrarem que grandes amizades continuam crescendo mesmo à distância.

A minha companheira Danielle, agradeço por todo apoio, amor e carinho que me deu e principalmente pela paciência e compreensão que teve comigo mesmo nos momentos mais difíceis. Obrigado por não desistir de mim.

Agradeço ao meu orientador Luiz por seus ensinamentos, atenção, paciência e acima de tudo por sua amizade.

Agradeço aos professores do programa Haroldo, Gustavo e Anderson pela dedicação e empenho que tiveram em transmitir seus conhecimentos.

Sumário

Lista de Figuras

Lista de Tabelas

"Run, rabbit run. Dig that hole, forget the sun. And when at last the work is done.
Don't sit down it's time to dig another one."

— Breathe, Pink Floyd

Capítulo 1

Introdução

A quantidade de dados disponível no mundo em ambientes computacionais tem aumentado consideravelmente a cada dia. Dessa forma, a crescente necessidade de ferramentas computacionais que nos auxiliem a extrair informações úteis desse grande volume de dados motivou o surgimento da área de pesquisa e aplicação em Ciência da Computação denominada Mineração de Dados (Fayyad et al., 1996).

A mineração de dados corresponde a apenas uma das etapas de um processo maior denominado Processo de Descoberta de Conhecimento em Base de Dados (*Knowledge Discovery in Database – KDD*), que também inclui o pré-processamento dos dados e o pós-processamento da informação minerada (Fayyad et al., 1996).

O principal objetivo do pré-processamento é preparar o conjunto de dados para que ele possa ser utilizado por alguma técnica de mineração de dados. A discretização é um dos processos frequentemente realizados na etapa de pré-processamento dos dados (Liu et al., 2002). O seu objetivo é transformar atributos contínuos em atributos categóricos. Essa transformação é feita associando intervalos de valores contínuos à novos valores categóricos. Assim, os métodos de discretização reduzem e simplificam os dados, tornando o aprendizado mais rápido e os resultados mais compactos (Garcia et al., 2013).

Em mineração de dados, a classificação é uma das tarefas que, devido à sua importância, desperta o interesse de muitos pesquisadores. Seu objetivo é, a partir de uma base de dados contendo instâncias com características e classes conhecidas, gerar modelos capazes de predizer a classe de novas instâncias a partir de suas características.

A maioria dos problemas de classificação abordados na literatura são considerados

problemas de classificação plana, onde cada instância é associada à uma ou mais classes pertencentes a um conjunto de classes que não possuem relacionamentos entre si. No entanto, existem problemas de classificação mais complexos onde as classes a serem preditas estão estruturadas de acordo com uma hierarquia. Esses problemas são conhecidos na literatura como problemas de classificação hierárquica (Freitas and Carvalho, 2007).

Apesar de as aplicações do mundo real geralmente envolverem atributos contínuos, alguns algoritmos de classificação lidam somente com atributos categóricos. Portanto, para viabilizar a utilização desses algoritmos, faz-se necessária a discretização dos atributos contínuos. Além disso, para alguns métodos de classificação, ainda que eles sejam capazes de lidar com os atributos contínuos, o seu desempenho preditivo melhora quando os atributos contínuos são previamente discretizados (Kurgan and Cios, 2004).

Em (Garcia et al., 2013) os autores apresentaram uma nova taxonomia e categorizaram 87 métodos de discretização supervisionados existentes na literatura. Todos esses métodos assumem que não existem relações de dependência entre as classes do problema. Até onde temos conhecimento, não existem na literatura métodos de discretização que levem em consideração os relacionamentos entre classes existentes em problemas de classificação hierárquica. Portanto, trabalhos que abordaram problemas de classificação hierárquica e necessitaram realizar a discretização dos dados, tais como (Campos Merschmann and Freitas, 2013) e (Silla Jr et al., 2009a), utilizaram métodos de discretização não supervisionados.

A hipótese levantada neste trabalho é que métodos de discretização supervisionados, pelo fato de levarem em consideração o atributo classe no momento da discretização, poderiam proporcionar melhoria no desempenho preditivo de classificadores hierárquicos. Portanto, isso motivou a proposta de desenvolvimento de um método de discretização supervisionado para o contexto da classificação hierárquica.

A proposta aqui apresentada corresponde a uma adaptação do método CAIM (Kurgan and Cios, 2004), originalmente proposto para o contexto de classificação plana, para o contexto da classificação hierárquica. Essa adaptação, denominada HCAIM (Hierarchical CAIM), além de utilizar o atributo classe no processo de discretização, consegue levar em consideração a hierarquia de classes do problema.

Uma vez que os métodos de discretização não supervisionados *EqualWidth* e *EqualFrequency* vem sendo utilizados em trabalhos de classificação hierárquica, eles foram adotados como base de comparação com o método proposto HCAIM. A qualidade da discretização realizada por cada um dos métodos avaliados neste trabalho foi medida a

partir do método de classificação hierárquica *Global Model Naive Bayes – GMNB*.

Os experimentos computacionais mostraram que, na maioria dos casos, o desempenho preditivo do classificador hierárquico GMNB para uma base discretizada pelo HCAIM foi superior àquele obtido quando a mesma base de dados foi discretizada pelos métodos não supervisionados *EqualWidth* e *EqualFrequency*.

O restante deste trabalho está organizado como especificado a seguir. O Capítulo 2 traz um referencial teórico sobre os métodos de classificação hierárquica e métodos de discretização. Na sequência, o Capítulo 3 apresenta a adaptação do método de discretização supervisionado CAIM para o contexto de classificação hierárquica. Em seguida, o Capítulo 4 apresenta os experimentos computacionais realizados para avaliação do método proposto neste trabalho. Por fim, o Capítulo 5 apresenta as conclusões deste trabalho e as propostas de trabalhos futuros.

Capítulo 2

Referencial Teórico

Neste capítulo, é feito um levantamento sobre os métodos de classificação hierárquica e métodos de discretização existentes na literatura. Inicialmente, a Seção 2.1 apresenta as características dos problemas de classificação hierárquica e as diferentes formas que os classificadores lidam com a hierarquia de classes. Em seguida, a Seção 2.2 apresenta um estudo sobre discretização e a forma como os métodos de discretização podem ser categorizados. Por fim, na Seção 2.2.2, o método de discretização supervisionado CAIM, é apresentado com maiores detalhes pois, neste trabalho, foi proposta sua adaptação ao contexto de classificação hierárquica.

2.1 Classificação Hierárquica

A maioria dos problemas de classificação abordados na literatura são problemas de classificação plana, onde cada instância da base de dados está associada a uma ou mais classes, sendo que essas classes não possuem relacionamentos entre si. No entanto, existe um grande número de problemas reais em que as classes estão relacionadas de acordo com uma hierarquia. Esses problemas são conhecidos na literatura como problemas de classificação hierárquica (Freitas and Carvalho, 2007).

Em um problema de classificação hierárquica, os relacionamentos entre as classes são representados por uma estrutura hierárquica, que pode ser uma árvore ou um grafo acíclico direcionado (*Directed Acyclic Graph – DAG*). Como mostra a Figura 2.1, a principal diferença entre essas estruturas é que, enquanto em uma árvore um nó (classe) está associado a no máximo um nó (classe) pai, em um DAG um nó pode ter mais do

que um nó pai (classe 1-2.1 da Figura 2.1(b).

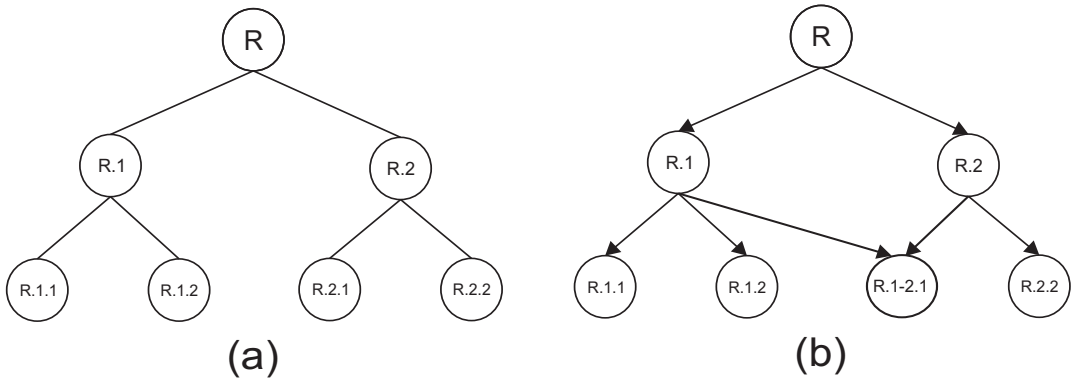

Figura 2.1: (a) Exemplo de uma árvore. (b) Exemplo de um DAG.

Considerando a hierarquia de classes da Figura 2.1(a), é natural considerar que uma instância pertencente à classe 2.1, também pertence à classe 2.

De acordo com Freitas and Carvalho (2007) e Sun and Lim (2001), os métodos de classificação hierárquica diferem em uma série de aspectos. O primeiro aspecto refere-se ao tipo de estrutura com a qual o método é capaz de lidar. Essa estrutura pode ser uma árvore ou um grafo acíclico direcionado (*DAG*). No caso deste trabalho, a estrutura hierárquica das classes corresponde sempre a uma árvore.

O segundo aspecto está relacionado à profundidade da execução da classificação na hierarquia. Isto é, um método pode realizar predições utilizando somente as classes dos nós folha da hierarquia (*Mandatory Leaf-Node Prediction – MLNP*) ou classes referentes a qualquer nó (interno ou folha) da estrutura hierárquica (*Non-Mandatory Leaf-Node Prediction – NMLNP*). Neste trabalho, é considerado o cenário onde a classificação pode ser feita utilizando qualquer classe da estrutura hierárquica (*NMLNP*).

O terceiro aspecto está relacionado ao número de classes (ramos da estrutura hierárquica) que um método é capaz de atribuir a uma instância. Como mostra a Figura 2.2, um método pode ser capaz de predizer múltiplas classes para uma determinada instância (Figura 2.2(b) - multirrótulo), desse modo envolvendo múltiplos ramos da hierarquia de classes (*multiple paths of labels*), ou somente uma classe (Figura 2.2(a) - monorrótulo), a qual estará vinculada a somente um ramo da hierarquia de classes (*single path of labels*). O método proposto neste trabalho lida com a classificação monorrótulo.

Por fim, o quarto aspecto está relacionado ao tipo de abordagem que o classificador utiliza para explorar a estrutura hierárquica. Segundo Silla Jr and Freitas (2011) existem três tipos de abordagens: (i) abordagem por classificação plana, na qual a hierarquia

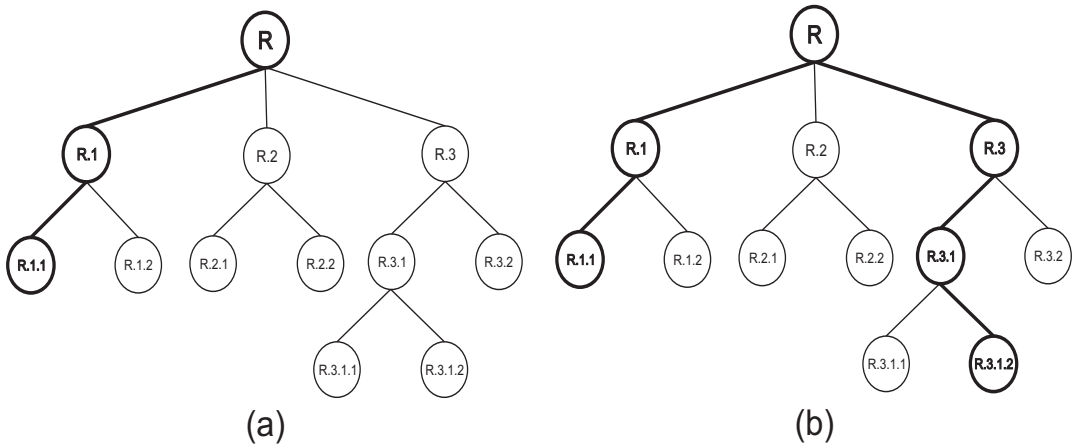

Figura 2.2: (a) Monorrótulo - Somente um ramo associado a uma dada instância (ramo em destaque). (b) Multirrótulo - Mais de um ramo associado a uma dada instância (ramos em destaque).

de classes é ignorada e as predições são realizadas considerando-se somente as classes dos nós folha da estrutura hierárquica; (ii) abordagens locais, onde são utilizados diversos classificadores planos tradicionais; (iii) abordagem global, onde um único modelo é construído considerando toda a hierarquia de classes durante a execução do método de classificação. O método de discretização proposto neste trabalho tem como objetivo adequar as bases de dados para serem utilizadas por classificadores hierárquicos globais, dado que para abordagem local, métodos de discretização supervisionados projetados para o cenário de classificação plana podem ser utilizados.

A seguir são apresentadas as diferentes abordagens utilizadas pelos classificadores para lidar com os problemas de classificação hierárquica.

