Programa Interunidades em Bioinformática
Universidade de São Paulo

Um Filtro Iterativo Utilizando Árvores de Decisão

Oscar Picchi Netto

Orientador: Prof. Dr. José Augusto Baranauskas

Dissertação de Mestrado

Ribeirão Preto
2013

À minha família por estar sempre presente independente da situação. Especialmente, aos meus pais pelo apoio, pela motivação e por todo apoio durante toda a caminhada. À minha namorada que nos momentos difíceis sempre esteve do meu lado para me ajudar.

Agradecimentos

Agradeço aos meus pais e meus familiares pelo carinho durante todo o período de mestrado.

A minha namorada Erica pelo apoio, incentivo e companheirismo.

Aos meus amigos, Rafael, Patricia, Thais, Silvio e todos os outros por toda amizade, companhia e o apoio que fizeram destes anos inesquecíveis.

Ao professor José Augusto Baranauskas do Departamento de Computação e Matemática pela orientação, atenção e incentivo durante a realização deste projeto.

À CAPES e o INCT-ADAPTA pelo apoio financeiro.

“A lógica é apenas o princípio da sabedoria e não o seu fim.” (Spock - Jornada nas Estrelas)

Resumo

Usar algoritmos de Aprendizado de Máquina é um dos modos eficientes de extrair as informações de grandes bases biológicas. Sabendo-se que a quantidade de dados que são coletados cresce a cada dia, o uso de alguma técnica de seleção de atributos eficiente é, em alguns casos, essencial não só para otimizar o tempo do algoritmo de Aprendizado da Máquina a ser aplicado posteriormente como também para reduzir os dados, de forma que possa ser possível testá-los, por exemplo, em uma bancada de laboratório em algumas situações específicas. O objetivo deste estudo é propor uma abordagem utilizando árvores de decisão em um filtro iterativo, visando auxiliar na extração de informação de grande bases biológicas. Pois, com uma base de menor dimensionalidade, um especialista humano pode entender melhor ou ainda utilizar um algoritmo de Aprendizado de Máquina de forma mais eficaz. O filtro proposto pode utilizar qualquer classificador com um seletor de atributos embutido e qualquer métrica pode ser utilizada para determinar se o atributo deve ser escolhido. Foi fixado, neste estudo, o algoritmo utilizado como J48 e a área embaixo da curva ROC (AUC) como métrica. Em experimentos utilizando diversas bases de dados biomédicas, o filtro proposto foi analisado e sua capacidade de compressão e desempenho foram avaliados em cinco diferentes paradigmas de aprendizado de máquina, utilizando dois limiares diferentes para a métrica escolhida. O melhor limiar obteve uma capacidade de compressão de cerca de 50% dos dados em geral e 99.4% em bases de baixa densidade, geralmente grandes bases. Os valores AUC obtidos pelo filtro quando comparados com cinco algoritmos de paradigmas de aprendizado diferentes mostraram um desempenho melhor em quatro das cinco situações avaliadas. O filtro proposto foi depois analisado e comparado com outros seletores de atributos da literatura e o indutor sozinho. Quanto ao tempo gasto pelo filtro em relação aos outros ele se apresentou no mesmo patamar de 3 dos 4 seletores testados. Quando comparado em relação ao AUC o filtro proposto se mostrou robusto nos cinco indutores analisados, não apresentando nenhuma diferença significativa em nenhum dos cenários testados. Em relação aos indutores, o filtro apresentou um desempenho melhor, mesmo que não significante, em 4 dos 5 indutores.

Palavras Chaves: Aprendizado de Máquina, Seleção de Atributos, Alta Dimensionalidade

Abstract

Using Machine Learning algorithms is an efficient way to extract information from large biological databases. But, in some cases, the amount of data is huge that using an efficient featured subset selection is, in some cases, essencial not only to optimize the learning time but also to reduce the amount of data, allowing, for example, a test in a laboratory workbench.

The objective of this study is to propose an approach using decision trees in a iterative filter. The filter helps information extraction from large biological databases, since in a database with few dimensions a human specialist can understand it better or can use Machine Learning algorithms in a more effective way.

The proposed filter can use any classifier with embed featured subset selection and can use any performance metric to determine which attribute must be chosen. In this study, we have fixed the algorithm used within the filter as J48 and AUC was used as metric for performance evaluation. In experiments using biomedical databases, the proposed filter was analyzed and its compression capacity and performance were tested. In five different Machine Learning paradigms, using two different thresholds for the chosen metric. The best threshold was capable of reducing around 50% of the data using all databases and 99.4% on the small density bases, usually high dimensional databases. AUC values for the filter when compared with the five algorithm got a better performance in four of five tested situations. The proposed filter then was tested against others featured subset selectors from the literature, and against the inducer alone. Analyzing time the proposed filter is in the same level as 3 of 4 of the tested selectors. When tested for AUC the proposed selector shows itself robust in the five inducers tested, not showing any significant difference in all tested scenarios. Against the inducers alone our filter showed a better performance, even not significant, in 4 of the 5 inducers.

Keywords: Machine Learning, Attribute Selection, High Dimensions

Conteúdo

Resumo i
Abstract ii

Conteúdo iii
Sumário iii

Lista de Figuras iv
Lista de Figuras iv

Lista de Tabelas v
Lista de Tabelas v

1 Introdução 1
1.1 Contextualização 1
1.2 Aprendizado de Máquina 2
1.3 Motivação. 3
1.4 Objetivo Geral 3
1.5 Objetivos Especificos 3
1.6 Organização 4

2 Seleção de Atributos 5
2.1 Seleção de Atributos como Busca Heurística 6
2.2 Abordagens para Seleção de Atributos 7
2.2.1 Embutida 7
2.2.2 Filtro 7
2.2.3 Wrapper 8
2.2.4 Hibrida 9
2.3 Seletores 9
2.3.1 CFSSubsetEval 9
2.3.2 ReliefF 11
2.3.3 *Gain Ratio* 11
2.4 Considerações Finais 11

3 **Proposta de Trabalho** 12
3.1 Trabalhos Relacionados . . . 12
3.1.1 *Bootstrap* . . . 14
3.2 Algoritmo Proposto . . . 15
3.2.1 Análise dos Resultados . . . 18
3.3 Considerações Finais . . . 18

4 **Experimentos, Resultados e Discussão** 20
4.1 Bases de Dados . . . 20
4.2 Configurações Experimentais . . . 20
4.3 Resultados e Discussão . . . 22
4.4 Considerações Finais . . . 25

5 **Conclusão** 26
5.1 Principais Resultados . . . 27
5.2 Publicações . . . 28
5.3 Trabalhos Futuros . . . 28

A **Indutores** 29
A.1 PART . . . 29
A.2 KNN . . . 31
A.3 Naive Bayes . . . 33
A.4 SMO . . . 35
A.5 Árvores de Decisão . . . 37
A.5.1 Construindo uma Árvore de Decisão . . . 37
A.5.2 Escolha do Melhor Atributo para Particionar . . . 39
A.5.3 Poda . . . 39
A.5.4 Classificando Novos Exemplos . . . 40

Referências Bibliográficas 29

B **Bases de Dados Utilizadas nos Experimentos** 41

C **Experimento I** 44

D **Experimento II** 53

E **Experimento III** 62

Lista de Figuras

2.1 Um exemplo de espaço de estados de quatro atributos (Adaptado de (**?**) 7
2.2 Abordagem filtro 8
2.3 Abordagem *wrapper* 10
2.4 Abordagem híbrida 10

3.1 Exemplo de Bootstrap 15

A.1 Exemplo de PART 31
A.2 Exemplo de KNN com 1 vizinho 32
A.3 Exemplo de KNN com 3 vizinho 33
A.4 Exemplo bidimensional 36
A.5 Classificador de margem qualquer 36
A.6 Classificador de margem máxima 36
A.7 Uma árvore de decisão simples para o diagnóstico de uma doença 38
A.8 Uma árvore grande é induzida de forma a super-ajustar os exemplos e então ela é podada até obter uma árvore menor (mais simples) 39

C.1 Compressão para cada configuração 45

D.1 Compressão para cada configuração 54

Lista de Tabelas

3.1 Exemplo de funcionamento do algoritmo sem considerar o número de iterações 17
3.2 Exemplo de funcionamento do algoritmo com o critério de parada $l \geq L$ 18

4.1 Benjamini-Hochberg *post-hoc* Test - Tempos dos Seletores para todas as bases / baixa densidade / altas densidades respectivamente 22
4.2 Benjamini-Hochberg *post-hoc* Test - Capacidade de Compressão dos Seletores para todas as bases / baixa densidade / altas densidades respectivamente 23
4.3 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo J48 para todas as bases / baixa densidade / altas densidades respectivamente 23
4.4 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo IBk3 para todas as bases, as de baixa densidade e as de altas densidades respectivamente 24
4.5 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo PART para todas as bases, as de baixa densidade e as de altas densidades respectivamente 24
4.6 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo *Naive Bayes* para todas as bases, as de baixa densidade e as de altas densidades respectivamente 24
4.7 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo SMO para todas as bases, as de baixa densidade e as de altas densidades respectivamente 24

A.1 Conjunto de teste: Exemplo 34

B.1 Resumo das bases utilizadas nos experimentos. ATTR representa o número de atributos em cada base; MISS representa a porcentagem de atributos com valores ausentes, sem considerar o atributo classe. As bases de dados estão em ordem crescente de densidade. 42

C.1 Capacidade de Compressão Média do Algoritmo 1 45
C.2 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o J48 47
C.3 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o IBk3 48
C.4 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o NB 49

C.5 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o PART 50
C.6 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o SMO 51
C.7 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus indutores 52

D.1 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o J48 55
D.2 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o IBk3 56
D.3 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o NaiveBayes 58
D.4 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o PART 59
D.5 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o SMO 60
D.6 Capacidade de Compressão Média do Algoritmo 1 com os 3 critérios de parada . . . 61
D.7 Benjamini-Hochberg *post-hoc* Test - Algoritmo 1 + indutor versus indutores 61

E.1 Valor do Tempo Computacional gasto por cada Seletor escala log_10 63
E.2 Porcentagem de Atributos Selecionados 64
E.3 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o J48 66
E.4 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o IBk3 67
E.5 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o NaiveBayes 68
E.6 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o PART 69
E.7 Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o SMO 70

Capítulo

1

Introdução

1.1 Contextualização

Mineração de Dados é um campo interdisciplinar que utiliza técnicas de Aprendizado de Máquina (AM), estatística, reconhecimento de padrões, banco de dados e visualização. Essa área foca em resolver problemas de extração de conhecimento em grandes bases de dados. A tecnologia atual permite a coleta e o armazenamento de um volume de dados cada vez maior. Para que nada importante seja perdido durante o processo de coleta, normalmente decide-se incluir todos os atributos, deixando para o algoritmo de AM selecionar aqueles mais relevantes. Depois dessa seleção o conhecimento extraído pode ser combinado de alguma forma (**?**). Por exemplo, um profissional utilizando algoritmos de AM para tentar separar diferentes doenças que apresentam sintomas parecidos utilizando como base vários registros médicos. Normalmente esses registros médicos apresentam diversas informações (atributos) que não necessariamente serão importantes para descrever as doenças estudadas, como por exemplo nome, idade, sexo, etc. Cabe ao algoritmo de AM separar o que é relevante. Outro ponto importante é que, dependendo da doença, mesmo com sintomas parecidos as informações necessárias para diagnosticá-las não necessariamente são as mesmas. O mesmo se aplica para a diferenciação entre genes relevantes ou não para um determinado fenômeno de interesse. Todavia, em muitas situações, não é simples escolher ou realizar o processo de seleção de atributos, pois não se sabe exatamente quais são mais relevantes para o aprendizado de um determinado con-

ceito. Por exemplo, o aumento exponencial da quantidade de dados biológicos disponível cria dois problemas: é necessário armazenar e manusear essa informação de forma eficiente e é necessário saber utilizar esses dados extraindo do melhor modo possível as informações úteis contidas (**?**). Um aspecto importante a ser considerado quando utiliza-se AM em mineração de dados de qualquer natureza é que a redução da dimensionalidade numa base de dados ocorre quando se reduz o número de atributos (*i.e.* deletar colunas na literatura de base de dados ou deletar atributos na literatura de Aprendizado de Máquina). O subconjunto resultante dessa redução mantém o mesmo número de exemplos porém apenas um subconjunto das propriedades com capacidade de predição se compararmos com a base de dados original. Isso é conhecido como o problema de Seleção de um Subconjunto de Atributos ou *Feature Subset Selection* (FSS), onde um dos principais problemas consiste em selecionar propriedades relevantes e/ou eliminar as irrelevantes.

1.2 Aprendizado de Máquina

Aprendizado de Máquina é uma área da Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas por meio de solução bem sucedida de problemas anteriores. Dentro do aprendizado de máquina há o aprendizado indutivo, que pode ser entendido em três partes: aprendizado não supervisionado (**?**), aprendizado semi-supervisionado (**?**) e aprendizado supervisionado (**?**). Sobre este último, que será utilizado neste estudo, é dada uma breve descrição a seguir.

Aprendizado supervisionado é o processo de criar automaticamente um modelo de classificação a partir de uma base de exemplos (chamada base de treinamento) que possuem um atributo (coluna) chamado de atributo classe. Existem dois aspectos a serem considerados neste processo: qual propriedade se deve usar para descrever o conceito e como combinar essas propriedades. Uma vez que o modelo seja criado, ele pode ser usado para, automaticamente, predizer a classe de um conjunto não classificado de exemplos.

Em outras palavras, no aprendizado supervisionado, um indutor (algoritmo de aprendizado) recebe uma base de n exemplos de treinamento. Cada exemplo $\boldsymbol{x}$ é um elemento do conjunto $F_1 \times F_2 \times \ldots \times F_m$, onde F_j é o domínio da j-esima propriedade, $\boldsymbol{x}$ é um vetor e será representado dessa forma (não vetorial). Exemplos de treinamento são tuplas (x, y) onde y é o rótulo, saída ou classe do exemplo $\boldsymbol{x}$. Os valores do y são retirados, geralmente, de um conjunto discreto de classes $\{C_1, C_2, \ldots, C_k\}$ no caso de classificação ou um conjunto de valores reais no caso de regressão.

Neste trabalho apenas classificação será tratada. Dada uma base de treinamento, o algoritmo de aprendizado produz como resultado um classificador que, dado um novo exemplo é capaz de predizer, de modo preciso, a classe y.

1.3 Motivação

Usar algoritmos de AM é um dos modos eficientes de extrair as informações de grandes bases biológicas.Sabendo-se que a quantidade de dados que são coletados cresce a cada dia, o uso de alguma técnica de FSS eficiente é, em alguns casos, essencial não só para otimizar o tempo do algoritmo como também para reduzir os dados, de forma que possa ser possível testá-los, por exemplo, em uma bancada de laboratório em alguns casos específicos. Por essa razão FSS passou de um exemplo ilustrativo para um pré-requisito para a construção de modelos (**?**). Especialmente, em análise de dados de *microarray* ou mesmo em mineração de textos, onde há uma grande quantidade de dados, um algoritmo de FSS pode reduzir-lo consideravelmente.

Existem diversas razões para utilizar FSS. Uma delas é o fato de que, em geral, ela melhora a precisão, pois muitos algoritmos de AM perdem desempenho quando existem muitos atributos. Outra razão é que FSS, dependendo do algoritmo de AM utilizado, pode aumentar a compreensibilidade, que é a habilidade para humanos entenderem os dados e as regras de classificação induzidas pelo algoritmo de AM. Finalmente, FSS pode reduzir o custo de medição, pois quanto maior o número de características (atributos) maior pode ser o custo para medição. Neste estudo é apresentado, uma abordagem para FSS utilizando árvores de decisão dentro de um filtro (**?**)

1.4 Objetivo Geral

O objetivo deste estudo é propor uma abordagem utilizando árvores de decisão em um filtro iterativo para fazer seleção de atributos. Desse modo, foi criado um filtro que utilize um classificador com seleção embutida de atributos para fazer esta seleção.

1.5 Objetivos Especificos

- **Aplicabilidade em grandes bases de dados**. Visando auxiliar na extração de informação nessas grandes bases, especialmente em bases biomédicas. Com uma base de menor dimensionalidade, um especialista humano pode entender melhor ou ainda utilizar um algoritmo de aprendizado de máquina de forma mais eficaz.

- **Rapidez e Eficiência**. Criar um filtro que tenha boa velocidade em relação aos filtros da literatura assim como com a mesma eficiência. Um filtro rápido e eficiente se torna uma opção

válida para ser utilizada por especialistas.

1.6 Organização

O restante deste trabalho está organizado da seguinte maneira:

- No Capítulo 2 serão apresentados os conceitos de seleção de atributos, bem como técnicas utilizadas para fazer a seleção e uma breve descrição dos seletores utilizados neste trabalho;
- No Capítulo 3 são apresentados os trabalhos relacionados, assim como a metodologia proposta juntamente com o Algoritmo 1 e como será feita a análise dos resultados;
- No Capítulo 4 as bases de dados utilizadas são apresentadas, a configuração experimental e os resultados são apresentados e discutidos;
- No Capítulo 5 é apresentada a conclusão do trabalho, principais resultados, publicações e trabalhos futuros.
- No Apêndice A os indutores que foram usados no trabalho são apresentados, assim como uma descrição dos mesmos.
- No Apêndice B uma breve descrição assim como informações sobre as bases de dados utilizadas são apresentadas.
- No Apêndice C o primeiro experimento preliminar é apresentado utilizando uma versão inicial do Algoritmo 1.
- No Apêndice D o segundo experimento preliminar é apresentado, assim como seus resultados. A versão final apresentada neste trabalho foi utilizada neste experimento.
- No Apêndice E o terceiro experimento que utiliza a versão final do Algoritmo 1 é comparada com outros seletores da literatura.

Capítulo

2

Seleção de Atributos

Neste capítulo será explicado um pouco mais detalhadamente o tema seleção de atributos, explicando as principais técnicas de seleção de atributos. Além disso, uma breve explicação sobre os seletores utilizados neste trabalho é fornecida. O foco deste capítulo são os filtros, pois o Algoritmo 1 (Seção 3.2) proposto neste trabalho se encaixa nesta categoria.

Selecionar os atributos mais relevantes em uma base de dados é um dos problemas principais em AM. Embora a maioria dos algoritmos de aprendizado tente selecionar atributos, ou então atribuir graus de importância a eles, análises teóricas e estudos experimentais indicam que muitos algoritmos apresentam um comportamento ruim com a presença de um grande número de atributos irrelevantes (**?**). Mesmo os indutores de árvores de decisão, que explicitamente selecionam alguns atributos em detrimento de outros, exibem este comportamento para alguns conceitos. Outros algoritmos, tais como *naïve* Bayes (**?**), são robustos com relação à presença de atributos irrelevantes, mas podem ser extremamente sensíveis em domínios contendo atributos fortemente correlacionados, mesmo se os atributos são relevantes. Isto pode ser explicado pelo fato de que esses algoritmos assumem independência entre os atributos. Assim, isso sugere a necessidade de métodos adicionais para selecionar um subconjunto útil de atributos quando muitos deles encontram-se disponíveis (**?**).

2.1 Seleção de Atributos como Busca Heurística

O processo de FSS pode ser descrito como uma busca pelo espaço de estados, onde cada nó (estado) representa um subconjunto de atributos; o valor de um nó é uma estimativa da precisão na classificação e os operadores são, geralmente, adicionar ou remover atributos. A seguir, a seleção de atributos é descrita como uma busca heurística, na qual cada estado no espaço de busca é composto por um subconjunto de atributos. Qualquer método de seleção de atributos pode ser caracterizado por sua instanciação em relação a quatro questões básicas, as quais determinam a natureza do processo de busca heurística. São elas:

Ponto de partida que é o primeiro ponto a ser tratado, onde começar a sua busca pode influenciar como a busca será feita. Na Figura 2.1 é ilustrado o espaço de busca para quatro atributos, representados por uma sequência de quatro círculos (**?**). Pode-se observar que existe uma ordem parcial entre os estados, pois cada um deles possui um atributo a mais (círculos escuros) que o estado anterior, sendo o estado inicial, mais à esquerda, o conjunto vazio de atributos, representado pelos quatro círculos em branco. Essa abordagem é geralmente conhecida como seleção *forward*. Já a abordagem que inicia o ponto de partida com o subconjunto contendo todos os atributos e sucessivamente removendo-os é denominada de eliminação *backward*. Também podem ser empregadas variações de ambas abordagens, selecionando-se um estado inicial em algum ponto do espaço de busca e movendo-se a partir desse ponto conhecido como seleção *outward*.;

Organização da busca A cada ponto na busca, modificações locais no conjunto de atributos são consideradas; uma dessas é selecionada e uma nova iteração é realizada. Claramente, uma busca exaustiva em todo o espaço de estados é impraticável, já que para um número m de atributos existem 2^m possíveis estados. Uma abordagem mais prática é a utilização de um método guloso para a travessia do espaço de busca: em cada ponto da busca, consideram-se alterações locais sobre o conjunto corrente de atributos, seleciona-se um novo atributo e realiza-se uma nova iteração;

Estratégia de busca é a terceira questão considera a estratégia utilizada na avaliação dos subconjuntos alternativos de atributos. Uma métrica normalmente empregada envolve a habilidade de um atributo discriminar as classes de uma base de treinamento;

Critério de parada é o último ponto a ser tratado quando parar a busca é o ponto mais importante da busca portanto existem diversos critérios entre eles os mais utlizados são: parar

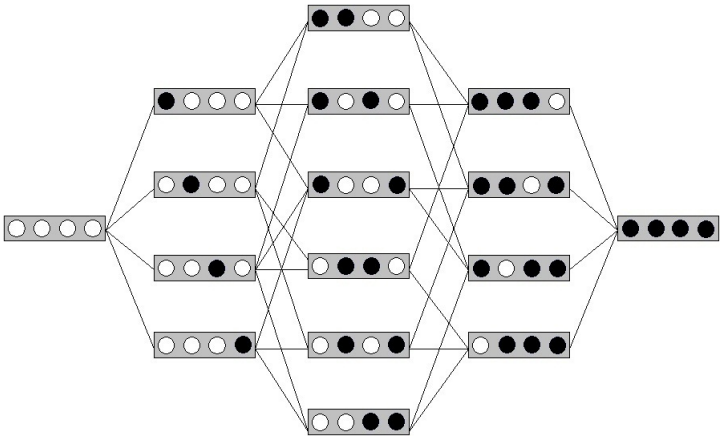

Figura 2.1: *Um exemplo de espaço de estados de quatro atributos (Adaptado de (?)*

de remover ou adicionar atributos quando nenhuma das alternativas melhora o desempenho do classificador; continuar gerando subconjuntos de atributos até que um extremo do espaço de busca seja alcançado e escolher o melhor desses subconjuntos; ordenar os atributos segundo algum critério e utilizar um parâmetro para determinar o ponto de parada, por exemplo, o número de atributos desejado no subconjunto.