2.1.1 Abordagem por classificação plana

A abordagem por classificação plana é uma maneira simplificada de lidar com problemas de classificação hierárquica, pois consiste basicamente em ignorar totalmente a hierarquia de classes. Nessa abordagem a predição ocorre considerando-se somente os nós folha da hierarquia.

Apesar de ser simples, essa abordagem tem a desvantagem de precisar gerar um classificador que lide com um grande número de classes (todos os nós folhas) sem explorar as informações de relacionamento entre as classes. Na Figura 2.3 a área delimitada pela linha tracejada representa o modelo de classificação.

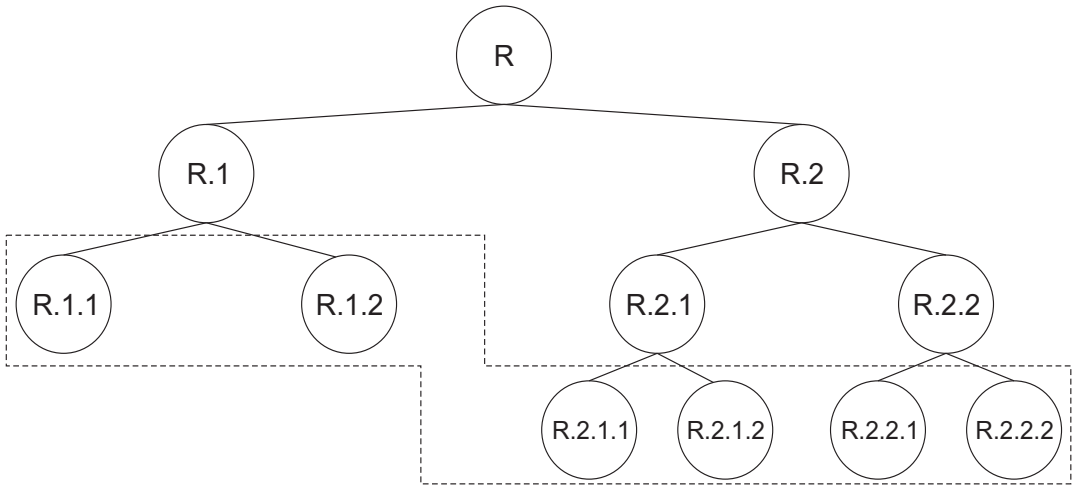

Figura 2.3: Abordagem plana para lidar com problemas de classificação hierárquica.

2.1.2 Abordagem Local

Na abordagem local, onde vários classificadores são construídos, a hierarquia de classes é explorada através da perspectiva local de cada um dos classificadores. As desvantagens desse tipo de abordagem são: a possibilidade de propagação de erros (um erro de classificação nos níveis superiores da hierarquia pode se propagar para os níveis mais profundos) e a geração de inconsistências entre os resultados produzidos pelos diferentes classificadores.

Segundo Silla Jr and Freitas (2011) os classificadores podem ser agrupados, de acordo com as diferentes maneiras de se utilizar a informação local, nas seguintes categorias: abordagem local por nó, abordagem local por nó pai e abordagem local por nível. A seguir, são apresentadas as diferentes abordagens locais.

Abordagem local por nó

Na abordagem local por nó, é gerado um classificador binário para cada nó da hierarquia (exceto para o nó raiz), assim como mostrado na Figura 2.4, onde cada quadrado pontilhado representa um classificador binário. Cada classificador binário prediz se a instância pertence ou não à classe com a qual ele está associado.

Essa abordagem permite que uma instância seja associada a classes de diferentes ramos (caminhos) da hierarquia de classes, o que caracteriza uma inconsistência quando lidamos com um problema de classificação hierárquica monorrótulo. Por exemplo, con-

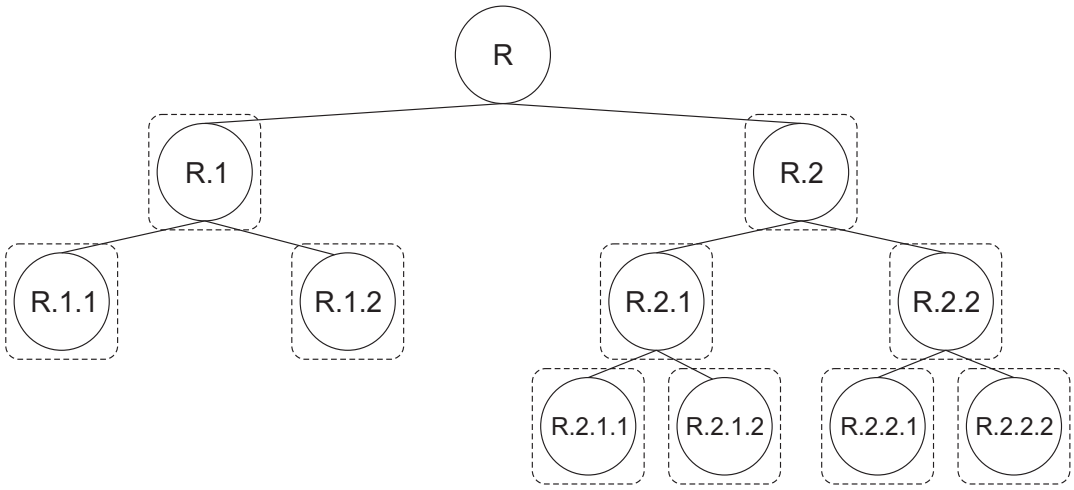

Figura 2.4: Abordagem local por nó (os quadrados tracejados com cantos arredondados representam os classificadores binários).

siderando a estrutura da Figura 2.4, uma instância pode receber resultado positivo para as classes 2, 1.2 e 2.1.1. Esse caso possui uma inconsistência entre a classe predita no nível 2 (classe 1.2) e nos níveis 1 e 3 (classes 2 e 2.1.1, respectivamente). Note que a classe 1.2 não está no mesmo ramo (caminho) da hierarquia que as classes 2 e 2.1.1.

Vários métodos foram propostos na literatura para tratar ou corrigir esse problema da inconsistência, tais como (Wu et al., 2005), (Xue et al., 2008) e (Barutcuoglu and DeCoro, 2006). Exemplos de trabalhos que utilizaram a abordagem local por nó são (Dumais and Chen, 2000), (D'Alessio et al., 2000), (Sun and Lim, 2001) e (Cesa-Bianchi and Valentini, 2009).

Abordagem local por nó pai

Na abordagem local por nó pai, um classificador plano é gerado para cada nó pai da hierarquia, como mostra a Figura 2.5, onde cada quadrado tracejado representa um classificador plano. Nesse caso, cada classificador plano, considera somente as classes associadas aos nós filhos do nó em questão. Por exemplo, dada uma nova instância para ser classificada, considerando a estrutura hierárquica apresentada na Figura 2.5, o primeiro classificador associado ao nó raiz (R) irá classificar essa nova instância como sendo de uma das classes associadas aos seus nós filhos (classe 1 ou 2).

Dando continuidade ao exemplo anterior, suponha que o classificador associado ao nó raiz rotule a nova instância com a classe 2. A nova instância será então processada

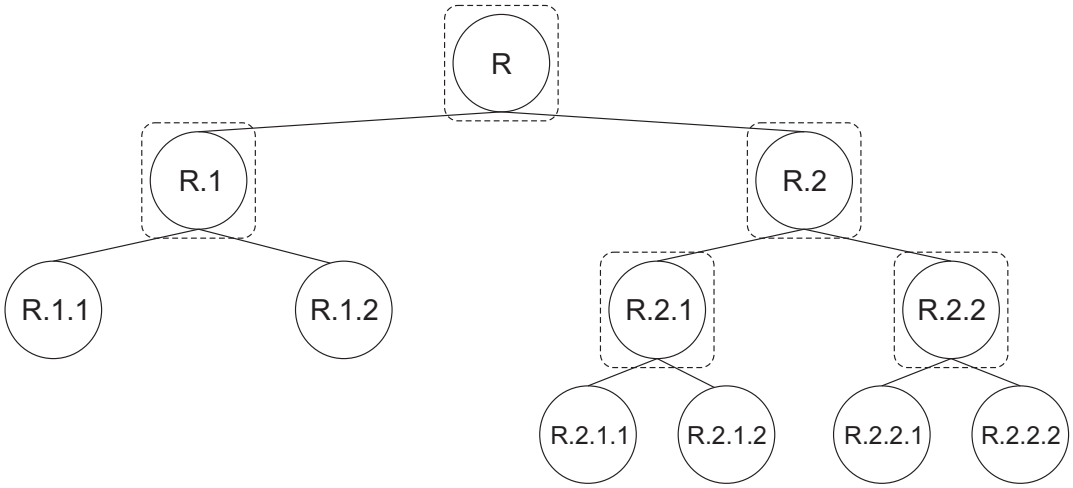

Figura 2.5: Abordagem local por nó pai (os quadrados tracejados com cantos arredondados representam os classificadores multi-classe em nós pais - predizendo suas classes filhas).

pelo classificador associado ao nó da classe 2, que, por sua vez, considerará somente as classes associadas aos seus nós filhos (classes 2.1 e 2.2) e assim por diante. Esse processo de classificação pode prosseguir até que a instância seja associada a uma classe de um nó folha ou pode para em algum nível intermediário da hierarquia.

Alguns exemplos de trabalhos que utilizaram a abordagem local por nó pai são (Koller and Sahami, 1997), (Secker et al., 2010) e (Silla Jr et al., 2009b).

Abordagem local por nível

Na abordagem local por nível, um classificador plano é gerado para cada nível da hierarquia, como mostra a Figura 2.6, onde os retângulos tracejados representam os classificadores planos. Segundo Silla Jr and Freitas (2011) esta é uma das abordagens menos difundidas na literatura.

A maior desvantagem dessa abordagem é a possibilidade da geração de inconsistências em problemas de classificação monorrótulo, assim como na abordagem local por nó. Por exemplo, considerando a estrutura hierárquica apresentada na Figura 2.6, onde temos três classificadores locais (cada um atuando em um nível da hierarquia), dada uma nova instância para ser classificada, o classificador do primeiro nível pode predizer a classe 2, o do segundo nível a classe 1.2 e o do terceiro nível a classe 2.1.2. Nesse caso, a classe predita pelo classificador do nível 2 (classe 1.2) é inconsistente com relação às classes preditas pelos classificadores dos níveis 1 e 3 (classes 2 e 2.1.2, respectivamente), pois

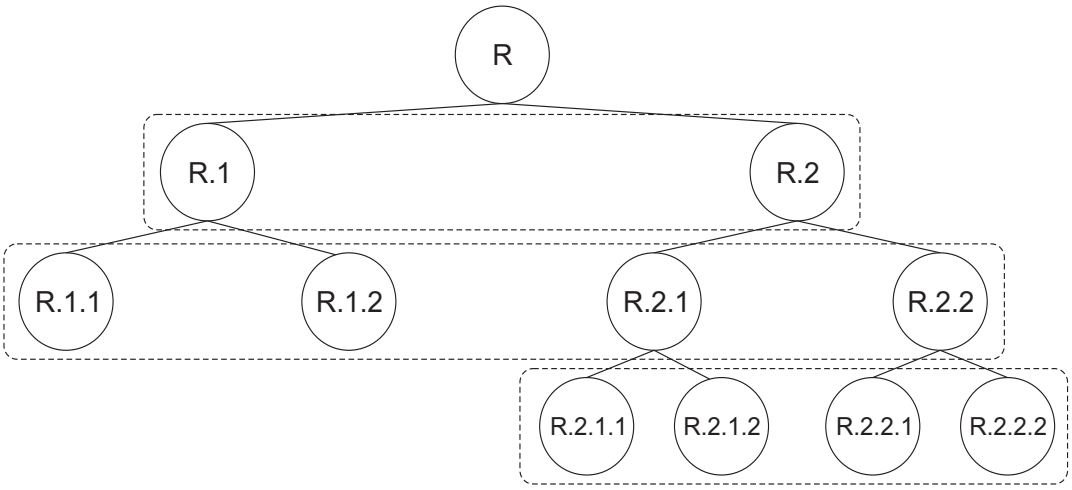

Figura 2.6: Abordagem local por nível (cada retângulo tracejado com cantos arredondados engloba as classes consideradas por cada classificador multi-classe).

as classes não pertencem ao mesmo caminho na hierarquia. Portanto, esta abordagem também necessita da aplicação de métodos para a correção de inconsistências.

2.1.3 Abordagem Global

A abordagem global apresenta uma complexidade maior em relação às outras abordagens, uma vez que o modelo de classificação é construído levando em consideração a hierarquia como um todo em uma única execução do algoritmo de classificação. Isso torna esta abordagem mais complexa. A Figura 2.7, onde o retângulo pontilhado representa o classificador, mostra como a abordagem global considera a hierarquia de classes.

Alguns exemplos de trabalhos que utilizaram a abordagem global são (Labrou and Finin, 1999), (Qiu et al., 2009) e (Silla Jr et al., 2009a).

Neste trabalho, um método de classificação hierárquica que utiliza a abordagem global, o *Global Model Naive Bayes – GMNB* (Silla Jr et al., 2009a), é utilizado para avaliar o desempenho dos métodos de discretização. Como esse método trabalha apenas com dados discretos, faz-se necessária uma etapa de pré-processamento para discretização dos dados. Uma breve descrição do *GMNB* é apresentada a seguir.