2.2 Abordagens para Seleção de Atributos

As abordagens para a seleção de atributos podem ser agrupadas em: *embutida*, *filtro*, *wrapper* e abordagem híbrida descritas a seguir (**?**).

2.2.1 Embutida

Alguns indutores são capazes de realizar sua própria seleção de atributos de forma dinâmica, enquanto procuram por uma hipótese. De fato, FSS é uma parte integral desses indutores. Essas abordagens são especificas de um dado algoritmo de aprendizado. A abordagem embutida tem como vantagem o fato que incluem uma interação com o modelo de classificação e não possuem o custo computacional de uma abordagem *wrapper*. Por exemplo, árvores de decisão realizam seleção embutida.

2.2.2 Filtro

Essa abordagem de seleção de atributos introduz um processo separado, o qual ocorre antes da aplicação do algoritmo de indução propriamente dito, como é mostrado na Figura 2.2. A relevância de cada atributo é determinada apenas olhando as propriedades intrínsecas dos dados. Geralmente filtros calculam e atribuem um valor de relevância para cada atributo e

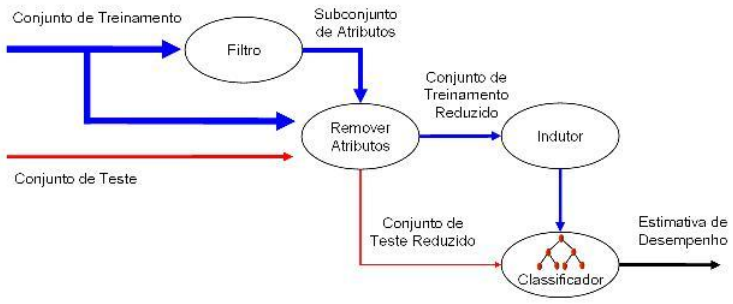

Figura 2.2: *Abordagem filtro*

utilizam um limiar para selecionar apenas os atributos de maior valor. As Figuras 2.2 e 2.3 foram adaptadas de (**?**).

Por não necessitar de um indutor para determinar os valores de relevância, a abordagem de filtro é computacionalmente simples e rápida mesmo em dados de alta dimensionalidade. Portanto, essa abordagem pode fazer a seleção de atributos apenas uma vez e apresentar esse subconjunto em diferentes indutores para avaliação.

Segundo (**?**) e (**?**), um dos esquemas mais simples de filtragem é a avaliação de cada atributo individualmente, baseada na sua correlação com a classe, escolhendo os atributos que fornecem o melhor valor. Esse método é comumente empregado em tarefas de categorização de textos, geralmente combinando filtro com algoritmos *naïve* Bayes (**?**) ou K-NN, os quais têm mostrado bons resultados empíricos.

A principal desvantagem dos filtros é justamente que a maioria dos filtros fazem a avaliação de cada atributo individualmente e, por isso, não vêem a correlação entre os atributos. Para tentar superar esse problema diversas abordagens de filtros multivariáveis, que avaliam vários atributos de uma só vez (**?**; **?**).

2.2.3 Wrapper

Diferentemente da abordagem de filtro, a abordagem *wrapper* tem embutida em sua busca um indutor de aprendizado supervisionado. O *wrapper* gera um subconjunto a partir da base original e apresenta-o ao indutor. O subconjunto é então avaliado de acordo com a precisão obtida pelo indutor, por isso o *wrapper* gera e avalia diversos subconjuntos. Porém,

quanto maior, em número de atributos, a base de dados original maior o espaço de busca e portanto maior o número de subconjuntos gerados. O *wrapper* repete esse processo de geração e avaliação até que um determinado critério de parada seja atingido.

A ideia geral por trás da abordagem *wrapper* é mostrada na Figura 2.3. O algoritmo de FSS existe como um *wrapper* ao redor do indutor e é responsável por conduzir a busca por um bom subconjunto de atributos. A qualidade de um subconjunto candidato é avaliada utilizando o próprio indutor como uma caixa-preta. O objetivo da busca é encontrar o subconjunto (nó) com a melhor qualidade, utilizando uma função heurística para guiá-la.

Um argumento a favor da abordagem *wrapper* é que o mesmo algoritmo de indução que vai utilizar o subconjunto de atributos selecionado deve prover uma estimativa melhor de precisão que um outro algoritmo, o qual pode possuir um viés de indução totalmente diferente (**?**).

2.2.4 Hibrida

Todas as abordagens já descritas possuem pontos positivos e negativos. Com o intuito de tentar achar um melhor modo de fazer a seleção de atributos uma abordagem híbrida pode ser utilizada. Essa abordagem híbrida entre filtro e *wrapper* visa juntar os pontos fortes e superar os pontos fracos dessas abordagens. Primeiramente a base de dados original é apresentada a um filtro, este por usar um *ranker* (uma tabela de posições), pode selecionar um subconjunto com uma métrica de corte para filtrar. O subconjunto resultante deste filtro é então apresentado a um *wrapper* que agora utilizando menos atributos faz uma melhor seleção com um custo computacional não tão grande. Na Figura 2.4 é mostrada uma visão geral dessa abordagem.

2.3 Seletores

Durante os experimentos realizados neste trabalho, alguns seletores de atributos foram utilizados. Este experimento testa o desempenho do filtro proposto contra outros filtros da literatura. Esses filtros estão presentes na WEKA (**?**) a seguir é feita uma breve descrição dos mesmos.

2.3.1 CFSSubsetEval

O CFS, significa (*Correlation-based Feature Selection*) ou Seleção de Atributos baseada em correlação. Esse seletor utiliza essa correlação em subconjuntos, avaliando a capacidade de predição de cada atributo no subconjunto juntamente com o grau de redundância entre os atributos. Este seletor considera um subconjunto como bom se os atributos contidos nele são altamente correlacionados com a classe e contém atributos não correlacionados entre si (**?**).

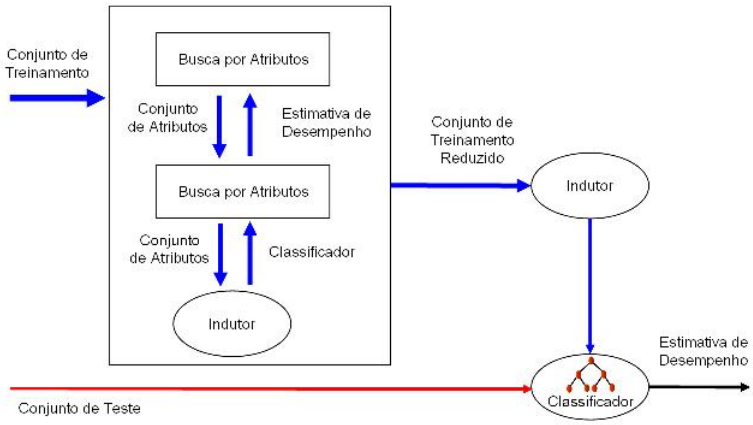

Figura 2.3: *Abordagem* wrapper

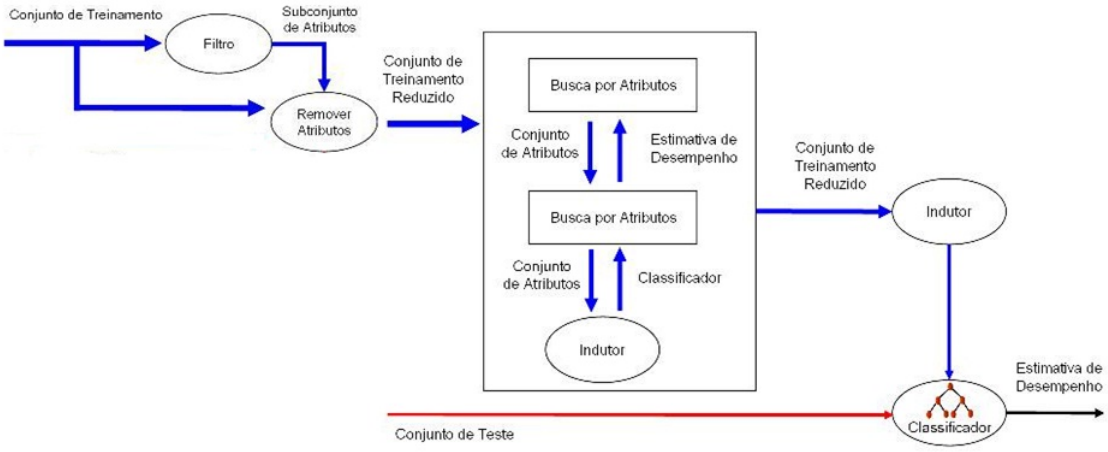

Figura 2.4: *Abordagem híbrida*

2.3.2 ReliefF

A ideia básica do ReliefF é escolher aleatóriamente um subconjunto de exemplos, calcular seus vizinhos próximos e ajustar um vetor. O ReliefF tenta achar uma boa estimativa utilizando a probabilidade descrita na equação:

$$\mathrm{w}_a = \mathrm{P(valor\ diferente\ de\ a|\ Classe\ diferente)\text{-}P(valor\ diferente\ de\ a|\ classe\ igual)}$$

$$to \tag{2.2}$$

que atribui o peso para cada atributo a (**?**).

2.3.3 *Gain Ratio*

Gain Ratio é uma métrica, mais que um seletor. O seletor *Gain Ratio* utiliza a métrica de mesmo nome para *rankear* todos os atributos de uma base de dados. A métrica é calculada utilizando o ganho de informação (*Gain*) de um atributo A contra o número de saídas que um teste com aquele atributo pode resultar (*SplitInfo*) como mostrado na equação:

$$GainRatio(A) = Gain(A)/SplitInfo(A)$$

Por utilizar essa equação o *Gain Ratio* tende a selecionar atributos com maior possibilidades de valores, mesmo que esses não sejam os mais relevantes (**?**).

2.4 Considerações Finais

Neste capítulo foram descritas as principais técnicas de seleção de atributos. Os seletores que foram utilizados no Experimento III (Apêndice E) também foram apresentados de forma sucinta. No próximo capítulo serão apresentados os trabalhos relacionados a este projeto, o filtro proposto e como foram feitas as análises dos resultados.

Capítulo

3

Proposta de Trabalho

No capítulo anterior foram mostradas diversas abordagens para seleção de atributos. Neste capítulo são apresentados trabalhos envolvendo seleção de atributos relacionados ao aqui apresentado. A seguir o algoritmo proposto neste estudo (um filtro descrito no Algoritmo 1) é detalhado com um exemplo simples de seu funcionamento bem como o teste estatístico utilizado na análise de resultados.

3.1 Trabalhos Relacionados

(**?**) propõem uma nova abordagem para seleção de biomarcadores que utiliza um algoritmo de evolução paralela que utiliza a abordagem *wrapper* para seleção de atributos que transforma uma base de dados com milhares de genes em um subconjunto pequeno contendo apenas atributos relevantes. O trabalho compara os resultados com o outro trabalho (**?**) e obtiveram resultados melhores em relação à precisão e ao número de genes selecionados. Este trabalho mostra que utilizar seleção de atributos em bases de alta dimensionalidade é uma boa abordagem e também utiliza *bootstrap* para criar as bases de treinamento e teste. Com base nisso, a utilização do *bootstrap* foi implementada no Algoritmo 1.