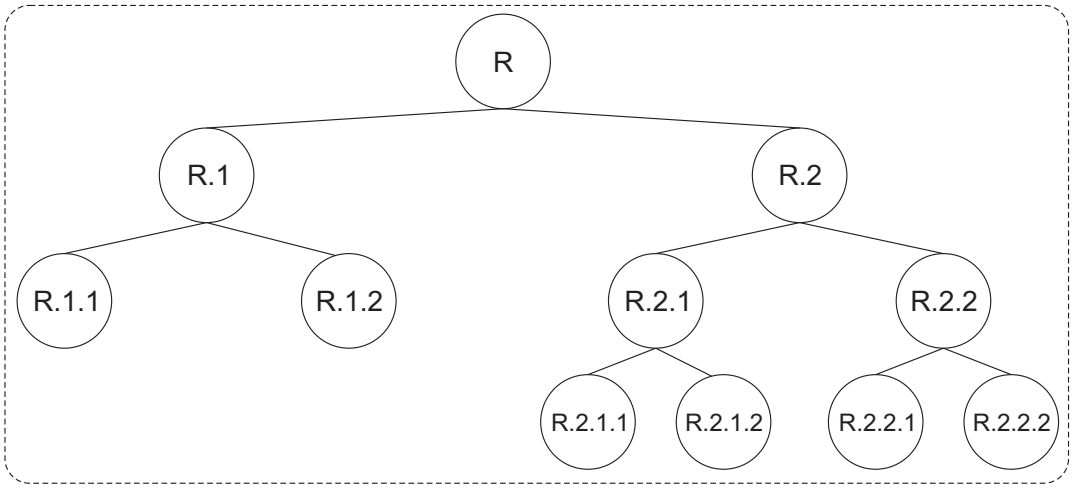

Figura 2.7: Abordagem global (o retângulo tracejado representa o classificador).

2.1.4 Global Model Naive Bayes (GMNB)

A maior parte das pesquisas realizadas em mineração de dados são focadas em problemas de classificação plana. Além disso, muitos dos trabalhos voltados para problemas de classificação hierárquica disponíveis na literatura, utilizam as abordagens de classificação hierárquica local (Silla Jr and Freitas, 2011). Dessa forma, a abordagem de classificação hierárquica global é pouco explorada.

No trabalho Silla Jr et al. (2009a) foi proposto um método de classificação que utiliza a abordagem de classificação hierárquica global. Esse método, denominado *Global Model Naive Bayes - GMNB*, é uma adaptação do classificador plano *Naive Bayes* (Duda and Hart, 1973).

Assim como no *Naive Bayes*, dada uma nova instância $X = \{x_1, x_2, \ldots, x_m\}$ para ser classificada, onde $x_1, x_2, ..., x_m$ correspondem aos valores dos atributos preditores $A_1, A_2, ..., A_m$ respectivamente, o classificador *GMNB* atribui essa nova instância à classe que tem a maior probabilidade *a posteriori* $P(C_k|\mathrm{X}) \propto P(X|C_k)P(C_k)$ onde $P(X|C_k) = \prod_{i=1}^{m}(x_i|C_k)$. Desse modo, a principal diferença para o classificador *Naive Bayes* está na forma como são calculadas as probabilidades $P(X|C_k)$ e $P(C_k)$, dado que no GMNB o cálculo dessas probabilidades leva em consideração os relacionamentos entre as classes.

Basicamente, o *GMNB* considera que qualquer instância da classe C_k pertence não somente a sua respectiva classe (C_k), mas também à todas as classes ancestrais de C_k na hierarquia. Por exemplo, se uma instância de treinamento pertence à classe $R.01.02$, ela

será utilizada no cálculo das probabilidades envolvendo a classe $R.01.02$ ($P(x_i|R.01.02)$ e $P(R.01.02)$) e também nos cálculos das probabilidades relacionadas à sua classe ancestral $R.01$ ($P(x_i|R.01)$ e $P(R.01)$).

Um raciocínio semelhante é utilizado para o cálculo das probabilidades *a priori* $P(C_k)$. Essa probabilidade é dada pela soma do número de instâncias de treinamento cuja classe é C_k com o número de instâncias de treinamento cuja classe é descendente de C_k, dividido pela quantidade total de instâncias de treinamento.

Essas alterações permitem ao método GMNB predizer classes em qualquer nível da hierarquia.

2.2 Discretização de Dados

A discretização, como uma estratégia de redução de dados (Dougherty et al., 1995), tem recebido crescente atenção dos pesquisadores nos últimos anos e tornou-se uma etapa de pré-processamento amplamente utilizada em mineração de dados (Garcia et al., 2013).

O processo de discretização transforma atributos contínuos em atributos discretos. Isso é realizado associando-se cada intervalo de valores contínuos com um valor discreto. Uma vez que a discretização é aplicada, os dados podem ser tratados como dados nominais durante qualquer processo de mineração de dados (Garcia et al., 2013).

Para o contexto de classificação plana existem vários métodos de discretização propostos na literatura.Garcia et al. (2013) categorizam os métodos de discretização de acordo com as seguintes características:

Estático ou dinâmico: Essa característica se refere ao momento em que a discretização é realizada e ao grau de dependência com o algoritmo de aprendizagem. Um discretizador dinâmico é executado no momento da construção do modelo, ou seja, o discretizador é embutido no algoritmo de aprendizagem. São exemplos de discretizadores dinâmicos o discretizador ID3 (Quinlan, 1993) e o ITFP (Au et al., 2006). Por outro lado, um discretizador estático é executado antes do processo de aprendizagem e, ao contrário do dinâmico, independe do algoritmo de aprendizagem (Hussain et al., 1999). Como exemplos de discretizadores estáticos temos (Kurgan and Cios, 2004), (Kerber, 1992) e (Liu et al., 2002).

Univariado ou multivariado: Métodos univariados consideram um atributo con-

tínuo por vez, não levando em conta a relação entre os atributos. Já os métodos multivariados podem considerar simultaneamente todos os atributos e a relação de dependência entre eles. Eles também podem discretizar um atributo por vez levando em consideração a sua relação com outros atributos. O interesse por discretizadores multivariados vem crescendo nos últimos tempos, principalmente em problemas de classificação mais complexos, onde existe muita correlação entre os atributos (Ferrandiz and Boullé, 2005), (Yang et al., 2011).

Supervisionado ou não supervisionado: A discretização é chamada supervisionada quando leva em consideração o atributo classe. Os métodos CAIM Kurgan and Cios (2004) e ChimergeKerber (1992) são exemplos de discretizadores supervisionados. Na discretização não supervisionada, o atributo contínuo é discretizado ignorando-se o atributo classe. São exemplos de discretizadores não supervisionados os métodos *EqualWidth* e *EqualFrequency* (Dougherty et al., 1995), PKID e FFD (Yang and Webb, 2009), e MVD (Bay, 2001).

Divisivo ou aglomerativo: Isto se refere ao procedimento para criar ou definir novos intervalos de discretização. Métodos divisivos realizam a discretização por meio de um processo iterativo de subdivisão do intervalo de valores inicial que é executado até que uma condição de parada seja satisfeita. Os métodos CAIM (Kurgan and Cios, 2004) e MDLP (Fayyad, 1993) são exemplos de discretizadores divisivos. Já os métodos aglomerativos iniciam com os valores do atributo contínuo particionados e, iterativamente, realizam a junção dessas partições enquanto um critério de parada não é alcançado. Os métodos ChiMerge (Kerber, 1992) e Fusinter (Zighed et al., 1998) são exemplos de métodos aglomerativos. Temos também os métodos híbridos, que alternam entre as estratégias de divisão e união das partições (Ching et al., 1995, Flores et al., 2007).

Global ou local: Para processar a discretização um método pode levar em consideração todo o conjunto de dados de um atributo ou apenas parte dele. Um discretizador global utiliza todas as informações disponíveis, enquanto um discretizador local faz uso apenas de parte das informações. (Kurgan and Cios, 2004), (Dougherty et al., 1995) e (Zighed et al., 1998) são exemplos de discretizadores globais. Todos os discretizadores dinâmicos são considerados locais, pois possuem acesso à apenas parte das informações. Exemplos de alguns discretizadores locais amplamente utilizados são ID3 (Quinlan, 1993) e MDLP (Fayyad, 1993).

Direto ou incremental: Os métodos de discretização diretos dividem os valores do atributo em um número de intervalos previamente definido pelo usuário. Por outro

lado, em métodos incrementais, o processo de discretização é realizado iterativamente até que um critério de parada seja alcançado.

Além das características já citadas, Garcia et al. (2013) categorizam os discretizadores de acordo com suas medidas de avaliação. Essas medidas de avaliação são utilizadas pelos discretizadores para comparar intervalos de valores durante o processo de discretização. As cinco principais famílias de medidas de avaliação são:

- Informação: Nesta família a entropia é a medida de avaliação mais utilizada durante a discretização (MDLP (Fayyad, 1993), ID3 (Quinlan, 1993), FUSINTER (Zighed et al., 1998)). Outra medida dessa família, derivada da teoria da informação, é o *Gini index* (Jin et al., 2009).

- Estatística: Formada por medidas que avaliam a dependência/correlação entre os atributos (Zeta (Ho and Scott, 1997), ChiMerge (Kerber, 1992), Chi2 (Liu and Setiono, 1997)), probabilísticas Boullé (2006), Wu (1996), de interdependência (CAIM (Kurgan and Cios, 2004)), coeficiente de contingência ((Tsai et al., 2008)) e outras.

- Conjuntos aproximados: Essa família é composta por métodos que avaliam a qualidade da discretização usando medidas de conjuntos aproximados ((Nguyen and Skowron, 1995)).

- *Wrapper*: Nessa família, a qualidade da discretização é realizada a partir da execução de algoritmos de classificação. A taxa de erro desses algoritmos é utilizada como medida de qualidade. Classificadores como o Naive Bayes podem ser utilizados nesse processo (NBIterative (Pazzani, 1995)).

- *Binning*: Essa categoria refere-se à ausência de uma medida de avaliação. Nesse caso, a discretização de um atributo é realizada através da criação de uma determinada quantidade pré-determinada de intervalos (*bins*). Nessa categoria, *EqualWidth* e *EqualFrequency* são exemplos de métodos amplamente utilizados.

De acordo com Garcia et al. (2013), há uma série de critérios que podem ser usados para avaliar os algoritmos de discretização. Como exemplos de critérios temos o número de intervalos gerados, o nível de inconsistência, a taxa de acerto de classificadores e o tempo de execução. No caso deste trabalho, os métodos de discretização foram avaliados a partir de um classificador.

Neste trabalho, o método de discretização supervisionado CAIM foi adaptado para o contexto de classificação hierárquica. Em Garcia et al. (2013) os autores avaliaram 30 discretizadores sobre 40 bases de dados utilizando 6 classificadores planos. Essa avaliação mostrou que o CAIM foi um dos métodos mais eficientes. Por isso, esse foi o método escolhido para ser adaptado neste trabalho.

Além disso, os métodos não supervisionados *EqualWidth* e *EqualFrequency* foram utilizados como base de comparação com o método aqui proposto. Portanto, nas seções a seguir serão apresentados mais detalhes dos métodos *EqualWidth*, *EqualFrequency* e CAIM.

2.2.1 *EqualWidth* e *EqualFrequency*

O *EqualWidth* e *EqualFrequency* estão entre os métodos de discretização não supervisionados mais simples.

O *EqualWidth* (Dougherty et al., 1995) divide o atributo contínuo em k intervalos iguais (de mesma amplitude), atribuindo a cada intervalo um rótulo diferente. O parâmetro k deve ser informado pelo usuário.

Outro método de discretização semelhante é o *EqualFrequency* (Dougherty et al., 1995). Esse método divide o atributo contínuo em k partições, de modo que, considerando uma base de dados com m instâncias, cada partição terá m/k valores adjacentes (podendo conter valores duplicados). Assim como no *EqualWidth*, o parâmetro k deve ser informado pelo usuário.

Pelo fato de não considerar o atributo classe em seu processamento, os métodos de discretização não supervisionados podem ser aplicados tanto em problemas de classificação plana quanto em problemas de classificação hierárquica.

2.2.2 CAIM

CAIM (*Class-Attribute Interdependency Maximization*) é um método de discretização supervisionado que independe de outros métodos de aprendizagem. Ele utiliza uma métrica para avaliar a interdependência entre o atributo classe e atributo a ser discretizado (Kurgan and Cios, 2004). Seu objetivo é encontrar o menor conjunto possível de intervalos minimizando a perda de interdependência entre o atributo a ser discretizado

e o atributo classe. Este método assume que o número de intervalos criados deve ser no mínimo igual ao número de classes existentes na base de dados.

Considerando uma base de dados com um conjunto de instâncias M, um conjunto de atributos numéricos F e um conjunto de classes S, onde $|M|$, $|F|$ e $|S|$ são, respectivamente, o número de instâncias, número de atributos e o número de classes. Cada instância M_k está associada à uma classe S_i onde $k \in \{1, 2, ..., |M|\}$ e $i \in \{1, 2, \ldots, |S|\}$.

Para cada atributo F_j onde $j \in \{1, 2, \ldots, |F|\}$, um algoritmo de discretização deve organizar os valores de F_j em ordem crescente e dividi-los em n intervalos limitados por pares de números da seguinte forma:

$$D = \{[d_0, d_1], (d_1, d_2], \ldots, (d_{n-1}, d_n]\} \tag{2.1}$$

No esquema D apresentado anteriormente, d_0 e d_n são, respectivamente, os valores mínimo e máximo do atributo F_j e $d_i < d_{i+1}$ pra $i \in \{0, 1, \ldots n-1\}$. Cada par de valores $(d_i,\ d_{i+1}]$ define um intervalo discreto do atributo F_j. Tal que o resultado da discretização D é chamado de esquema de discretização do atributo F_j onde temos o conjunto de pontos de corte $P = \{d_1, d_2, \ldots, d_{n-1}\}$.

Para realizar os cálculos da métrica CAIM, é utilizada uma matriz de frequência bidimensional denominada matriz de contingência, apresentada na Figura 2.8.

Classes	Intervalos					Instâncias por Classe
	$[d_0,\ d_1]$	...	$(d_{r-1}, d_r]$	...	$(d_{n-1},\ d_n]$	
C_1	q_{11}	...	q_{1r}	...	q_{1n}	M_{1+}
...	...	...	...	...	...	...
C_i	q_{i1}	...	q_{ir}	...	q_{in}	M_{i+}
...	...	...	...	...	...	...
C_s	q_{s1}	...	q_{sr}	...	q_{sn}	M_{s+}
Instâncias por Intervalo	M_{+1}	...	M_{+r}	...	M_{+n}	M

Figura 2.8: Matriz de contingência para o atributo F_j e esquema de discretização D

Na Figura 2.8, q_{ir} é o total de instâncias pertencentes à i-ésima classe contidas no

r-éssimo intervalo. M_{i+} é o total de instâncias pertencentes à i-ésima classe e M_{+r} é o total de instâncias que estão contidos no intervalo $D_r = (d_{r-1}, d_r]$ do atributo F_j.

A Equação 2.2 mede a dependência entre o conjunto de classes S e o esquema D de um dado atributo F_j, onde n é o número de intervalos, $r \in \{1, 2, \ldots, n\}$, max_r é o número máximo de instâncias pertencentes a uma classe contida no intervalo r.

$$CAIM(S, D|F_j) = \frac{\sum_{r=1}^{n} \frac{max_r^2}{M_{+r}}}{n} \tag{2.2}$$

A Equação 2.2 é utilizada como métrica para escolher, a cada iteração, o melhor ponto de corte a ser inserido no esquema de discretização. Quanto maior o valor retornado pela equação maior é dependência entre o esquema de discretização e o atributo classe. Além disso este valor e o número de intervalos criados formam o critério de parada do método.

Como encontrar o esquema de discretização ótimo sobre o espaço de todos os esquemas de discretização possíveis é um problema combinatório, o método CAIM utiliza uma abordagem gulosa que busca por uma boa aproximação do ótimo. O método CAIM pode ser divido em 3 (três) etapas: Inicialização, Avaliação e Verificação. Essas etapas são aplicadas a cada atributo contínuo F_j.

Inicialização: O primeiro passo para execução do método CAIM é identificar os valores mínimo (d_0) e máximo (d_n) do atributo F_j e criar o esquema de discretização inicial contendo apenas um intervalo limitado por estes dois valores ($D = \{[d_0, d_n]\}$), ou seja, o método inicia com um único intervalo ($k = 1$) contendo todos os valores do atributo F_j. A métrica de CAIM atribuída a este intervalo é 0 ($GlobalCaim = 0$).

Em seguida, o método inicializa o conjunto dos possíveis pontos de corte B, este conjunto é formado calculando os pontos médios entre todos os valores adjacentes do conjunto de valores distintos do atributo F_j organizados em ordem crescente. Por exemplo, se $U = \{1, 2, 3, 4, 5\}$ é o conjunto de valores distintos de F_j, então o conjunto de possíveis pontos de corte será $B = \{1,5; 2,5; 3,5; 4,5\}$. Após isso, o método passa para a próxima etapa (etapa de avaliação).

Avaliação: É uma etapa iterativa que consiste em avaliar todos os pontos de corte contidos em B enquanto o critério de parada não for satisfeito. Para cada possível ponto de corte p do conjunto B, o método cria um esquema de discretização D' a partir do

esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida, ele gera um matriz de contingência para o esquema D' e aplica a métrica de avaliação CAIM(S, $D'|F_j$), apresentado na Equação 2.2.

Após avaliar todos os possíveis pontos de corte contidos em B, o método armazena o ponto de corte $p*$ que obteve o maior valor para o critério de avaliação CAIM. Essas informações são utilizadas na etapa seguinte (etapa de verificação).

Verificação: Nesta etapa é realizada a verificação do critério de parada. O critério de parada do método CAIM é baseado em duas verificações:

1. Se o número de intervalos gerados até o momento (k) é menor do que o número de classes ($|S|$).
2. Se o valor de CAIM para o ponto de corte $p*$ é maior do que o obtido na iteração anterior ($GlobalCaim$);

O algoritmo para sempre que as duas verificações anteriores forem falsas. Nesse caso, a discretização do atributo F_j é encerrada. Caso o contrário, o método remove o ponto de corte $p*$ do conjunto B e adiciona-o no esquema D; incrementa o número de intervalos criados ($k = k + 1$); atualiza o valor da métrica CAIM para o esquema D ($GlobalCaim = caim$); e então, volta para a etapa de avaliação, para que um novo ponto de corte seja avaliado.

Para melhor compreensão do método, na seção a seguir apresentamos um exemplo prático de aplicação do CAIM.

Exemplo Prático

Para exemplificação da aplicação do método CAIM, vamos considerar a base de dados apresentada na Figura 2.9, onde a coluna # contém o número de cada instância, a coluna $Atributo1$ corresponde ao atributo contínuo a ser discretizado, a coluna $Classe$ contém a classe de cada instância. As linhas horizontais tracejadas ilustram os possíveis pontos de corte, que definem o vetor inicial B como sendo $B = \{1,5; 2,5; 3,5; 4,5\}$. Vale observar que a base de dados já está ordenada de acordo com os valores do atributo contínuo $Atributo1$.

Observe que a base de dados da Figura 2.9 contém: 12 instâncias ($|M| = 12$); 3 classes ($|S| = 3$), sendo S = {A, B, C}; e 1 Atributo contínuo ($|F| = 1$), sendo $F =$

#	Atributo1	Classe
01	1	A
02	1	A
03	2	A
04	2	A
05	2	A
06	3	B
07	3	B
08	3	B
09	4	B
10	4	C
11	5	C
12	5	C

Figura 2.9: Base de dados de exemplo

$\{Atributo1\}$. Além disso, o $Atributo1$ (F_1) contém 5 valores distintos ($q = 5$), formando o conjunto $U = \{1, 2, 3, 4, 5\}$ o conjunto de valores distintos.

O primeiro passo do método CAIM é identificar os valores mínimo (d_0) e máximo (d_n) do atributo $Atributo1$ e criar o esquema de discretização inicial contendo apenas um intervalo limitado por estes dois valores ($D = \{[1, 5]\}$). Além disto, é atribuído a este esquema de discretização inicial o valor 0 como métrica de CAIM ($GlobalCaim = 0$). O número de intervalos inicial é 1 ($k = 1$).

Em seguida o método inicializa o conjunto dos possíveis pontos de corte B. Sendo $U = \{1, 2, 3, 4, 5\}$ o conjunto de valores distintos do atributo $Atributo1$, então o conjunto de possíveis pontos de corte será $B = \{1{,}5, 2{,}5, 3{,}5, 4{,}5\}$.

Assim temos as seguintes definições necessárias para a execução do método:

- $D = \{[1, 5]\}$;
- $GlobalCaim = 0$;
- $B = \{1{,}5; 2{,}5; 3{,}5; 4{,}5\}$;
- $|S| = 3$;

- $k = 1$;

O método CAIM avalia todos os possíveis pontos de corte e insere o melhor no esquema de discretização até que o esquema contenha o número de intervalos mínimo igual ao número de classes ou enquanto a métrica CAIM continuar melhorando a cada iteração. A seguir apresentamos um passo-a-passo da execução do método.

Primeira iteração: Avaliar todos os possíveis pontos de corte contidos em B.

Primeiro possível ponto de corte ($p = 1{,}5$): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema $D' = \{[1;\ 1{,}5],\ (1{,}5;\ 5]\}$ (Figura 2.10).

Classes	Intervalos		Instâncias por Classe
	[1; 1,5]	(1,5; 5]	
A	2	3	5
B	0	4	4
C	0	3	3
Instâncias por Intervalo	2	10	12

Figura 2.10: Matriz de contingência para o esquema $D' = \{[1;\ 1{,}5],\ (1{,}5;\ 5]\}$

No quadro abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.10):

- Intervalo 1 [1; 1,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{2^2}{2} = 2 \tag{2.3}$$

- Intervalo 2 (1,5; 5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{4^2}{10} = 1,6 \tag{2.4}$$

- Total:

$$caim_1 = \frac{3,6}{2} = 1,8 \tag{2.5}$$

Segundo possível ponto de corte (p = 2,5): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema D'= {[1; 2,5], (2,5; 5]} (Figura 2.11).

Classes	Intervalos		Instâncias por Classe
	[1; 2,5]	(2,5; 5]	
A	5	0	5
B	0	4	4
C	0	3	3
Instâncias por Intervalo	5	7	12

Figura 2.11: Matriz de contingência para o esquema D' = {[1; 2,5], (2,5; 5]}

Abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.11):

- Intervalo 1 [1; 2,5] (r = 1):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{5^2}{5} = 5 \tag{2.6}$$

- Intervalo 2 (2,5; 5] (r = 2):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{4^2}{7} = 2,28 \tag{2.7}$$

- Total:

$$caim_2 = \frac{7,28}{2} = 3,64 \tag{2.8}$$

Terceiro possível ponto de corte (p = 3,5): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'.

Em seguida gerar a matriz de contingência para o esquema D'= {[1; 3,5], (3,5; 5]} (Figura 2.12).

Classes	Intervalos		Instâncias por Classe
	[1; 3,5]	(3,5; 5]	
A	5	0	5
B	3	1	4
C	0	3	3
Instâncias por Intervalo	8	4	12

Figura 2.12: Matriz de contingência para o esquema $D' = \{[1; 3,5], (3,5; 5]\}$

Abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.12):

- Intervalo 1 [1; 3,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{5^2}{8} = 3,12 \tag{2.9}$$

- Intervalo 2 (3,5; 5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{4} = 2,25 \tag{2.10}$$

- Total:

$$caim_3 = \frac{5,31}{2} = 2,68 \tag{2.11}$$

Quarto possível ponto de corte (p = 4,5): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema D'= {[1; 4,5], (4,5; 5]} (Figura 2.13).

Abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.13):

Classes	Intervalos		Instâncias por Classe
	[1; 4,5]	(4,5; 5]	
A	5	0	5
B	4	0	4
C	1	2	3
Instâncias por Intervalo	10	2	12

Figura 2.13: Matriz de contingência para o esquema $D' = \{[1; 4{,}5], (4{,}5; 5]\}$

- Intervalo 1 [1; 4,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{5^2}{10} = 2,5 \tag{2.12}$$

- Intervalo 2 (4,5; 5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{2^2}{2} = 2 \tag{2.13}$$

- Total:

$$caim_4 = \frac{4,5}{2} = 2,25 \tag{2.14}$$

Após avaliar todos os possíveis pontos de corte o método verifica se o critério de parada do CAIM está satisfeito. Neste caso k é menor que $|S|$ ($1 < 3$), portanto o método insere ponto de corte que obteve maior valor de CAIM, atualiza o valor de $GlobalCaim$, remove da lista de possíveis pontos de corte B o ponto de corte inserido no esquema D e, por fim, atualiza o valor de k. Nesta iteração o segundo ponto de corte ($p = 2{,}5$) obteve a melhor avaliação ($caim_2 = 3{,}64$):

- Esquema $D = \{[1; 2{,}5], (2{,}5; 5]\}$;
- $GlobalCaim = 3{,}64$;
- $B = \{1{,}5; 3{,}5; 4{,}5\}$;
- $k = 2$;

Segunda iteração: avaliar todos os possíveis pontos de corte contidos em B.

Primeiro possível ponto de corte ($p = 1{,}5$): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema $D' = \{[1;\ 1{,}5],\ (1{,}5;\ 2{,}5],\ (2{,}5;\ 5]\}$ (Figura 2.14).