(**?**) apontam o fato que seleção de atributo em bases de bioinformática está se tornando um passo essencial. Existem dois problemas em bioinformática: a alta dimensionalidade e poucos

exemplos. Isso indica que utilizar seleção de atributos é uma boa abordagem. É apontado também que nos últimos anos diversas abordagens de seleção de atributos estão sendo estudadas para criar filtros multivariáveis. Filtros multivariáveis é uma área de estudo promissora na comunidade de bioinformática. Este trabalho serviu como fonte motivadora para a criação de um filtro multivariável na pesquisa aqui desenvolvida.

(**?**) apresenta um novo método para seleção de atributos denominado FROC (*Feature selection based on ROC-curves*) que realiza a seleção de atributos em bases de *microarray*. FROC permite que atributos não redundantes e não isolados sejam selecionados de forma robusta utilizando a propriedade não paramétrica da curva ROC. O FROC foi testado contra dois métodos e obteve resultados melhores em 39 de 40 casos. Este trabalho indica que usar um limiar no algoritmo aqui proposto usando AUC tem fundamento. A utilização do AUC como métrica para selecionar os atributos foi implementada no Algoritmo 1.

No trabalho (**?**) uma abordagem de seleção de atributos híbrida é apresentada utilizando uma abordagem de filtro para reduzir o custo computacional do *wrapper* e acima de tudo melhorar o desempenho da classificação. A diferença da abordagem proposta e a de filtro convencional são dois pontos: a primeira é utilizar a capacidade de discriminação visto que um filtro convencional apenas olha a capacidade individual do atributo, enquanto a abordagem proposta pelos autores olha a capacidade de separação de classes do atributo contra os atributos já selecionados; a segunda diferença é como o filtro seleciona: a abordagem convencional utiliza um limiar para selecionar os atributos enquanto que a abordagem proposta seleciona candidatos aleatoriamente pela capacidade de discriminação. A metodologia proposta utiliza um filtro com limiar ao invés de uma abordagem híbrida por razões de desempenho.

Em (**?**) uma abordagem de conjunto de filtros e um conjunto de classificadores foram testados visando retirar os atributos irrelevantes e selecionar os relevantes. A ideia é testar diversos classificadores em um conjunto de vários filtros, ou seja, para um classificador todos os filtros foram utilizados e o resultado desse conjunto é combinado e um único classificador é gerado a partir da combinação de todos os outros classificadores gerados pelos filtros. Foram utilizados diversos filtros já conhecidos para selecionar os atributos e três classificadores, também utilizados nesse trabalho, o C4.5 (similar ao J48), IBl (similar ao IBk3) e NaiveBayes (Apêndice A). As bases de dados utilizadas foram todas de baixa densidade e de microarrays de cancêr. Os resultados mostraram que individualmente os filtros não obtem resultados bons, enquanto que em conjunto eles atingem o melhor resultado em 4 de 10 bases testadas, em média.

Em (?) apresentam uma abordagem para selecionar atributos. Porém essa abordagem tenta não só excluir os atributos redundantes e irrelevantes mas também selecionar os atributos relevantes e que também considera interações entre os atributos em bases de baixa densidade. O algoritmo FRFS proposto no artigo foi testado contra diversos outros seletores de atributos e sua eficácia e eficiência foram testados em quatro classificadores bem conhecidos o *Naive Bayes*, C4.5, PART e IBl. Os resultados em bases de dados sintéticas mostrou que o FRFS pode identificar atributos irrelevantes ou redundantes enquanto seleciona os que apresentam interações com outros. Em bases reais em relação a outros seletores o FRFS reduziu mais eficientemente a base de dados mas também melhorou o desempenho dos quatro classificadores.

No trabalho de (?) é discutida uma versão inicial do Algoritmo 1 proposto nesse estudo. Nele, dez árvores de decisão são criadas iterativamente, a cada iteração os atributos que aparecem na árvore são retirados da base de treinamento. Com isso, foi possível perceber que algumas árvores intermediárias possuíam resultados melhores que o das primeiras. A melhor árvore encontrada também obteve um resultado melhor que algumas abordagens usando apenas 1-2 atributos (genes) da literatura, por exemplo, (?; ?; ?). A abordagem proposta do filtro descrito no Algoritmo 1 é baseada neste trabalho inicial e é descrita na seção seguinte.

3.1.1 *Bootstrap*

Existem diversos métodos de amostragem utilizados para particionar uma base de dados; dentre eles existe um que gera uma amostragem aleatória a partir da base de dados original. De especial interesse neste estudo está o *bootstrap* que consiste em construir uma base de treinamento utilizando uma amostragem com reposição de n casos a partir de uma base de exemplos de tamanho n (?). Uma amostragem com reposição significa que, mesmo que um exemplo seja escolhido da base original, este mesmo exemplo pode ser novamente escolhido, podendo aparecer portanto uma (ou mais vezes) na base de treinamento ou podendo não ser escohlido. A base de treinamento final de tamanho n pode possuir vários exemplos iguais e seu tamanho pode variar de acordo com uma porcentagem da base original. A base de treinamento é conhecida também como *Bag*. O restante dos exemplos, que não estão presentes na base de treinamento, são então denominados como base de teste, também conhecido como *out-of-bag*. Uma amostra *bootstrap* tende a convergir para que 63% dos dados fiquem no conjunto de treinamento e os 37% restantes constituam o conjunto de teste (?)

Por exemplo, considere uma base de dados com dez exemplos representados pelos números

de 1 a 10. Na Figura 3.1 é mostrado como ficariam as bases de treinamento e teste depois da execução de um *bootstrap*, sendo que os exemplos 4, 8 e 9 não foram escolhidos nenhuma vez, ficando assim na base de teste, enquanto os outros ficam na base de treinamento. Os exemplos 1, 2 e 6 foram selecionados duas vezes e aparecem, portanto, duas vezes na base de treinamento.

3.2 Algoritmo Proposto

Em geral, algoritmos de filtros avaliam cada atributo individualmente para algum grau de relevância relacionado com a classe. Algumas vezes dois ou mais atributos podem ser considerados, porém isso faz com que o custo computacional eleve-se (**?**). A abordagem, aqui proposta, difere dessas no sentido que utilizar uma árvore de decisão pode ser capaz de capturar relações entre vários atributos. Além disso, induzir uma árvore de decisão é rápido, o que permite utilizar esta abordagem em dados de alta dimensionalidade, por exemplo dados de expressão gênica ou grandes bases de dados médicas.

De forma resumida, a abordagem iterativamente constrói árvores de decisão (Seção A.5), que seleciona os atributos que aparecem naquela árvore (baseado em um limiar obtido pela primeira árvore induzida) e remove esses atributos da base de treinamento. Esses passos são repetidos até que todos os atributos sejam removidos da base de treinamento ou que uma folha seja induzida (o que indica que nenhum atributo restante foi capaz de separar as classes da base). No final o filtro retorna os atributos selecionados.

No Algoritmo 1 é mostrado em código de alto nível a abordagem de seleção de atributos aqui proposta, no qual N representa o número de exemplos na base de treinamento, x_i e y_i, $i = 1, \ldots, N$ representam o vetor contendo os valores dos atributos e a classe rotulada do exemplo i, respectivamente.

O algoritmo proposto utiliza como métrica para seleção de atributos a área embaixo da curva ROC (*Area Under ROC curve* ou AUC). ROC é a sigla para *receiver operating characteristic* e a curva ROC é um gráfico de sensibilidade (ou taxa de verdadeiros positivos) *versus* taxa de

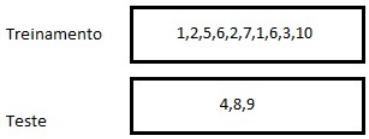

Figura 3.1: *Exemplo de Bootstrap*

Algoritmo 1 An Iterative Decision Tree Threshold Filter — IDTTF

Require: Instances: conjunto de exemplos rotulados de tamanho N $\{(x_i, y_i), i = 1, 2, \ldots, N\}$
Θ: limiar para selecionar atributos, $0 \leq \Theta \leq 1$, valor *default* $\Theta = 1$
L: número de iterações, *default* log_2(número máximo de atributos)
Ensure: SelectedAttr : um subconjunto de atributos
1: **procedure** idttf(Instances, Θ, L)
2: Bag $\leftarrow$ BootstrapSample(Instances)
3: OutOfBag $\leftarrow$ Instances $\setminus$ Bag
4: SelectedAttr $\leftarrow \emptyset$
5: $C \leftarrow$ BuildDecisionTree(Bag)
6: $l \leftarrow 0$
7: $\theta \leftarrow \Theta \times$AUC($C$,OutOfBag)
8: **repeat**
9: AttrOnClassifier $\leftarrow$ todos os atributos que apareceram em C
10: **if** AUC(C,OutOfBag) $\geq \theta$ **then**
11: SelectedAttr $\leftarrow$ SelectedAttr $\cup$ AttrOnClassifier
12: **end if**
13: Bag $\leftarrow$ Bag $\setminus$ AttrOnClassifier
14: OutOfBag $\leftarrow$ OutOfBag $\setminus$ AttrOnClassifier
15: $C \leftarrow$ BuildDecisionTree(Bag)
16: $l \leftarrow l + 1$
17: **until** C é folha **or** não há mais atributos em Bag **or** $l \geq L$
18: **return** SelectedAttr

falsos positivos. O AUC é a medida mais comumente utilizada para se medir a curva ROC (**?**).

Inicialmente uma amostra *bootstrap* (vide Seção 3.1.1) é criada a partir de todos os exemplos, criando assim a base de treinamento (Linha 2). Os exemplos que não apareceram na base de treinamento (*Bag*) são separados e colocados na base de teste *out-of-bag* (Linha 3). A primeira árvore de decisão é então criada (Linha 5) e o seu AUC obtido é utilizado juntamente com Θ, um parâmetro do algoritmo, para inicializar o limiar θ (Linha 7). O próximo passo é selecionar os atributos, que é feito do seguinte modo: a cada iteração a árvore de decisão é avaliada pelo limiar θ (Linha 10), que limita se o atributo vai ser ou não selecionado. Todos os atributos que apareceram na árvore atual são removidos da base de treinamento (Linha 13) e da base de teste (Linha 14) e, então, uma nova árvore de decisão (Linha 15) é criada. Neste instante a iteração acaba e seu contador é incrementado (Linha 16). Esse processo se repete até que a árvore induzida seja uma folha (Linha 17) ou todos os atributos já foram removidos ou o número de iterações inserido pelo usuário seja atingido. Finalmente, todos os atributos selecionados são retornados como resultado obtido (Linha 18).

Na Tabela 3.1 é mostrado um exemplo do funcionamento do algoritmo. Considere a base de dados utilizada como exemplo na Seção 3.1.1, Figura 3.1 que possua dez atributos nomeados $a_1, a_2, \ldots, a_{10}$ e um atributo classe nomeado c. É induzida uma árvore de decisão que seleciona os atributos a_1, a_5 e a_9 com um AUC de 90% que junto com o parâmetro Θ inicializa o limiar

Tabela 3.1: *Exemplo de funcionamento do algoritmo sem considerar o número de iterações*

Iteração	Árvore	Atributos Utilizados	AUC	θ
	T_1	a_1, a_5 e a_9	90%	90%
I_1	T_2	a_4, a_2, a_{10} e a_8	75%	90%
I_2	T_3	a_6, a_7 e a_3	95%	90%
I_3	T_4	$\emptyset$	FIM	

θ: todas as árvores induzidas que possuam AUC maior que θ terão seus atributos selecionados pelo algoritmo. A primeira iteração começa analisando a árvore criada, esta possui AUC = 90% e, portanto, terá seus atributos selecionados. Ainda na primeira iteração, os atributos que apareceram na primeira árvore são removidos e a segunda árvore é criada e esta utilizou os atributos a_4, a_2, a_{10} e a_8. A segunda iteração começa analisando a segunda árvore que obteve AUC = 75%; sendo assim os seus atributos não serão selecionado pelo algoritmo, porém como antes, estes atributos são removidos da base. A próxima árvore é então induzida e assuma que os atributos que apareceram foram a_6, a_7 e a_3. A terceira iteração começa e testa se a árvore possui AUC maior que $\theta = 90\%$. Como a terceira árvore possui AUC = 95% seus atributos serão selecionados e removidos da base. No final da terceira iteração a quarta árvore é criada, porém como todos os atributos já foram removidos a árvore criada é uma folha e o algoritmo atinge seu critério de parada. Os atributos selecionados a_1, a_5, a_9, a_6, a_7 e a_3 são então retornados, nesta ordem, como resultado obtido.

A implementação do critério de parada $l \geq L$ foi feita para reduzir o custo computacional. Este critério utiliza o logaritmo, na base 2, do máximo de atributos para limitar as iterações do algoritmo e assim diminuir as iterações do algoritmo.

Na Tabela 3.2 é dado um exemplo de como o Algoritmo 1 funciona com o critério de parada $l \geq L$. O exemplo, igual ao anterior, utiliza uma base de dados com 10 atributos nomeados $a_1, a_2, \ldots, a_{10}$ e um atributo classe nomeado c. Como a base possui 10 atributos o número de iterações para o critério ser atingido é 3. É induzida uma árvore de decisão que seleciona os atributos a_1, a_5 e a_9 com um AUC de 90% que junto com o parâmetro Θ inicializa o limiar θ: todas as árvores induzidas que possuam AUC maior que θ terão seus atributos selecionados pelo algoritmo. A primeira iteração começa analisando a árvore criada, esta possui AUC = 90% e, portanto, terá seus atributos selecionados. Ainda na primeira iteração, os atributos que apareceram na primeira árvore são removidos e a segunda árvore é criada e esta utilizou os atributos a_4, a_2 e a_8. A segunda iteração começa analisando a segunda árvore que obteve

AUC = 75% sendo assim os seus atributos não serão selecionados, pelo algoritmo, porém como antes, estes atributos são removidos da base. A próxima árvore é então induzida e assuma que os atributos que apareceram foram a_6, a_7 e a_3. A terceira iteração começa e testa se a árvore possui AUC maior que $\theta = 90\%$. Como a terceira árvore possui AUC = 95% seus atributos serão selecionados e removidos da base. No final da terceira iteração a quarta árvore é criada, porém como o número de iterações foi atingido o algoritmo atinge seu critério de parada. Os atributos selecionados a_1, a_5, a_9, a_6, a_7 e a_3 são então retornados, nesta ordem, como resultado obtido.

3.2.1 Análise dos Resultados

Existem diversos modos de estudar se o conhecimento extraído resulta em algum novo conhecimento, confirmação do conhecimento ou se o conhecimento extraído foi errôneo. Entre esses modos estão: a avaliação de um especialista, análise estatísticas, comparação com a literatura, entre outros. Neste trabalho a análise estatística foi utilizada por meio do teste não-paramétrico de Friedman (**?**) considerando um nível de significância de 5%; a hipótese nula assume que todos os classificadores têm o mesmo desempenho. Se a hipótese nula é rejeitada, o teste Benjamini-Hochberg *post-hoc* (**?**) foi usado para detectar qualquer diferença estatisticamente significante entre os classificadores.

3.3 Considerações Finais

Neste capítulo foram descritos alguns trabalhos relacionados a este projeto. Também foi proposto um filtro iterativo utilizando árvores de decisão para seleção de um subconjunto de atributos, descrito no Algoritmo 1. O filtro pode utilizar qualquer classificador (com um seletor de atributos embutido) e qualquer métrica pode ser utilizada para determinar se atributo deve ser escolhido. Foi fixado, neste estudo, o algoritmo de indução de árvores de decisão J48 (vide tese:indu:j48) e AUC como métrica. No próximo capítulo serão apresentados os resultados experimentais efetuados com o Algoritmo 1, utilizando a análise de resultados aqui

Iteração	Árvore	Atributos Utilizados	AUC	θ
	T_1	a_1, a_5 e a_9	90%	90%
I_1	T_2	a_4, a_2 e a_8	75%	90%
I_2	T_3	a_6, a_7 e a_3	95%	90%
I_3	T_4	$\emptyset$	FIM	

Tabela 3.2: *Exemplo de funcionamento do algoritmo com o critério de parada $l \geq L$*

proposta.

Capítulo

4

Experimentos, Resultados e Discussão

Neste capítulo serão apresentadas as bases de dados, as configurações utilizadas nos experimentos bem como os resultados obtidos nos experimentos com a discussão dos mesmos.

4.1 Bases de Dados

Os experimentos aqui reportados utilizam 30 bases de dados. Todas representam dados médicos reais, como expressões gênicas, enquetes, diagnósticos, etc. O domínio médico geralmente impõe uma variedade de obstáculos para os algoritmos de aprendizado: alta dimensionalidade, muitos ou pouquíssimos exemplos, classes com muitas possibilidades, muitos resultados para cada classe, classes desbalanceadas, entre outros. Este tipo de dado é indicado para filtros pois possuem uma eficiência computacional melhor que *wrappers* (**?**). Maiores informações sobre as bases de dados encontram-se no Apêndice B.

4.2 Configurações Experimentais

Nesta seção uma avaliação do filtro proposto é realizada. Foram utilizados diferentes paradigmas de aprendizado de máquina e diversas bases de dados. Os paradigmas utilizados foram os descritos no Apêndice A. Todos os algoritmos foram utilizados com os parâmetros *default*, exceto o IBk onde utilizou-se $k = 3$, referido como, a partir de agora, IBk3.

Foi utilizada a plataforma Weka[1] (**?**) para executar os experimentos. O filtro proposto foi implementado como uma nova classe nesta plataforma. Como mostrado no Algoritmo 1, o limiar é inicializado utilizando como referência o AUC da primeira árvore. Duas configurações foram utilizados nos experimentos. Dois parâmetros Θ foram avaliados em 100% e 95%. Essas configurações serão referidas de agora em diante como m100 e m09 respectivamente.

Para avaliar o filtro foram testados dois aspectos diferentes:

- **A capacidade de compressão dos dados**. A capacidade de compressão pode ser definida como quão compactada uma base de dados pode ser pelo filtro, ou seja, quantos atributos o filtro remove da base de dados original, com uma perda de informação tida como não relevante. Por exemplo, uma base de dados com 1000 atributos, diz-se que o filtro obteve uma capacidade de compressão de 75% quando ao passar a base de dados pelo filtro, este cria uma base resultante com apenas 250 atributos.
- **O valor do AUC**. Neste estudo, a base de comparação para a perda de informação foi o valor AUC obtido pelo classificador induzido com todos os atributos por meio de uma validação cruzada de dez partições contra o valor do AUC obtido pela base criada após passar o filtro.

A validação cruzada feita pelo filtro ocorre da seguinte maneira. Primeiramente a base de dados é dividida em dez partições, destes apenas nove são considerados como a base de dados original, ou seja, a décima partição não é vista pelo filtro. Assim, o subconjunto criado utiliza apenas nove partições. Após a criação os atributos são removidas das nove partições juntamente com a partição remanescente. O subconjunto com nove partições é criado e apresentado a um dos indutores previamente mencionados e AUC é então calculado em cima da partição remanescente. Este processo é repetido dez vezes e os resultados consolidados por meio da média.

O teste *pós-hoc* mostra se o termo na linha é melhor ou pior, significativamente ou não em relação ao termo na coluna. Nas tabelas de *pós-hoc*:

- $\triangle$ indica que a linha é melhor que a coluna, mas não significativamente
- $\triangledown$ indica que a linha é pior que a coluna, mas não significativamente
- $\blacktriangle$ indica que a linha é melhor significativamente que a coluna
- $\blacktriangledown$ indica que a linha é pior significativamente que a coluna

[1] www.cs.waikato.ac.nz/~ml/weka

Tabela 4.1: *Benjamini-Hochberg post-hoc Test - Tempos dos Seletores para todas as bases / baixa densidade / altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT
CFS	○/○/○	▼/▼/▼	▲/▽/▲	▲/▽/▲	▲/▽/▲	△/▼/△
GAIN RATIO		○/○/○	▲/△/▲	▲/▲/▲	▲/▲/▲	▲/△/▲
RELIEF F			○/○/○	▽/△/▼	▽/△/▼	▼/▽/▼
M100				○/○/○	▽/▽/△	▼/▽/▼
M095					○/○/○	▼/▽/▼
UT						○/○/○

– ○ indica que não houve diferença entre a linha e a coluna

Durante os experimentos foram utilizados para a comparação os seletores CFS, *Gain Ratio*, ReliefF descritos na Seção 2.3, as versões do Algoritmo 1 m100 e m095 e uma nova versão onde o Algoritmo 1 faz somente uma iteração ($L = 1$) criando uma única árvore, essa versão foi denominada UT.

4.3 Resultados e Discussão

Utilizando as configurações experimentais descritas na Seção 4.2 e o Algoritmo 1 foram feitas duas análises preliminares reportadas nos Apêndices C e D.

Após avaliar o comportamento do algoritmo em relação à diversos paradigmas de aprendizado de máquina, uma avaliação comparando o Algoritmo 1 com outros seletores de atributos foi realizado. Esses seletores, descritos na Seção 2.3, foram utilizados para tal comparação. Para avaliar o comportamento do Algoritmo 1 contra esses seletores foram utilizadas três métricas: (i) tempo que cada seletor levou para selecionar os atributos; (ii) a porcentagem de atributos selecionados; e (iii) análise estatística comparando o Algoritmo 1 e os seletores de diversos paradigmas do aprendizado de máquina. Essas três métricas foram obtidas por meio do Experimento III, descrito no Apêndice E.

Os detalhes da primeira métrica podem ser encontrados no Apêndice E, porém como é possível observar na Tabela 4.1 o seletor *Gain Ratio* teve o melhor tempo em relação a todos os outros, *Relief F* obteve os piores tempos em todos os grupos testados. Em relação aos tempos do Algoritmo 1 em suas duas configurações m100 e m095 para bases de baixa densidade os resultados foram diferenças não significativas exceto para o *Gain Ratio*. A metodologia de árvore única apresentou um desempenho bom em relação quase todos os outros seletores.