Classes	Intervalos			Instâncias por Classe
	[1; 1,5]	(1,5; 2,5]	(2,5; 5]	
A	2	3	0	5
B	0	0	4	4
C	0	0	3	3
Instâncias por Intervalo	2	3	7	12

Figura 2.14: Matriz de contingência para o esquema $D' = \{[1;\ 1{,}5],\ (1{,}5;\ 2{,}5],\ (2{,}5;\ 5]\}$

Abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.14):

- Intervalo 1 [1; 1,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{2^2}{2} = 2 \tag{2.15}$$

- Intervalo 2 (1,5; 2,5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{3} = 3 \tag{2.16}$$

- Intervalo 3 (2,5; 5] ($r = 3$):

$$(r_3) = \frac{max_r^2}{M_{+r}} = \frac{4^2}{7} = 2,28 \tag{2.17}$$

- Total:

$$caim_1 = \frac{7,28}{3} = 2,42 \tag{2.18}$$

Segundo possível ponto de corte ($p = 3{,}5$): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema $D' = \{[1;\ 2{,}5],\ (2{,}5;\ 3{,}5],\ (3{,}5;\ 5]\}$ (Figura 2.15).

Classes	Intervalos			Instâncias por Classe
	[1; 2,5]	(2,5; 3,5]	(3,5; 5]	
A	5	0	0	5
B	0	3	1	4
C	0	0	3	3
Instâncias por Intervalo	5	3	4	12

Figura 2.15: Matriz de contingência para o esquema $D' = \{[1;\ 2{,}5],\ (2{,}5;\ 3{,}5],\ (3{,}5;\ 5]\}$

Abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.15):

- Intervalo 1 [1; 2,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{5^2}{5} = 5 \tag{2.19}$$

- Intervalo 2 (2,5; 3,5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{3} = 3 \tag{2.20}$$

- Intervalo 3 (3,5; 5] ($r = 3$):

$$(r_3) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{4} = 2,25 \tag{2.21}$$

- Total:

$$caim_2 = \frac{10,25}{3} = 3,41 \tag{2.22}$$

Terceiro possível ponto de corte ($p = 4{,}5$): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema $D' = \{[1;\ 2{,}5],\ (2{,}5;\ 4{,}5],\ (4{,}5;\ 5]\}$ (Figura 2.16).

Classes	Intervalos			Instâncias por Classe
	[1; 2,5]	(2,5; 4,5]	(4,5; 5]	
A	5	0	0	5
B	0	4	0	4
C	0	1	2	3
Instâncias por Intervalo	5	5	2	12

Figura 2.16: Matriz de contingência para o esquema $D' = \{[1;\ 2{,}5],\ (2{,}5;\ 4{,}5],\ (4{,}5;\ 5]\}$

No quadro abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.16):

- Intervalo 1 [1; 2,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{5^2}{5} = 5 \tag{2.23}$$

- Intervalo 2 (2,5; 4,5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{4^2}{5} = 3,2 \tag{2.24}$$

- Intervalo 3 (4,5; 5] ($r = 3$):

$$(r_3) = \frac{max_r^2}{M_{+r}} = \frac{2^2}{2} = 2 \tag{2.25}$$

- Total:

$$caim_3 = \frac{10,2}{3} = 3,4 \tag{2.26}$$

Após avaliar todos os possíveis pontos de corte o método verifica se o critério de parada do CAIM está satisfeito. Neste caso k é menor que $|S|$ ($2 < 3$), portanto o

método insere ponto de corte que obteve maior valor de CAIM, atualiza o valor de $GlobalCaim$, remove da lista de possíveis pontos de corte B o ponto de corte inserido no esquema D e, por fim, atualiza o valor de k. Nessa iteração o segundo ponto de corte ($p = 3{,}5$) obteve a melhor avaliação ($CAIM_2 = 3{,}41$):

- Esquema $D = \{[1;\ 2{,}5],\ (2{,}5;\ 3{,}5],\ (3{,}5;\ 5]\}$;
- $GlobalCaim = 3{,}41$;
- $B = \{1{,}5;\ 4{,}5\}$;
- $k = 3$;

Terceira iteração: avaliar todos os possíveis pontos de corte contidos em B.

Primeiro possível ponto de corte ($p = 1{,}5$): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema $D' = \{[1;\ 1{,}5],\ (1{,}5;\ 2{,}5],\ (2{,}5;\ 3{,}5],\ (3{,}5;\ 5]\}$(Figura 2.17).

Classes	Intervalos				Instâncias por Classe
	[1; 1,5]	(1,5; 2,5]	(2,5; 3,5]	(3,5; 5]	
A	2	3	0	0	5
B	0	0	3	1	4
C	0	0	0	3	3
Instâncias por Intervalo	2	3	3	4	12

Figura 2.17: Matriz de contingência para o esquema $D' = \{[1;\ 1{,}5],\ (1{,}5;\ 2{,}5],\ (2{,}5;\ 3{,}5],\ (3{,}5;\ 5]\}$

No quadro abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.17):

- Intervalo 1 [1; 1,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{2^2}{2} = 2 \tag{2.27}$$

- Intervalo 2 (1,5; 2,5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{3} = 3 \tag{2.28}$$

- Intervalo 3 (2,5; 3,5] ($r = 3$):

$$(r_3) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{3} = 3 \tag{2.29}$$

- Intervalo 3 (3,5; 5] ($r = 4$):

$$(r_4) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{4} = 2,25 \tag{2.30}$$

- Total:

$$caim_1 = \frac{10,25}{4} = 2,26 \tag{2.31}$$

Segundo possível ponto de corte ($p = 4,5$): Criar um esquema de discretização D' a partir do esquema D e insere o ponto de corte p no esquema de discretização D'. Em seguida gerar a matriz de contingência para o esquema $D' = \{[1; 2,5], (2,5; 3,5], (3,5; 4,5], (4,5; 5]\}$ (Figura 2.18).

Classes	Intervalos				Instâncias por Classe
	[1; 2,5]	(2,5; 3,5]	(3,5; 4,5]	(4,5; 5]	
A	5	0	0	0	5
B	0	3	1	0	4
C	0	0	1	2	3
Instâncias por Intervalo	5	3	2	2	12

Figura 2.18: Matriz de contingência para o esquema $D' = \{[1; 2,5], (2,5; 3,5], (3,5; 4,5], (4,5; 5]\}$

No quadro abaixo apresentamos a aplicação da métrica de avaliação CAIM (Equação 2.2) à matriz de contingência gerada (Figura 2.18):

- Intervalo 1 [1; 2,5] ($r = 1$):

$$(r_1) = \frac{max_r^2}{M_{+r}} = \frac{5^2}{5} = 5 \tag{2.32}$$

- Intervalo 2 (2,5; 3,5] ($r = 2$):

$$(r_2) = \frac{max_r^2}{M_{+r}} = \frac{3^2}{3} = 3 \tag{2.33}$$

- Intervalo 3 (3,5; 4,5] ($r = 3$):

$$(r_3) = \frac{max_r^2}{M_{+r}} = \frac{1^2}{2} = 0,5 \tag{2.34}$$

- Intervalo 3 (4,5; 5] ($r = 4$):

$$(r_4) = \frac{max_r^2}{M_{+r}} = \frac{2^2}{2} = 2 \tag{2.35}$$

- Total:

$$caim_2 = \frac{10,5}{4} = 2,62 \tag{2.36}$$

Após avaliar todos os possíveis pontos de corte o método verifica se o critério de parada do CAIM está satisfeito. Neste caso k não é menor que $|S|$ ($3 = 3$), e o maior valor de CAIM obtido ($caim_2$) não é maior que $GlobalCaim$ (2,62 < 3,41). Portanto a discretização do atributo F_1 está completa. A Figura 2.19 mostra com ficou o resultado da discretização, do atributo $Atributo1$ F_1 da base da dados original **(A)**, as linhas horizontais na base de dados discretizada **(B)** representam os pontos de corte utilizados na discretização do atributo.

Neste capítulo foi apresentado método de classificação hierárquica *GMNB*, bem com uma breve descrição sobre os tipos de abordagens de classificação hierárquica. Também foi mostrado como os métodos de discretização são categorizados. Além disso, o método de discretização supervisionado CAIM (Kurgan and Cios, 2004) foi explicado com detalhes, pois esse foi o método adaptado neste trabalho. No próximo capitulo é mostrada a adaptação do método CAIM para o contexto da classificação hierárquica.

#	Atributo1	Classe
01	1	A
02	1	A
03	2	A
04	2	A
05	2	A
06	3	B
07	3	B
08	3	B
09	4	B
10	4	C
11	5	C
12	5	C

(A)

#	Atributo1	Classe
01	[1; 2,5]	A
02	[1; 2,5]	A
03	[1; 2,5]	A
04	[1; 2,5]	A
05	[1; 2,5]	A
06	(2,5; 3,5]	B
07	(2,5; 3,5]	B
08	(2,5; 3,5]	B
09	(3,5; 5]	B
10	(3,5; 5]	C
11	(3,5; 5]	C
12	(3,5; 5]	C

(B)

Figura 2.19: **(A)**: Base de dados original. **(B)**: Base de dados discretizada.

Capítulo 3

Método Proposto

Neste capítulo é apresentada a adaptação do método de discretização supervisionado CAIM para o contexto de problemas de classificação hierárquica. Primeiramente, a Seção 3.1 apresenta as considerações iniciais para o problema de discretização em problemas de classificação hierárquica. Em seguida, na Seção 3.2 é explicado com detalhes a adaptação da métrica CAIM para o contexto de classificação hierárquica. Por fim, Seção 3.3 apresenta uma explicação detalhada do método de discretização HCAIM e seu pseudocódigo.

3.1 Considerações Iniciais

O principal problema na utilização de métodos de discretização supervisionados tradicionais (utilizados em conjunto com classificadores planos) com bases de dados relacionadas com o contexto de classificação hierárquica corresponde ao fato de esses discretizadores não serem capazes de considerar as informações dos relacionamentos entre as classes do problema.

Por exemplo, considerando a hierarquia de classes apresentada na Figura 3.1, um discretizador tradicional considera que as classes $R.01$ e $R.01.02$ são tão diferentes quanto as classes $R.01.02$ e $R.02$. No entanto, a hierarquia de classes apresentada na Figura 3.1 mostra que as classes $R.01$ e $R.01.02$ possuem uma relação de parentesco por descendência (pai-filho), o que não ocorre entre as classes $R.01.02$ e $R.02$. Neste trabalho, parte-se da hipótese de que esse tipo de informação, se considerada ao longo do processo de discretização, pode contribuir na geração de uma base de dados discretizada de melhor

qualidade para a tarefa de classificação.

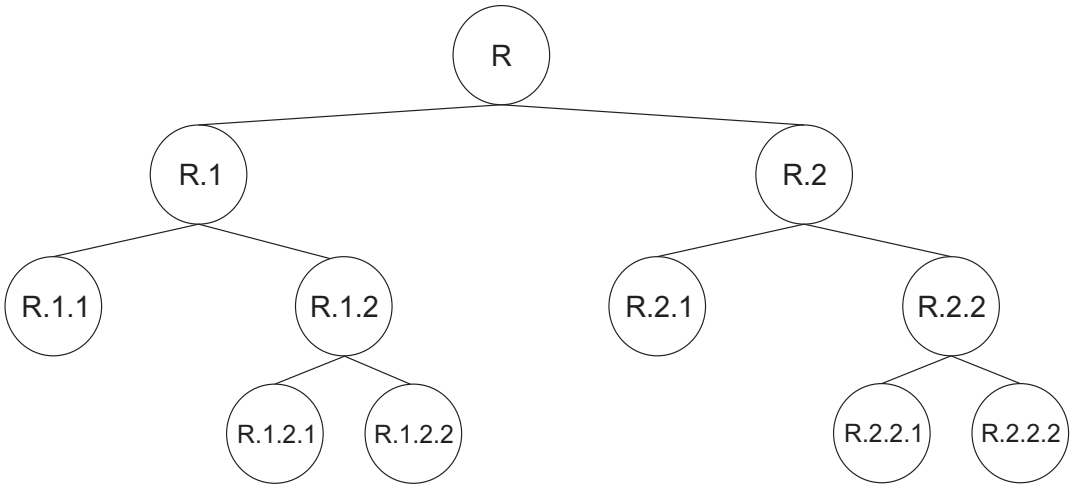

Figura 3.1: Exemplo de estrutura hierárquica

Desse modo, o foco deste trabalho é lidar com o problema de discretização de valores de atributos contínuos em bases de dados onde as classes estão estruturadas de acordo com uma hierarquia, que neste caso corresponde a uma árvore.

Portanto, o método de discretização aqui proposto considera a hierarquia de classes enquanto avalia os possíveis pontos de corte a serem inseridos no esquema de discretização. Para o exemplo apresentado anteriormente, isso significa que o método de discretização irá priorizar pontos de corte que formam intervalos contendo classes que possuem relação de parentesco por descendência (como as classes $R.01$ e $R.01.02$), ao invés de intervalos que agrupam classes que não possuem esse tipo de relacionamento (tal como as classes $R.02$ e $R.01.02$).

O método aqui proposto, denominado HCAIM (Hierarchical CAIM), corresponde a uma adaptação do método de discretização CAIM tradicionalmente utilizado no contexto de classificação plana (Kurgan and Cios, 2004). A principal adaptação para o contexto hierárquico foi realizada alterando-se a métrica de avaliação utilizada pelo método original para definição dos pontos de corte do esquema de discretização. Com essa alteração, o HCAIM consegue lidar com os relacionamentos existentes entre as classes de um problema de classificação hierárquica. A seção a seguir apresenta essa adaptação da métrica de avaliação.

3.2 Métrica de avaliação

Considerando o esquema de discretização apresentado em 2.1 (ver Seção 2.2.2), para avaliá-lo, o método CAIM verifica quão bons (puros em relação às classes associadas aos valores contidos no intervalo) são os intervalos contidos nesse esquema. Para avaliar um intervalo, o método utiliza uma métrica, também denominada CAIM, que mede a correlação entre os valores existentes no intervalo e as classes nele contidas. Essa correlação é dada por $(\frac{max_r^2}{M_{r+}})$, onde max_r é o número de ocorrências da classe mais frequente no intervalo r e M_{r+} é o total de instâncias contidas nesse mesmo intervalo. Esse cálculo permite o método CAIM:

1. Considerar o grau de pureza do intervalo (quanto mais próximo max_r for de M_{r+}, mais puro é o intervalo), e
2. Priorizar intervalos com maior número de instâncias.

Entretanto, essa métrica CAIM não consegue levar em consideração a hierarquia de classes existente em um problema onde as classes estão hierarquicamente organizadas. Por exemplo, dado um intervalo contendo 9 instâncias ($M_{r+} = 9$), sendo 3 instâncias da classe *R.02* e 6 da classe *R.02.01*, a avaliação desse intervalo segundo a métrica CAIM é dada por $6^2/9 = 4$.

Porém, no contexto hierárquico, instâncias da classe *R.02.01* também pertencem à classe *R.02*. Desse modo, nesse exemplo, isso deveria ser considerado no cálculo da métrica CAIM. Se observarmos somente até o primeiro nível da hierarquia de classes, todas a instâncias do intervalo pertencem à classe *R.02*. Nesse caso, a classe *R.02* é a única classe contida no intervalo ($max_r = 9$), o que resulta num valor para métrica CAIM igual a $9^2/9 = 9$. Ou seja, considerando somente até o primeiro nível da hierarquia de classes, o intervalo é puro. A impureza do intervalo aparece somente quando levamos em consideração o segundo nível hierárquico, quando as instâncias ficam distribuídas entre as classes *R.02* e *R.02.01*.

Portanto, neste trabalho, a métrica CAIM foi adaptada para avaliar os intervalos considerando a hierarquia de classes. Basicamente, a ideia é calcular o grau de pureza de um intervalo observando as suas classes até um determinado nível hierárquico. Esse cálculo é realizado para cada nível hierárquico de modo que o valor final da métrica para o intervalo corresponde a uma média ponderada dos valores calculados para cada um dos níveis.

A adaptação da métrica proposta neste trabalho foi denominada HCAIM (*Hierarchical CAIM*). Ela mede a dependência entre o atributo classe C e o esquema D para um dado atributo F_j levando em consideração a hierarquia de classes. A equação da métrica é apresentada a seguir.

$$HCAIM(C, D|F_j) = \frac{\sum_{r=1}^{n} \sum_{l=1}^{H_r} \frac{max_{r,l}^2}{M_{+r}} . W_{l,r}}{n}, \tag{3.1}$$

Na Equação 3.1 n é o número de intervalos, H_r é a profundidade da hierarquia de classes referente ao intervalo r, $max_{r,l}$ é o número de ocorrências da classe mais frequente no intervalo r considerando-se a hierarquia de classes até o nível l, M_{+r} é o total de instâncias contidas no intervalo r e $W_{l,r}$ é peso associado ao nível l da hierarquia de classes referente ao intervalo r.

Para exemplificar o cálculo da métrica HCAIM, considere a base de dados apresentada na Figura 3.2, onde a coluna F_1 corresponde ao atributo contínuo a ser discretizado e a coluna C representa o atributo classe do problema. Além disso, $D = \{[1, 9]\}$ será o esquema considerado nesse exemplo.

F_1	C
1	R.01
2	R.01
3	R.01
4	R.01.02.02
5	R.01.02.02
6	R.01.02.02
7	R.02
8	R.02
9	R.02

Figura 3.2: Base de dados de exemplo para cálculo de HCAIM.

Dado que a métrica HCAIM necessita da frequência das classes considerando-se os diferentes níveis da hierarquia de classes, serão apresentadas a seguir as matrizes de contingência contendo as frequências das classes considerando-as até o nível l da hierarquia. Como no exemplo em questão temos uma classe que alcança o terceiro nível da

hierarquia (*R.01.02.02*), três matrizes de contingência serão construídas.

A matriz de contingência para o nível l da hierarquia contabiliza a frequência de todas as classes até esse nível. Desse modo, as frequências são contabilizadas para as classes cujo nível mais específico é menor ou igual l e também para todas em que o nível mais específico é maior do que l. No entanto, nesse último caso, a frequência é contabilizada apenas para o nível l. Por exemplo, para uma matriz de contingência que considera até o nível 1 da hierarquia, a classe *R.01.02.02* será contabilizada como uma ocorrência da classe *R.01*. Se fosse numa matriz de contingência para o nível 2, essa mesma classe (*R.01.02.02*) seria contabilizada como uma ocorrência da classe *R.01.02*.

Seguindo o exemplo para a base de dados apresentada na Figura 3.2, a matriz de contingência para o nível 1 é mostrada na Figura 3.3. Nessa matriz, as classes da base de dados são consideradas até o primeiro nível da hierarquia ($l = 1$). Desse modo, a classe R.01.02.02 é contabilizada como classe R.01. Além dela, as classes R.01 e R.02 são contabilizadas normalmente. A partir dessa matriz, calcula-se a contribuição do nível 1 da hierarquia para o cálculo do HCAIM como sendo $\frac{max_{r,1}^2}{M_{+r}} = \frac{6^2}{9} = 4$.

Classes	Intervalos	Instâncias por Classe
	[1, 9]	
R.01	6	6
R.02	3	3
Instâncias por Intervalo	9	9

Figura 3.3: Matriz de contingência para o primeiro nível da hierarquia

A Figura 3.4 apresenta a matriz de contingência para o nível 2, ou seja, as classes são consideradas até o segundo nível da hierarquia ($l = 2$). Nesse caso, a classe R.01.02.02 é contabilizada como classe R.01.02. As demais classes, R.01 e R.02, são contabilizadas normalmente. A partir dessa matriz, calcula-se a contribuição do nível 2 da hierarquia para o cálculo do HCAIM como sendo $\frac{max_{r,2}^2}{M_{+r}} = \frac{3^2}{9} = 1$.

A Figura 3.5 apresenta a matriz de contingência para o nível 3, quando as classes são consideradas até esse nível ($l = 3$). Como não existem classes cujo nível mais específico é maior que 3, a frequência de todas as classes é contabilizada diretamente. A partir dessa matriz, calcula-se a contribuição do nível 3 da hierarquia para o cálculo do HCAIM como sendo $\frac{max_{r,3}^2}{M_{+r}} = \frac{3^2}{9} = 1$.

Classes	Intervalos	Instâncias por Classe
	[1, 9]	
R.01	3	3
R.01.02	3	3
R.02	3	3
Instâncias por Intervalo	9	9

Figura 3.4: Matriz de contingência para o segundo nível da hierarquia

Classes	Intervalos	Instâncias por Classe
	[1, 9]	
R.01	3	3
R.01.02.02	3	3
R.02	3	3
Instâncias por Intervalo	9	9

Figura 3.5: Matriz de contingência para o terceiro nível da hierarquia

No cálculo da métrica HCAIM para um determinado intervalo r, além das matrizes de contingência para cada nível hierárquico, há necessidade de se calcular os pesos $W_{l,r}$ que serão aplicados de acordo com o nível hierárquico l e a profundidade da hierarquia naquele intervalo H_r. O cálculo do peso para cada um dos níveis hierárquicos é realizado conforme a Equação 3.2.

$$W(l,r) = (H_r - l + 1)\frac{2}{H_r \times (H_r + 1)} \tag{3.2}$$

Essa equação é uma adaptação do calculo de pesos utilizado em (Chen et al., 2009). Vale ressaltar que $\sum_{l=1}^{H_r} W_{l,r} = 1$. Além disso, pode-se verificar a partir da Equação 3.2 que $W_{1,r}, W_{2,r} \ldots, W_{H_r,r}$ corresponde a uma série aritmética onde os maiores pesos são atribuídos aos níveis menos profundos da hierarquia de classes.

Voltando ao exemplo na Figura 3.2, para o intervalo $[1, 9]$, os pesos associados a cada

um dos três níveis hierárquicos são 3/6 (nível 1), 2/6 (nível 2) e 1/6 (nível 3).

Portanto, para o exemplo em questão onde $n = 1$ e $H_r = 3$, o valor da métrica HCAIM é dado por:

$$HCAIM(C, D|F_1) = \frac{\sum_{r=1}^{1}\sum_{l=1}^{3}\frac{max_{r,l}^2}{M_{+r}}.W_{l,r}}{n} = 4 \times \frac{3}{6} + 1 \times \frac{2}{6} + 1 \times \frac{1}{6} = 2,5. \quad (3.3)$$

Na próxima seção, são apresentados os detalhes do funcionamento do método HCaim.

3.3 Método HCAIM

Neste trabalho, o método de discretização CAIM foi adaptado para o contexto da classificação hierárquica. A principal alteração ocorreu na métrica de avaliação utilizada na definição dos pontos de corte do esquema de discretização (ver Seção 3.2), de modo que, no HCAIM, essa métrica leva em consideração a hierarquia de classes do problema. Além da adaptação da métrica, uma outra alteração foi realizada na maneira de inicializar o conjunto dos possíveis pontos de corte B para o processamento da discretização de um determinado atributo.

O pseudocódigo do método HCAIM é apresentado na Figura 3.6. Como entrada de dados, o algoritmo recebe uma base de dados composta por N instâncias, S classes distintas e atributos contínuos F_i. Basicamente, para cada atributo F_i a ser discretizado, o algoritmo executa duas etapas: a) inicialização do conjunto dos possíveis pontos de corte B e do esquema de discretização D; b) consecutivas inserções de pontos de corte no esquema de discretização D a partir das avaliações dos mesmos pela métrica adaptada HCAIM. O detalhamento de cada uma dessas etapas encontra-se descrito a seguir.

A etapa de inicializações é realizada para cada atributo F_i (linhas 3 a 7). Nessa etapa, a primeira inicialização (linha 3) é a do conjunto dos possíveis pontos de corte B. Considerando-se que o atributo a ser discretizado F_i encontra-se ordenado, os pontos de corte inseridos no conjunto B correspondem à média dos valores do atributo F_i para cada par de instâncias vizinhas que encontram-se associadas a classes diferentes e possuem valores distintos para o atributo em questão. Em seguida (linha 4), o esquema de discretização D é inicializado com um único intervalo [-∞,+∞]. Por fim, entre as

linhas 5 e 7, as variáveis $globalHCaim$ (armazena o melhor valor da métrica HCAIM ao longo de todo o processo de discretização), k (controla o número de pontos de corte inseridos no esquema D) e $parar$ (controla a finalização do processo de discretização do atributo F_i) também são inicializadas.

Finalizadas as inicializações anteriormente descritas, enquanto o critério de parada não é alcançado, novos pontos de corte são consecutivamente inseridos no esquema de discretização D (linhas 8 a 27). A inserção de um novo ponto de corte no esquema de discretização é realizada escolhendo-se, a cada iteração, dentre os pontos de corte contidos em B, aquele que é melhor avaliado pela métrica HCAIM (linhas 11 a 19). A variável $localHCaim$, inicializada na linha 10, é responsável por armazenar o melhor valor da métrica HCAIM ao longo desse processo de escolha do melhor ponto de corte contido em B. Toda vez que um novo ponto de corte é inserido no esquema D (linha 22), ele também é removido da lista de possíveis pontos de corte B (linha 23). O critério de parada do método (linha 20) estabelece que novos pontos de corte devem ser inseridos no esquema de discretização D enquanto o número de pontos de corte já inseridos no esquema for menor do que o número de classes distintas da base de dados ou enquanto a inserção de um novo ponto de corte melhorar o melhor valor já obtido para a métrica de avaliação (armazenado em $globalHCaim$). Quando o critério de parada é atingido, o esquema de discretização D do atributo F_i é armazenado em uma lista de esquemas (linha 28) e todo esse processo de discretização é novamente realizado se existirem outros atributos contínuos na base de dados.

A Figura 3.7 apresenta o pseudocódigo do algoritmo usado para o cálculo da métrica HCAIM, que recebe como entrada a base de dados BD, o esquema T e o atributo que está sendo discretizado F_i . Para cada intervalo I existente no esquema de discretização T são construídas listas L_k (linha 8) contendo a frequência das classes (considerando-as até o nível k da hierarquia) associadas às instâncias cujo valor do atributo F_i está contido no intervalo I. A partir dessas listas L_k, o valor da métrica para cada intervalo I é calculado (linhas 11 e 12) utilizando-se: a) o maior valor contido em L_k, ou seja, aquele associado à classe mais frequente (linha 10); b) o número total de instâncias cujo valor do atributo F_i está contido no intervalo I (linha 6) e c) peso atribuído ao nível k de uma estrutura hierárquica com profundidade H (linha 9). O valor final da métrica corresponde à média dos valores de HCAIM calculados para cada intervalo I (linha 16).