Na Tabela 4.2 foi analisada a capacidade de compressão dos seletores. Não houve diferença

Tabela 4.2: *Benjamini-Hochberg post-hoc Test - Capacidade de Compressão dos Seletores para todas as bases / baixa densidade / altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT
CFS	○/○/○	△/△/△	▲/△/▲	△/▽/▲	△/▽/▲	▽/▼/△
GAIN RATIO		○/○/○	△/△/△	△/▽/△	△/▽/△	▽/▼/▽
RELIEF F			○/○/○	▽/▼/▽	▽/▼/▽	▼/▼/▼
M100				○/○/○	○/○/○	▽/▽/▽
M095					○/○/○	▽/▽/▽
UT						○/○/○

Tabela 4.3: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo J48 para todas as bases / baixa densidade / altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT	J48
CFS	○/○/○	△/△/△	△/△/△	△/△/△	△/△/△	▲/▲/▲	△/△/△
GAIN RATIO		○/○/○	△/△/△	△/△/△	△/△/△	▲/▲/△	▽/▽/△
RELIEF F			○/○/○	▽/▽/▽	▽/▽/▽	▲/▲/△	▽/▽/▽
M100				○/○/○	○/○/○	▲/▲/△	▽/▽/△
M095					○/○/○	▲/▲/△	▽/▽/△
UT						○/○/○	▼/▼/▲
J48							○/○/○

significativa entre os seletores; a exceção foi o *Relief F* que principalmente para as bases de baixa densidade teve um desempenho ruim. Quanto às metodologias m100 e m095 elas não apresentaram diferenças significativas para as baixas densidades, porém em relação ao CFS para as bases de alta densidade a diferença foi significativa em vantagem para o CFS. A metodologia de árvore única se mostrou muito eficiente em termos de compactação de dados, obtendo para as bases de baixa densidade sempre uma vantagem significativa em relação aos outros seletores.

No geral o Algoritmo 1 apresentou um tempo relativamente pequeno, enquanto que conseguiu compactar os dados em mais de 99%. Na média tempo/compactação, o Algoritmo 1 ficou atrás apenas dele mesmo fixando uma única iteração, como era esperado. A análise dessas comparações pode ser encontrada no Apêndice E. A seguir é mostrada a análise estatística utilizando o teste não paramétrico de Friedman e o teste Benjamini-Hochberg *post-hoc* desses resultados.

Analisando os testes *post-hoc* mostrados nas Tabelas 4.3, 4.4, 4.6, 4.5 e 4.7 é possível apurar que para todos os seletores o filtro proposto mostrou consistência sempre obtendo resultados sem diferença estatisticamente significativa. Para os indutores IBk3, NB e PART os seletores

Tabela 4.4: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo IBk3 para todas as bases, as de baixa densidade e as de altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT	IBK3
CFS	○/○/○	△/△/△	△/△/△	△/▽/△	△/▽/△	▲/▲/▲	△/△/△
GAIN RATIO		○/○/○	○/○/○	▽/▽/▽	▽/▽/▽	▲/▲/▲	▽/▽/▽
RELIEF F			○/○/○	▽/▽/▽	▽/▽/▽	▲/▲/▲	▽/▽/▽
M100				○/○/○	○/○/○	▲/▲/▲	△/△/△
M095					○/○/○	▲/▲/▲	△/△/△
UT						○/○/○	▼/▼/▼
IBK3							○/○/○

Tabela 4.5: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo PART para todas as bases, as de baixa densidade e as de altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT	PART
CFS	○/○/○	△/○/△	△/△/△	△/▽/△	△/▽/△	▲/▲/△	△/△/△
GAIN RATIO		○/○/○	▽/△/▽	▽/▽/▽	▽/▽/▽	▲/▲/△	△/△/▽
RELIEF F			○/○/○	▽/▽/▽	▽/▽/▽	▲/▲/△	△/△/▽
M100				○/○/○	○/○/○	▲/▲/△	△/△/▽
M095					○/○/○	▲/▲/△	△/△/▽
UT						○/○/○	▼/▼/▽
PART							○/○/○

Tabela 4.6: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo Naive Bayes para todas as bases, as de baixa densidade e as de altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT	*Naive Bayes*
CFS	○/○/○	△/△/△	△/△/△	△/△/△	△/△/△	▲/▲/△	△/△/△
GAIN RATIO		○/○/○	▽/○/▽	▽/▽/▽	▽/▽/▽	▲/▲/△	▽/▽/▽
RELIEF F			○/○/○	▽/▽/▽	▽/▽/▽	▲/▲/△	▽/▽/▽
M100				○/○/○	○/○/○	▲/▲/△	△/△/△
M095					○/○/○	▲/▲/△	△/△/△
UT						○/○/○	▼/▼/▽
Naive Bayes							○/○/○

Tabela 4.7: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus outros Seletores e o Algoritmo SMO para todas as bases, as de baixa densidade e as de altas densidades respectivamente*

	CFS	GAIN.RATIO	RELIEF.F	M100	M095	UT	SMO
CFS	○/○/○	▼/▼/▽	▼/▽/▽	▽/▽/△	▽/▽/△	▲/▲/△	▽/▽/▽
GAIN RATIO		○/○/○	△/△/○	△/△/△	△/△/△	▲/▲/▲	△/△/△
RELIEF F			○/○/○	△/△/△	△/△/△	▲/▲/▲	△/△/△
M100				○/○/○	○/○/○	▲/▲/△	▽/△/▽
M095					○/○/○	▲/▲/△	▽/△/▽
UT						○/○/○	▼/▼/▼
SMO							○/○/○

do filtro proposto mostrou melhor resultado que os outros seletores; em relação a UT apesar de se obter uma compressão maior com ela o teste *post-hoc* mostrou que esse tipo de abordagem gera resultados muito ruim em termos de AUC tendo resultados significativemente pior em todos os seletores e até mesmo contra o indutor sozinho. Em relação a comparação seletor e indutor os seletores do filtro conseguiram ser melhor, mesmo que não significativamente, em todos os casos exceto o J48 e o SMO.

4.4 Considerações Finais

Neste capítulo foram apresentadas as bases de dados que foram utilizadas nos experimentos, as configurações experimentais utilizadas, os resultados preliminares foram discutidos brevemente usando como teste diversas bases de dados biomédicos. O filtro proposto foi avaliado e sua capacidade de compressão e desempenho foram testados em cinco diferentes paradigmas de aprendizado de máquina. Os resultados do trabalho foram apresentados testando o desempenho do filtro proposto comparando com outros seletores de atributo da literatura e com o indutor sozinho. Os resultados mais detalhados assim como alguns outros testes realizados são apresentados nos apêndices deste trabalho.

Capítulo

5

Conclusão

Esta dissertação teve como objetivo propor uma abordagem utilizando árvores de decisão em um filtro iterativo, visando auxiliar na extração de informação em grande bases biológicas. Com uma base de menor dimensionalidade, um especialista humano pode entender melhor ou ainda utilizar um algoritmo de aprendizado de máquina de forma mais eficaz. Diversas etapas tiveram que ser realizadas para alcançar tal objetivo: a primeira delas foi conhecer o estado atual da área utilizada no estudo por meio de uma revisão bibliográfica. Essa revisão mostrou que apesar de não ser uma área nova, pelo contrário, ela é muito explorada, ainda não existe um consenso na literatura sobre qual é o melhor método ou o melhor caminho para se fazer uma seleção de atributos eficiente. Cada algoritmo ou metodologia proposta excede em algumas áreas deixando a desejar em outras. Na área de bioinformática em especial, existem diversos métodos para se selecionar atributos, dado que este passo dentro de todo processo da bioinformática está se tornando cada vez mais importante visto que as bases estão cada vez maiores e analisar todo o seu conteúdo sem um pré-processamento adequado acaba se tornando muito custoso para o pesquisador. Feita a revisão, o aluno passou a etapa de desenvolvimento do algoritmo, um filtro iterativo que utiliza árvores de decisão. O algoritmo produzido tem como propósito selecionar um subconjunto de atributos da base original, tentando assim selecionar somente os atributos mais relevantes da base de dados original. Para chegar em tal

subconjunto o algoritmo primeiramente faz uma amostra *bootstrap* da base e depois passa pelo filtro que classifica se os atributos que foram apresentados ao filtro são relevantes, a cada iteração os atributos que foram apresentados ao filtro são removidos e novos atributos são escolhidos para passarem pelo filtro. Assim que um dos critérios de parada é alcançado o algoritmo termina.

Os atributos selecionados são *rankeados* e geram uma nova base de dados, que geralmente, é menor que a original. Os experimentos feitos tiveram como objetivo avaliar a capacidade de compressão do filtro e se os atributos por ele selecionados. Um último experimento feito comparou o filtro por este estudo proposto com outros seletores de atributos na literatura.

5.1 Principais Resultados

Nos dois primeiros experimentos que analisavam o desempenho do filtro comparando com a base original é possível observar que o filtro, para as grandes bases de dados, ou seja as que possuíam uma baixa densidade, o filtro conseguia selecionar um grupo pequeno como subconjunto, tendo um tamanho, em média, de 16% da base original no primeiro experimento e no segundo onde um novo critério de parada foi implementado o tamanho ficou, em média, 99.5% da base original. Além da capacidade de compressão, uma análise estatística foi realizada nos dois experimentos e mostrou que na maioria dos casos o Algoritmo 1 consegue, com seu tamanho reduzido, não apresentar diferenças significantes em relação à base original, sendo em alguns casos até melhor.

O terceiro experimento que compara o Algoritmo 1 com os algoritmos: CFS, Gain Ratio e ReliefF, nos aspectos de tempo, de AUC e de capacidade de compressão. O experimento mostrou que o algoritmo consegue fazer um tempo mediano em relação aos três outros algoritmos sendo mais rápido em alguns casos e mais lento em outros. Quanto a análise estatística do AUC o algoritmo apresentou em média para as bases de baixa densidade o segundo melhor AUC e em alguns casos alcançou o melhor AUC entre os algoritmos testados. Quanto a capacidade de compressão ele atingiu a melhor capacidade dentre os 4 algoritmos com a média de 99.5% de compressão dos dados. Quando analisamos todos os dados em conjunto é possível notar que o filtro proposto consegue ter um ótimo desempenho conseguindo, em um tempo razoavelmente bom, comprimir bem os dados e atingir um AUC bom.

5.2 Publicações

Até o presente foram publicados 3 artigos pelo aluno. No primeiro (**?**) é discutida uma versão inicial do Algoritmo 1 proposto nesse estudo. Nele, dez árvores de decisão são criadas iterativamente, a cada iteração os atributos que aparecem na árvore são retirados da base de treinamento. Com isso, foi possível perceber que algumas árvores intermediárias possuíam resultados melhores que o das primeiras. A melhor árvore encontrada também obteve um resultado melhor que algumas abordagens usando apenas 1-2 atributos (genes) da literatura, por exemplo, (**?**; **?**; **?**). No segundo trabalho, não relacionado a este estudo, porém publicado em sua duração, (**?**) foi criada uma hierarquia gerada por meio de mineração de texto, utilizando como base laudos médicos do Hospital das Clínicas de Ribeirão Preto. Uma versão inicial de um algoritmo foi proposto com o objetivo final de preencher automáticamente laudos médicos. O trabalho que foi alvo do estudo do aluno durante a tese de conclusão de curso foi melhorado e publicado. No terceiro (**?**) o aluno publicou a parte inicial deste trabalho, uma versão inicial do Algoritmo 1 e o Experimento I (vide C) assim como seus resultados e discussão foram publicados neste artigo.