A Figura 3.8 apresenta o pseudocódigo para o cálculo dos pesos utilizados na métrica de avalição HCAIM. A função recebe o nível atual e a profundidade da hierarquia de classes associadas ao intervalo. Essa função apenas calcula e retorna o peso para os

Algoritmo `HCAIM` (Base de dados BD)
1: Inicializar $listaEsquemas = \emptyset$;
2: **para** cada atributo contínuo F_i **faça**
3: Inicializar o conjunto de possíveis pontos de corte B do atributo A;
4: Criar o esquema de discretização inicial $D = \{[-\infty, +\infty]\}$;
5: Inicializar $globalHCaim = 0$;
6: Inicializar $k = 1$;
7: Inicializar $parar$ = FALSO;
8: **enquanto** !$parar$ **faça**
9: $parar$ = VERDADEIRO;
10: Inicializar $localHCaim = 0$;
11: **para** cada possível ponto de corte c contido em B **faça**
12: $T = D$;
13: Inserir o ponto de corte c no esquema T;
14: $hcaim$ = métricaHCAIM(BD, T, F_i);
15: **se** ($hcaim > localHCaim$) **então**
16: $localHcaim = hcaim$;
17: $p = c$;
18: **fim se**
19: **fim para**
20: **se** ($localHCaim > globalHCaim$ **ou** $k < |S|$) **então**
21: $globalHCaim = localHCaim$;
22: Inserir o ponto p no esquema D em ordem crescente;
23: Remover o ponto p da lista B;
24: $parar$ = FALSO
25: **fim se**
26: $k = k + 1$;
27: **fim enquanto**
28: Inserir o esquema D em $listaEsquemas$
29: **fim para**
fim.

Figura 3.6: Pseudocódigo do método HCAIM.

parâmetros recebidos de acordo com a Equação 3.2.

Na próxima seção são apresentadas os experimentos computacionais realizados e os resultados obtidos na avaliação do método proposto neste trabalho contra os métodos de discretização não supervisionados *EqualWidth* e *EqualFrequency*.

Função `métricaHCAIM`(Base de dados BD, Esquema T, Atributo F_i)

1: Inicializar $hcaim = 0.0$;
2: Inicializar $numeroIntervalos = 0$;
3: **para** cada intervalo $I \in T$ **faça**
4: S = Conjunto de instâncias da base de dados cujo valor de $F_i \subset I$
5: H = Profundidade da hierarquia de classes associada às instâncias $\in S$;
6: M = número total de instancias contidas em S
7: **para** k de 1 até H **faça**
8: Criar a lista L_k com a frequência das classes das instâncias $\in S$, considerando-as até o nível k da hierarquia;
9: W = **calcularPeso**(k, H);
10: max = maior valor contido em L_k;
11: $caimIntervalo = (max^2 \ / \ M) * W$;
12: $hcaim = hcaim + caimIntervalo$;
13: **fim para**
14: $numeroIntervalos = numeroIntervalos + 1$;
15: **fim para**
16: **Retorne** ($hcaim$ / $numeroIntervalos$);

fim.

Figura 3.7: Pseudocódigo do algoritmo usado no cálculo da métrica HCAIM.

Função `calcularPeso`(*Nível* k, *Profundidade* H)

1: **Retorne** $(H - k + 1) * (2 \ / \ (H * (H + 1)))$;

fim.

Figura 3.8: Pseudocódigo para o calculo de pesos.

Capítulo 4

Experimentos Computacionais

Neste capítulo são apresentados os experimentos computacionais realizados para avaliação do método proposto. Inicialmente, as bases de dados utilizadas nos experimentos e os pré-processamentos realizados nas mesmas são descritos na Seção 4.1. Em seguida, na Seção 4.2 são apresentadas as configurações experimentais. Por fim, os resultados obtidos nos experimentos são mostrados e analisados na Seção 4.3.

4.1 Bases de dados

Todos os experimentos foram conduzidos a partir de nove bases de dados relacionadas com a classificação de funções de genes. Nessas bases, os atributos preditores incluem diversos tipos de dados da área de bioinformática, tais como: estrutura secundária da sequência, fenótipo, homologia, estatísticas da sequência e outros. Essas bases de dados, inicialmente utilizadas em (Clare and King, 2003) e depois adaptadas e utilizadas em (Vens et al., 2008), eram multirrótulo, ou seja, cada instância encontra-se associada a uma ou mais classes da hierarquia.

Como este trabalho lida com o cenário monorrótulo (*single path of labels*), as bases de dados foram transformadas para conter uma única classe por instância. Essa transformação foi realizada selecionando-se, para cada instância, a classe mais frequente na base de dados original.

A partir das bases de dados monorrótulo, um pré-processamento foi realizado para substituição dos valores ausentes de atributos nessas bases. O procedimento descrito

a seguir foi adotado para a substituição dos valores ausentes. Quando identificado um valor ausente para um determinado atributo F_j de uma instância associada à classe C_i, calcula-se a média dos valores conhecidos do atributo F_j de todas as demais instâncias da base associadas à classe C_i e, em seguida, utiliza-se essa média para substituição do valor ausente. Se para a classe C_i nenhuma instância possuir valor conhecido para o atributo F_j, calcula-se a média dos valores conhecidos do atributo F_j de todas as instâncias da base associadas às classes descendentes de C_i na hierarquia e, em seguida, utiliza-se essa média para substituição do valor ausente. Em último caso, se a classe C_i não possuir classes descendentes ou se para as classes descendentes de C_i nenhuma instância possuir valor conhecido para o atributo F_j, então substitui-se o valor ausente pela média global do atributo F_j.

A Tabela 4.1 mostra as principais características das bases de dados após o pré-processamento descrito anteriormente. Essa tabela apresenta, para cada base de dados, o número de instâncias, número de atributos, número de níveis da hierarquia de classes, o número total de classes e o número de classes por nível da hierarquia ($1^{\circ}|2^{\circ}|3^{\circ}|4^{\circ}|5^{\circ}|6^{\circ}$).

Tabela 4.1: Características das bases de dados

Bases	Instâncias	Atributos	Níveis	Classes	Classes por Nível
Church	3755	28	6	190	7|37|72|47|25|2
Eisen	2424	80	6	143	4|26|55|34|22|2
Cellcycle	3757	78	6	190	7|37|73|46|25|2
Gasch2	3779	53	6	191	7|37|73|46|26|2
Gasch1	3764	174	6	191	7|37|73|46|26|2
Derisi	3725	64	6	190	7|37|72|47|25|2
Spo	3703	81	6	191	7|37|73|46|26|2
Seq	3919	479	6	192	7|37|73|47|26|2
Expr	3779	552	6	191	7|37|72|47|26|2

4.2 Configuração Experimental

Os métodos de discretização não supervisionados *EqualFrequency* e *EqualWidth* foram utilizados como referência para comparação com método proposto, o HCAIM. Esses métodos foram escolhidos para as comparações pelo fato de serem comumente adotados

em trabalhos de classificação hierárquica existentes na literatura, uma vez que não há métodos de discretização supervisionados para esse contexto.

Os métodos *EqualFrequency* e *EqualWidth* possuem um parâmetro k, que define o número de intervalos a serem criados no processo de discretização. Para tornar a comparação justa, os experimentos foram executados para diferentes valores de k, a saber, 5, 10, 15 e 20. Esses métodos foram executados a partir das suas implementações disponíveis na ferramenta *WEKA* (Garner et al., 1995).

Para avaliar a qualidade da discretização realizada por cada um dos métodos foi utilizado o classificador hierárquico *Global Model Naive Bayes – GNMB* (Silla Jr et al., 2009a). Esse classificador, que utiliza a abordagem de classificação hierárquica global, foi descrito na Seção 2.1.4 do Capítulo 2.

Para expressar o desempenho preditivo do classificador hierárquico *GMNB* adotou-se a métrica *F-measure* hierárquica (hF) posposta em (Kiritchenko et al., 2005).

(Kiritchenko et al., 2005) definem a métrica F-measure (hF) como $hF = \frac{2 \times hP \times hR}{hP+hR}$, onde hP e hR representam a precisão hierárquica e a revocação hierárquica respectivamente. Onde o hP e hR são definidos como $hP = \frac{\sum_i |P_i \cap T_i|}{\sum_i |P_i|}$, $hR = \frac{\sum_{i=1}^{n} |P_i \cap T_i|}{\sum_{i=1}^{n} |T_i|}$. Onde P_i é o conjunto composto pela classe mais específica prevista para a instância de teste i e todas as suas classes ancestrais e T_i é o conjunto composto pela verdadeira classe mais específica para uma instância de teste i e todas as suas classes ancestrais. A Tabela 4.2 apresenta alguns exemplos da utilização das métricas hP, hR e hF.

Tabela 4.2: Exemplos de aplicação das métricas hierárquicas

Instância	Classe Predita	Classe Real	hP	hR	hF
1	R.1	R.1.2	1/1	1/2	0,666
2	R.1	R.1.2.1	1/1	1/3	0,5
3	R.1	R.1.2.1.1	1/1	1/4	0,4
4	R.2.2	R.2	1/2	1/1	0,666
5	R.2.2.1	R.2	1/3	1/1	0,5
6	R.2.2.1.3	R.2	1/4	1/1	0,4

Além disso, o método k-validação cruzada (k=10) foi utilizado na avaliação do desempenho preditivo do *GMNB*. Para cada base de dados, os mesmos dez pares de bases

(treinamento e teste) foram utilizados na avaliação de todos os métodos que estão sendo comparados. Vale ressaltar também que a discretização dos dados ocorreu somente após o particionamento da base pelo método 10-validação cruzada, ou seja, para cada base, ela foi aplicada considerando-se cada uma das 10 partições de treinamento.

4.3 Resultados

A Tabela 4.3 mostra o *hF* médio obtido pelo classificador *GMNB* para cada base de dados discretizada utilizando-se os diferentes métodos considerados neste experimento e o desvio padrão é apresentado entre parênteses. Nessa tabela, a primeira coluna apresenta os nomes das bases de dados e as demais o desempenho do GMNB após a discretização da base realizada pelo método que dá nome à colunaNo caso dos métodos de discretização *EqualFrequency* (*EF*) e *EqualWidth* (*EW*), o nome da coluna é formado pelo nome do método acrescido, entre parênteses, do valor do parâmetro k utilizado. Os valores em negrito indicam o melhor resultado obtido para cada base de dados.

Os resultados apresentados na Tabela 4.3 mostram que o método de discretização proposto neste trabalho (HCAIM) proporcionou o maior desempenho preditivo ao GMNB para 6 das 9 bases de dados utilizadas nos experimentos. Para as 3 bases em que o HCAIM não foi superior a todos os demais métodos, exceto para base Seq, ele obteve um desempenho próximo aos métodos não supervisionado.

Para cada base de dados, para determinar se a diferença entre o resultado produzido a partir do uso do método HCAIM e cada um dos demais métodos de discretização é estatisticamente significativa, o teste estatístico de Wilcoxon, proposto em (Wilcoxon, 1945) foi utilizado. Nos testes estatísticos utilizou-se um nível de confiança de 95%.

A Tabela 4.4 apresenta os resultados dos testes estatísticos na comparação par-a-par do método HCAIM com os métodos de discretização *EqualFrequency* e *EqualWidth*. A primeira coluna dessa tabela apresenta os nomes métodos (entre parênteses tem-se o valor do parâmetro k) utilizados na comparação com o HCAIM. Da segunda até a décima coluna temos os resultados do teste estatístico para cada uma das bases de dados utilizadas nos experimentos. Nessas colunas, o valor 1 indica que o método HCAIM obteve melhor desempenho com significância estatística, 0 indica que os resultados dos métodos são estatisticamente equivalentes e -1 significa que o método HCAIM obteve desempenho significativamente inferior ao do método utilizado na comparação. Por fim,

da décima até a décima segunda coluna apresenta-se a contabilização do número vitórias (V), empates (E) e derrotas (D) do método HCAIM com relação a cada um dos demais métodos utilizados na avaliação comparativa.

Os testes estatísticos mostram que na comparação do HCAIM com cada um dos outros métodos utilizados nos experimentos, ele sempre apresenta um desempenho estatisticamente superior ou igual ao dos demais métodos para a maioria das bases de dados avaliadas. Por exemplo, quando comparado com o método *EqualFrequency* com $k = 5$ (EF(5)), o HCAIM é superior para 6 bases de dados, equivalente em 2 bases e inferior para a apena 1 base.