5.3 Trabalhos Futuros

Muito pode ser realizado em pesquisas futuras considerando o algoritmo proposto. Em sua versão atual o Algoritmo 1 possui 3 critérios de parada, porém um estudo mais aprofundado pode ser capaz de identificar um critério de parada novo e que pode melhorar ainda mais o desempenho do algoritmo. Outro ponto que pode ser modificado é como a amostra *bootstrap* é feita no filtro: nesta versão a amostra é tirada apenas um vez e a partir dela as árvores são induzidas e filtradas. O que pode ser feito é colocar o *bootstrap* dentro da iteração fazendo com que em cada iteração uma nova amostra *bootstrap* seja realizada e analisar o custo-beneficio dessa alteração. Para finalizar é possível testar o filtro proposto em outras áreas de conhecimento, neste estudo o foco foi a área médica. Porém o Algoritmo 1 funciona para qualquer tipo de base de dados, contando que esta base possua uma classe discreta. Assim sendo pode-se testar o seu comportamento em outras áreas e comparar se o filtro tem um bom desempenho.

Referências

Aggarwal, C. C. (2012). *Mining text data*. Springer Science+ Business Media. [Citado na página 2]

Aha, D., Kibler, D. & Albert, M. (1991). Instance-based learning algorithms. *Machine learning 6*(1), 37–66. [Citado na página 37]

Bacauskiene, M., Verikas, A., Gelzinis, A. & Valincius, D. (2009). A feature selection technique for generation of classification committees and its application to categorization of laryngeal images. *Pattern Recognition 42*(5), 645–654. [Citado na página 8]

Bamber, D. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of mathematical psychology 12*(4), 387–415. [Citado na página 17]

Baranauskas, J. A. (2001). *Extração Automática de Conhecimento Utilizando Múltiplos Indutores*. Ph. D. thesis, ICMC-USP. http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08102001-112806/. [Citado na página 5]

Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B 57*, 289–300. [Citado nas páginas 19 e 52]

Blum, A. L. & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence 97*(1–2), 245–271. [Citado na página 8]

Bolón-Canedo, V., Sánchez-Marono, N. & Alonso-Betanzos, A. (2012). An ensemble of filters and classifiers for microarray data classification. *Pattern Recognition 45*(1), 531–539. [Citado na página 14]

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression*

Trees. Pacific Grove, CA: Wadsworth & Books. [Citado na página 45]

Broad Institute (2010). Cancer program data sets. http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. [Citado na página 49]

Chapelle, O., Schölkopf, B., Zien, A. et al. (2006). *Semi-supervised learning*, Volume 2. MIT press Cambridge. [Citado na página 2]

Chow, M., Moler, E. & Mian, I. (2001). Identifying marker genes in transcription profile data using a mixture of feature relevance experts. *Physiol. Genomics 5*, 99–111. [Citado nas páginas 15 e 29]

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, San Francisco, CA, pp. 115–123. Morgan Kaufmann. [Citado na página 35]

Efron, B. & Tibshirani, R. (1994). An introduction to the bootstrap (chapman & hall/crc monographs on statistics & applied probability). [Citado na página 15]

Efron, B. & Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association 92*(438), 548–560. [Citado na página 16]

Fayyad, U. M., Djorgovski, S. G. & Weir, N. (1996). From digitized images to on-line catalogs: Data mining a sky survey. *AI Magazine 17*(2), 51–66. [Citado na página 1]

Frank, A. & Asuncion, A. (2010). UCI machine learning repository. http://archive.ics.uci.edu/ml. [Citado na página 49]

Frank, E. & Witten, I. (1998). Generating accurate rule sets without global optimization. [Citado na página 35]

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics 11*(1), 86–92. [Citado nas páginas 19 e 50]

Gamberger, D., Lavrač, N., Zelezný, F. & Tolar, J. (2004). Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. *Journal of Biomedical Informatics 37*(4), 269–284. [Citado nas páginas 15 e 29]

Gao, K., Khoshgoftaar, T. & Van Hulse, J. (2010). An evaluation of sampling on filter-based feature selection methods. In *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference*, pp. 416–421. [Citado na página 16]

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller,

H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. & Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science 286*, 531–537. [Citado nas páginas 15 e 29]

Hall, M. A. & Smith, L. A. (1997). Feature subset selection: a correlation based filter approach. [Citado na página 11]

Hall, M. A. & Smith, L. A. (1998). Practical feature subset selection for machine learning. [Citado na página 11]

Han, J., Kamber, M. & Pei, J. (2011). *Data mining: concepts and techniques*. Morgan Kaufmann. [Citado na página 5]

Hinton, G. E. & Sejnowski, T. J. (1999). *Unsupervised learning: foundations of neural computation*. The MIT press. [Citado na página 2]

John, G. & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence*, pp. 338–345. Morgan Kaufmann Publishers Inc. [Citado nas páginas 5, 39, e 41]

Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. Wiley-IEEE Press. [Citado na página 21]

Kohavi, R. & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence 97*(1–2), 273–324. [Citado na página 3]

Langley, P. (1996). *Elements of Machine Learning*. Morgan Kaufmann Publishers, Inc. [Citado nas páginas v, 6, e 7]

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J., Armañanzas, R., Santafé, G., Pérez, A. et al. (2006). Machine learning in bioinformatics. *Briefings in bioinformatics 7*(1), 86–112. [Citado na página 2]

LaTorre, A., Peña, J., González, S., Cubo, O. & Famili, F. (2007). Breast cancer biomarker selection using multiple offspring sampling. In *2 ND WORKSHOP IN DATA MINING IN FUNCTIONAL GENOMICS AND PROTEOMICS*, pp. 23. [Citado na página 13]

Mamitsuka, H. (2006). Selecting features in microarray classification using roc curves. *Pattern Recognition 39*(12), 2393–2404. [Citado nas páginas 8 e 14]

Mitchell, T. (Ed.) (1997). *Machine Learning*. McGraw Hill. [Citado na página 8]

Netto, O., Macedo, A., de Azevedo Marques, P. & Baranauskas, J. (2011). Uma metodologia para estruturaç ao de laudos médicos usando ontologias. pp. 10. [Citado na página 29]

Netto, O. P. & Baranauskas, J. A. (2012). An iterative decision tree threshold filter. *Proceedings of International Conference on Machine Learning*. Submetido. [Citado na página 29]

Netto, O. P., Nozawa, S. R., Mitrowsky, R. A. R., Macedo, A. A. & Baranauskas, J. A. (2010). Applying decision trees to gene expression data from DNA microarrays: A leukemia case study. In *XXX Congress of the Brazilian Computer Society, X Workshop on Medical Informatics*, Belo Horizonte, MG, pp. 10. [Citado nas páginas 15, 29, e 47]

Oshiro, T. M., Perez, P. S. & Baranauskas, J. A. (2012). How many trees in a random forest? *In Proceedings of the 8th International Conference on Machine Learning and Data Mining*. Submitted. [Citado na página 47]

Peng, Y., Wu, Z. & Jiang, J. (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics 43*(1), 15–23. [Citado na página 14]

Quinlan, J. R. (1993). C4.5*: Programs for Machine Learning*. Morgan Kaufmann. San Francisco, CA. [Citado nas páginas 35 e 45]

Rezende, S. O. (Ed.) (2003). *Sistemas Inteligentes - Fundamentos e Aplicações*. Manole. [Citado na página 8]

Saeys, Y., Inza, I. & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics 23*(19), 2507–2517. [Citado nas páginas 3, 7, e 13]

Van't Veer, L., Dai, H., Van De Vijver, M., He, Y., Hart, A., Mao, M., Peterse, H., Van Der Kooy, K., Marton, M., Witteveen, A. et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *nature 415*(6871), 530–536. [Citado na página 13]

Wang, G. & Song, Q. (2012). Selecting feature subset for high dimensional data via the propositional foil rules. *Pattern Recognition*. [Citado na página 15]

Wang, Y. & Makedon, F. (2004). Application of relief-f feature filtering algorithm to selecting informative genes for cancer classification using microarray data. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pp. 497–498. IEEE. [Citado na página 11]

Wang, Y., Tetko, I., Hall, M., Frank, E., Facius, A., Mayer, K. & Mewes, H. (2005). Gene selection from microarray data for cancer classificationa machine learning approach. *Computational Biology and Chemistry 29*(1), 37–46. [Citado na página 8]

Witten, I. H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and*

Techniques, Second Edition. Morgan Kaufmann. [Citado nas páginas 9, 22, 43, e 45]

Yang, Q., Shao, J., Scholz, M. & Plant, C. (2011). Feature selection methods for characterizing and classifying adaptive sustainable flood retention basins. *Water research 45*(3), 993–1004. [Citado na página 9]

Apêndice

A

Indutores

Para fazer a avaliação dos resultados obtidos pela a metodologia proposta foram utilizados diversos indutores, descritos a seguir.

A.1 PART

O PART é um algoritmo criado por (**?**) com o paradigma de regras, ou seja, ele cria regras para tentar classficar os exemplos. A criação do PART veio visando buscar um algoritmo mais simples para aprendizado por regra. Os dois algoritmos principais para aprendizado por regra eram o C4.5 (**?**) e o RIPPER (**?**), ambos funcionam em duas etapas. Na primeira etapa eles geram um conjunto de regras e depois, por meio de uma fase um tanto quanto complexa de otimização, eles mostram as regras que são as melhores. O C4.5 após obter o conjunto de regras transforma-o de árvores de decisões não podadas em regras de decisão. Depois cada uma das regras é simplificada e apenas as que gerarem um bom subconjunto para cada classe é escolhida. Após essa seleção as regras vão sendo deletadas uma a uma desde que isso faça com que o erro diminua no conjunto de treinamento. Um dos modos de evitar essa abordagem de gerar as regras e depois ir retirando-as é utilizar a abordagem de dividir e conquistar, esse tipo de abordagem gera uma regra por vez, removendo os exemplos que nela apareceram e iterativamente construindo mais regras.

O RIPPER utiliza essa estratégia de dividir e conquistar criando um subconjunto do conjunto de treinamento para determinar quando uma conclusão de uma regra é excluida e incorpora um tamanho máximo como critério de parada. Na fase de otimização o RIPPER revisa a regra e para aproxima-la

O PART utiliza uma combinação entre o C4.5 e o RIPPER numa tentativa de evitar os problemas de ambas as abordagens. Para isso ele utiliza árvores de decisão parciais e assim sendo não utiliza uma otimização global adicionando a sua principal vantagem, a simplicidade. A tática do dividir para conquistar é utilizada da seguinte maneira, a cada nova regra induzida todos os exemplos cobertos por aquela regra são retiradas da base de dados e continua gerando regras até que nenhum exemplo tenha sobrado na base de dados. A diferença está em como estas regras são criadas.

Basicamente a regra é criada a partir de uma árvore de decisão normal, porém apenas uma regra é criada e para selecionar esta regra a folha da árvore com a maior cobertura, ou seja, a que abrange mais exemplos é a escolhida. O resto da árvore de decisão criada é descartada. Esse tipo de abordagem evita com que as árvores se tornem extremas, abrangendo um número muito grande com poucos exemplos em cada ou abrangendo muitos exemplos diferentes com um erro muito alto. Adicionando a árvores de decisão a ideia de dividir e conquistar acrescenta a flexibilidade das árvores a velocidade da abordagem. Porém, utilizar uma árvore de decisão inteira não é o melhor jeito de se criar essas regras e utilizar árvores de decisão parciais acelera o processo sem sacrificar as vantagens citadas.