A partir da Tabela 4.4 podemos observar também que do total de 72 comparações (8 métodos $\times$ 9 bases), o método HCAIM mostrou-se superior em 43 comparações, equivalente em 21 e inferior em apenas 8. Portanto, os testes estatísticos confirmam a superioridade do HCAIM com relação aos demais métodos de discretização utilizados nos experimentos. Essa superioridade significa que, quando a base de dados é discretizada pelo método HCAIM, na maioria das vezes, o desempenho preditivo do classificador hierárquico GMNB é melhor do que aquele obtido quando a discretização é realizada pelos outros métodos avaliados.

Tabela 4.3: Valores médios de hF obtidos pelo GMNB.

Base	HCAIM	EF(5)	EF(10)	EF(15)	EF(20)	EW(5)	EW(10)	EW(15)	EW(20)
Cellcycle	31.853	20.833	24.559	26.031	26.564	15.407	17.079	18.962	19.096
	(1.69)	(1.39)	(2.69)	(2.21)	(2.13)	(1.75)	(1.37)	(1.63)	(2.03)
Church	18.630	10.070	11.566	12.003	13.144	10.901	11.848	11.756	12.633
	(1.13)	(1.25)	(1.31)	(1.63)	(1.45)	(1.01)	(1.51)	(1.41)	(1.58)
Derisi	12.422	9.361	10.981	11.732	11.521	8.917	9.313	9.692	9.908
	(0.82)	(1.00)	(1.20)	(1.07)	(1.07)	(0.75)	(1.28)	(1.30)	1.14)
Eisen	21.279	22.564	22.524	21.857	21.782	20.740	22.169	22.404	22.488
	(1.43)	(1.33)	(2.10)	(2.23)	(1.70)	(2.35)	(1.42)	(1.27)	(1.97)
Expr	46.409	43.912	45.820	45.667	45.537	26.821	29.615	32.599	34.462
	(1.66)	(1.39)	(1.88)	(2.35)	(2.49)	(1.69)	(1.88)	(1.49)	(1.27)
Gasch1	26.857	18.639	21.751	22.385	22.838	16.800	18.365	19.624	19.361
	(1.16)	(1.97)	(2.33)	(1.51)	(1.75)	(1.47)	(1.61)	(2.27)	(2.06)
Gasch2	25.510	16.475	17.587	19.367	19.693	14.751	16.105	15.773	16.606
	(1.74)	(1.70)	(1.45)	(1.90)	(1.68)	(1.17)	(1.47)	(1.32)	(1.47)
SPO	13.144	13.624	14.309	14.839	14.301	13.768	13.171	13.017	13.622
	(1.73)	(1.41)	(1.25)	(1.37)	(0.78)	(1.43)	(1.19)	(1.63)	(0.85)
Seq	18.098	21.393	19.578	18.827	18.748	24.018	24.913	24.046	24.049
	(1.08)	(0.87)	(1.03)	(1.32)	(1.15)	(1.67)	(1.47)	(1.11)	(1.26)

Tabela 4.4: Resultado do teste estatístico.

	Church	Eisen	Cellcycle	Gasch2	Gasch1	Derisi	SPO	Seq	Expr	V	E	D
EF(5)	1	0	1	1	1	1	0	-1	1	6	2	1
EF(10)	1	0	1	1	1	1	0	-1	0	5	3	1
EF(15)	1	0	1	1	1	0	-1	0	0	4	4	1
EF(20)	1	0	1	1	1	0	0	0	0	4	5	0
EW(5)	1	0	1	1	1	1	0	-1	1	6	2	1
EW(10)	1	0	1	1	1	1	0	-1	1	6	2	1
EW(15)	1	-1	1	1	1	1	0	-1	1	6	1	2
EW(20)	1	0	1	1	1	1	0	-1	1	6	2	1

Capítulo 5

Conclusões

Apesar da importância dos métodos de discretização para o pré-processamento das bases de dados utilizadas em trabalhos envolvendo a tarefa de classificação, até onde se tem conhecimento, para problemas de classificação hierárquica, não existem na literatura propostas de métodos de discretização supervisionados que possam ser utilizados em conjunto com classificadores hierárquicos globais.

Portanto, este trabalho propôs um método de discretização supervisionado para o contexto da classificação hierárquica. Essa proposta corresponde a uma adaptação do método de discretização supervisionado denominado CAIM, o qual foi originalmente proposto para trabalhar no contexto da classificação plana. A adaptação proposta, denominada HCAIM, permite que a hierarquia de classes seja considerada durante o processo de discretização dos atributos contínuos.

Dada a ausência de métodos supervisionados para o contexto hierárquico, diversos trabalhos de classificação hierárquica apresentados na literatura que necessitaram realizar a discretização de dados como uma etapa de pré-processamento ficaram limitados à utilização de técnicas não supervisionadas, tais como *EqualWidth* e *EqualFrequency*.

Desse modo, os métodos de discretização não supervisionados *EqualWidth* e *EqualFrequency* foram utilizados como base de comparação com o método proposto HCAIM. A qualidade da discretização realizada por cada um dos métodos avaliados foi realizada a partir do método de classificação hierárquica *Global Model Naive Bayes – GMNB*.

Os experimentos computacionais realizados com 9 bases de dados de bioinformática mostraram que o método HCAIM, para a maioria das bases, permitiu ao GMNB al-

cançar desempenho preditivo superior àqueles alcançados quando a base de dados foi pré-processada pelos métodos não supervisionados *EqualWidth* e *EqualFrequency*. Esse resultado confirma o potencial de aplicação do método proposto para a realização da discretização de dados que serão utilizados em trabalhos de classificação hierárquica.

Um trabalho futuro relevante corresponde a uma avaliação de como o método HCAIM pode influenciar no desempenho de outros métodos de classificação hierárquica que utilizam a abordagem global. Além disso, é interessante avaliar a adaptação do método para o contexto multirrótulo.

Referências Bibliográficas

Au, W.-H., Chan, K. C. and Wong, A. K.: 2006, A fuzzy approach to partitioning continuous attributes for classification, *Knowledge and Data Engineering, IEEE Transactions on* **18**(5), 715–719.

Barutcuoglu, Z. and DeCoro, C.: 2006, Hierarchical shape classification using bayesian aggregation, *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, IEEE, pp. 44–44.

Bay, S. D.: 2001, Multivariate discretization for set mining, *Knowledge and Information Systems* **3**(4), 491–512.

Boullé, M.: 2006, Modl: A bayes optimal discretization method for continuous attributes, *Machine learning* **65**(1), 131–165.

Campos Merschmann, L. H. and Freitas, A. A.: 2013, An extended local hierarchical classifier for prediction of protein and gene functions, *Data Warehousing and Knowledge Discovery*, Springer, pp. 159–171.

Cesa-Bianchi, N. and Valentini, G.: 2009, Hierarchical cost-sensitive algorithms for genome-wide gene function prediction, *Machine Learning in Systems Biology, Proceedings of the Third international workshop*, pp. 25–34.

Chen, Y.-L., Hu, H.-W. and Tang, K.: 2009, Constructing a decision tree from data with hierarchical class labels, *Expert Systems with Applications* **36**(3), 4838–4847.

Ching, J. Y., Wong, A. K. and Chan, K. C.: 1995, Class-dependent discretization for inductive learning from continuous and mixed-mode data, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **17**(7), 641–651.

Clare, A. and King, R. D.: 2003, Predicting gene function in saccharomyces cerevisiae, *Bioinformatics* **19**(suppl 2), ii42–ii49.

D'Alessio, S., Murray, K. A., Schiaffino, R. and Kershenbaum, A.: 2000, The effect of using hierarchical classifiers in text categorization., *RIAO*, pp. 302–313.

Dougherty, J., Kohavi, R., Sahami, M. et al.: 1995, Supervised and unsupervised discretization of continuous features, *Machine learning: proceedings of the twelfth international conference*, Vol. 12, pp. 194–202.

Duda, R. O. and Hart, P. E.: 1973, Pattern recognition and scene analysis.

Dumais, S. and Chen, H.: 2000, Hierarchical classification of web content, *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 256–263.

Fayyad, U. M. Irani, K. B.: 1993, Multi-interval discretization of continuous-valued attributes for classification learning.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P.: 1996, From data mining to knowledge discovery in databases, *AI magazine* **17**(3), 37.

Ferrandiz, S. and Boullé, M.: 2005, Multivariate discretization by recursive supervised bipartition of graph, *Machine learning and data mining in pattern recognition*, Springer, pp. 253–264.

Flores, J. L., Inza, I. and Larrañaga, P.: 2007, Wrapper discretization by means of estimation of distribution algorithms, *Intelligent Data Analysis* **11**(5), 525–545.

Freitas, A. A. and Carvalho, A. C.: 2007, A tutorial on hierarchical classification with applications in bioinformatics.

Garcia, S., Luengo, J., Sáez, J. A., López, V. and Herrera, F.: 2013, A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning, *Knowledge and Data Engineering, IEEE Transactions on* **25**(4), 734–750.

Garner, S. R. et al.: 1995, Weka: The waikato environment for knowledge analysis, *Proceedings of the New Zealand computer science research students conference*, Citeseer, pp. 57–64.

Ho, K. and Scott, P.: 1997, Zeta: a global method for discretization of cotitinuous variables, *3rd International Conference on Knowledge Discovery and Data Mining (KDD99), NewPort Beach, USA*, pp. 191–194.

Hussain, F., Huan, L., TAN, C. L. and Manoranjan, D.: 1999, Discretization: An enabling technique.

Jin, R., Breitbart, Y. and Muoh, C.: 2009, Data discretization unification, *Knowledge and Information Systems* **19**(1), 1–29.

Kerber, R.: 1992, Chimerge: Discretization of numeric attributes, *Proceedings of the tenth national conference on Artificial intelligence*, Aaai Press, pp. 123–128.

Kiritchenko, S., Matwin, S. and Famili, F.: 2005, Functional annotation of genes using hierarchical text categorization.

Koller, D. and Sahami, M.: 1997, Hierarchically classifying documents using very few words.

Kurgan, L. A. and Cios, K. J.: 2004, Caim discretization algorithm, *Knowledge and Data Engineering, IEEE Transactions on* **16**(2), 145–153.

Labrou, Y. and Finin, T.: 1999, Yahoo! as an ontology: using yahoo! categories to describe documents, *Proceedings of the eighth international conference on Information and knowledge management*, ACM, pp. 180–187.

Liu, H., Hussain, F., Tan, C. L. and Dash, M.: 2002, Discretization: An enabling technique, *Data mining and knowledge discovery* **6**(4), 393–423.

Liu, H. and Setiono, R.: 1997, Feature selection via discretization, *IEEE Transactions on Knowledge and Data Engineering* (4), 642–645.

Nguyen, S. H. and Skowron, A.: 1995, Quantization of real value attributes-rough set and boolean reasoning approach, *Proc. of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, Sept 28-Oct 1*, Citeseer.

Pazzani, M. J.: 1995, An iterative improvement approach for the discretization of numeric attributes in bayesian classifiers., *KDD*, pp. 228–233.

Qiu, X., Gao, W. and Huang, X.: 2009, Hierarchical multi-class text categorization with global margin maximization, *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Association for Computational Linguistics, pp. 165–168.

Quinlan, J. R.: 1993, *C4. 5: programs for machine learning*, Vol. 1, Morgan kaufmann.

Secker, A., Davies, M. N., Freitas, A. A., Clark, E., Timmis, J. and 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.

Silla Jr, C. N. and Freitas, A. A.: 2011, A survey of hierarchical classification across different application domains, *Data Mining and Knowledge Discovery* **22**(1-2), 31–72.

Silla Jr, C. N., Freitas, A. et al.: 2009a, A global-model naive bayes approach to the hierarchical prediction of protein functions, *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, IEEE, pp. 992–997.

Silla Jr, C. N., Freitas, A. et al.: 2009b, Novel top-down approaches for hierarchical classification and their application to automatic music genre classification, *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, IEEE, pp. 3499–3504.

Sun, A. and Lim, E.-P.: 2001, Hierarchical text classification and evaluation, *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, IEEE, pp. 521–528.

Tsai, C.-J., Lee, C.-I. and Yang, W.-P.: 2008, A discretization algorithm based on class-attribute contingency coefficient, *Information Sciences* **178**(3), 714–731.

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

Wilcoxon, F.: 1945, Individual comparisons by ranking methods, *Biometrics bulletin* pp. 80–83.

Wu, F., Zhang, J. and Honavar, V.: 2005, Learning classifiers using hierarchically structured class taxonomies, *Abstraction, Reformulation and Approximation*, Springer, pp. 313–320.

Wu, X.: 1996, A bayesian discretizer for real-valued attributes, *The Computer Journal* **39**(8), 688–691.

Xue, G.-R., Xing, D., Yang, Q. and Yu, Y.: 2008, Deep classification in large-scale text hierarchies, *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 619–626.

Yang, P., Li, J.-S. and Huang, Y.-X.: 2011, Hdd: a hypercube division-based algorithm for discretisation, *International Journal of Systems Science* **42**(4), 557–566.

Yang, Y. and Webb, G. I.: 2009, Discretization for naive-bayes learning: managing discretization bias and variance, *Machine learning* **74**(1), 39–74.

Zighed, D. A., Rabaséda, S. and Rakotomalala, R.: 1998, Fusinter: a method for discretization of continuous attributes, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**(03), 307–326.