A ideia então é criar uma árvore de decisão parcial, ao invés de uma completa. Uma árvore de decisão parcial pode ser vista como uma árvore de decisão normal que contém diversos ramos e subárvores, porém para gerar uma árvore parcial é preciso integrar o processo de criação da árvore com as operações de poda para achar uma subárvore que consiga abranger um número de exemplos uma vez que esta subárvore é encontra o algoritmo PART para e a regra simples é criada.

Na Figura A.1 é mostrada uma árvore criada normalmente (passo 1) exceto que ao invés de expandir todos os nós como seria normal, a árvore expande apenas o nó com menor entropia, no caso o nó 3. Desse nó sai um nó folha e o nó 5 (passo 2), a expansão continua apesar do nó folha possuir menor entropia, por ser folha, ele não pode ser expandido. Logo o nó 5 é expandido em 2 nós folhas (passo 3). Nesse momento o algoritmo atinge um ponto onde não pode mais expandir e começa a podar a árvore. Suponha que o nó 5 possa ser podado,

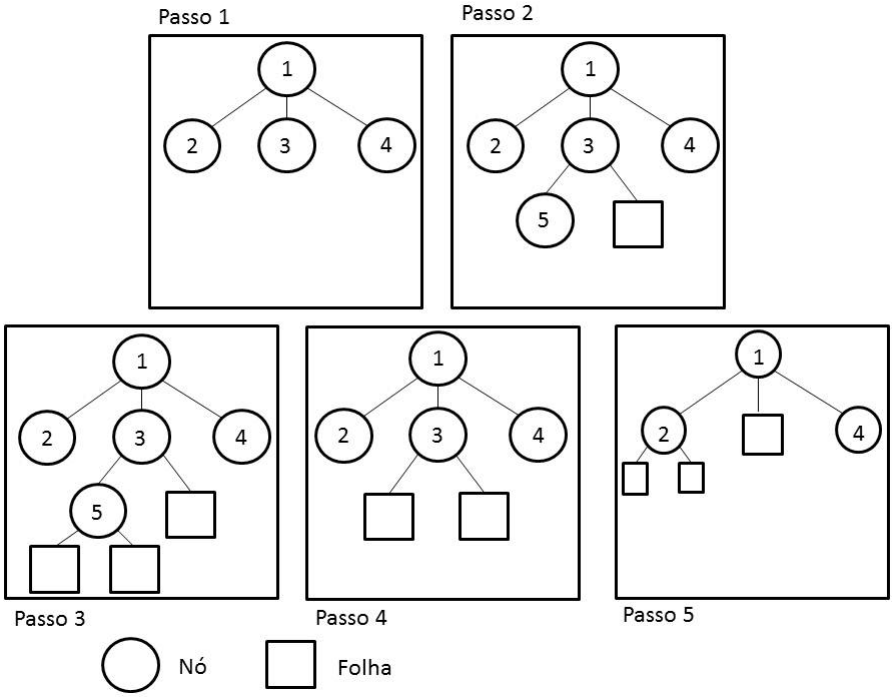

Figura A.1: *Exemplo de PART*

o algoritmo então poda o nó (passo 4) e continua a retroceder até o nó 3, verificando se este pode ser podado. Assumindo que o nó 3 possa também ser podado ele é podado e substituído por uma folha. O algoritmo agora verifica os outros dois nós pelo o de menor entropia. Assuma que o nó 2 possui a menor entropia ele então é expandido em dois nós folha, por não poder mais expandir o algoritmo agora tenta podar o nó 2. Supondo que este nó não pode ser podado o algoritmo termina, criando uma árvore parcial de 3 folhas no passo 5.

A.2 KNN

Algoritmos IBL (*Instance-based Learning*) são derivados do padrão de classificação conhecido como vizinhos próximos ou KNN (*K Nearest Neighbors*) é um algoritmo do paradigma baseado em memorização, ou seja, o indutor criado classifica novos exemplos utilizando os exemplos já presentes na base (**?**). No caso do KNN ele analisa os vizinhos mais próximos ao novo exemplo e classifica esse novo exemplo com a classe majoritária dos vizinhos próximos. O número de vizinhos analisados é dado como um parâmetro do algoritmo representado pela letra K.

Os algoritmos IBL apresentam alguns problemas como o fato de serem computacionalmente custosos por guardarem todas os exemplos de treinamento e são intolerante a atributos irrelevantes.

O algoritmo utlizado neste trabalho foi o IBk3 que utiliza os 3 vizinhos mais próximos para classificar os exemplos, as principais caracteristicas do IBk3 são: mantém um *ranking* de classificação de cada exemplo salvo, por exemplo, ele guarda o número de classificações corretas e incorretas daquele exemplo; utiliza um teste de significância para determinar quais exemplos servem como bons classificadores e quais exemplos podem ser ruídos. Os exemplos bons são utilizados para classificar outros exemplos presentes, as consideradas ruídos são descartados.

O IBk3 tem dois parâmetros, o primeiro é um nível de confiança que determina se o exemplo é bom o suficiente para ser utilizado para classificar outros exemplos, o valor padrão é 90%. O segundo parâmetro é um nível de confiança que exclui o exemplo, ou seja classifica-o como ruído, esse parâmetro tem o valor padrão em 75%. Com esses valores um exemplo necessita ter um nível de confiança muito alto para ser considerada boa como classificador, porém para não ser considerado um ruído ele necessita apenas de um nível de confiança moderado. A seguir um exemplo simplificado de como algoritmos IBL funcionam.

Na Figura A.2 é mostrado como um novo exemplo, representado pelo quadrado azul, é classificado pelo algoritmo KNN usando $k = 1$, assim sendo somente o exemplo mais próximo é analisado para classificar o novo exemplo. Nesse caso o quadrado azul será classificado como a classe triângulo preto.

Na Figura A.3 o K é mudado para 3 ao invés de 1 e com isso o novo exemplo, ao invés de ser classificado como triângulo preto, agora é classificado como círculo vermelho. Pois apesar do exemplo mais próximo ser um triângulo os outros dois exemplos mais próximos são círculos, como o novo exemplo é classificado usando a maioria dos exemplos analisados o novo exemplo é classificado como círculo vermelho. Pela utilização desse método de classificar por maioria,

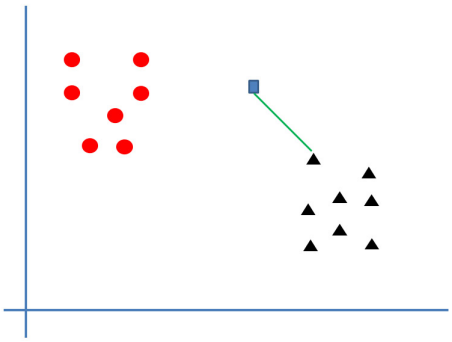

Figura A.2: *Exemplo de KNN com 1 vizinho*

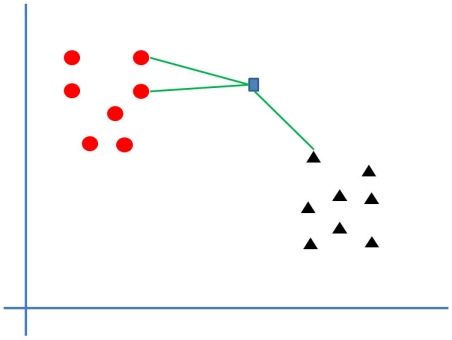

Figura A.3: *Exemplo de KNN com 3 vizinho*

geralmente o K escolhido como parâmetro é ímpar.

A.3 Naive Bayes

O *Naive Bayes* ou NB foi criado como um algoritmo de aprendizado estatístico utilizando como base uma rede Bayesiana, o termo *Naive* ou ingênuo no português se da pelo fato que o indutor criado assume que todos os atributos são independentes, ou seja a informação de um atributo não tem relação com as informações contidas no outro atributo dada uma classe (**?**), o que raramente ocorre. O algoritmo tem como objetivo calcular a probabilidade que uma amostra desconhecida pertença a cada uma das classes possíveis, ou seja, predizer a classe mais provável. Este tipo de predição é chamada de classificação estatística, pois é completamente baseada em probabilidades.

Primeiramente para classificar um conjunto de teste, ou seja uma amostra desconhecida o algoritmo procede da seguinte maneira: cada classe do conjunto de treinamento tem sua probabilidade calculada. Na maioria dos casos apenas duas classes são consideradas como, por exemplo, uma classe que indica se o paciente está doente ou saudável baseados nos sintomas que ele apresenta ou em um diagnóstico prévio. Para se calcular a probabilidade, o cálculo é feito dividindo-se o número de exemplos de determinada classe pelo número total de exemplos do conjunto de treinamento, como é possível observar na equação:

$$\text{P(c1) = número de exemplos de c1 / número de exemplos} \quad to \tag{A.2}$$

Uma vez calculadas essas probabilidades o próximo passo é calcular a probabilidade de cada atributo do conjunto de teste em relação a cada classe. Esse passo é o mais pesado computacionalmente do algoritmo, onde para cada valor do atributo é necessário calcular a probabilidade dele aparecer em cada classe. Suponha que a base possua um atributo binário, ou seja com dois valores possíveis (verdadeiro ou falso) e uma classe também com dois valores possíveis (verdadeiro ou falso). Suponha também que esse conjunto de testes tenha 4 exemplos como pode se observar na Tabela A.1. Nesse exemplo para o valor V do Atributo1 a probabilidade de a classe ser V é de 66,67% enquanto que para a classe F o valor V do Atributo 1 tem probabilidade complementar de 33,33%. O mesmo procedimento é feito com o valor F do Atributo1 sendo que para a classe F o valor tem probabilidade de 100% e assim sendo para o valor V tem probabilidade de 0%.

Como podemos perceber com este exemplo simples quanto mais atributos o conjunto de exemplos tiver ou mesmo quanto mais diversificados os valores de cada atributo for, mais tempo computacional será gasto calculando estas probabilidades.

O último passo para classificar um exemplo desconhecido é utilizar as probabilidades já calculadas, para cada atributo a probabilidade daquele valor ser de uma determinada classe é multiplicada com as outras probabilidades dos valores dos outros atributos todos para a mesma classe. Após todas as probabilidades serem multiplicadas, o resultado é multiplicado pela probabilidade que foi calculada no começo, ou seja a probabilidade de ser aquela classe em relação a todas as demais classes.

Para simplificar vamos utilizar o conjunto de teste que está exposto na Tabela A.1 como nosso conjunto de treinamento. Suponha que um exemplo desconhecido seja introduzida ao algoritmo. O primeiro passo é calcular a probabilidade das classes, nesse caso é 50% para cada uma. O segundo passo é calcular a probabilidade do valor do Atributo1 do nosso exemplo desconhecido, suponha que esse valor seja V. Assim a probabilidade dela ser da classe V é de 66,67% e do exemplo ser da classe F é de 33,33%. O próximo passo é multiplicar a

Exemplo	Atributo1	Classe
1	V	V
2	F	F
3	V	V
4	V	F

Tabela A.1: *Conjunto de teste: Exemplo*

probabilidade com as dos outros atributos, como nosso exemplo só possui 1 atributo essa etapa é ignorada e assim sendo é feita a multiplicação das probabilidades com as probabilidades de ser cada classe. Neste nosso exemplo cada classe tem uma probabilidade de 50%. Logo a probabilidade de nosso exemplo desconhecido ser da classe V é de $66,67\% \times 50\%$ que dá $33,34\%$ e a probabilidade de ser da classe F é de $33,33\% \times 50\%$ que dá $16,66\%$. Feita esta multiplicação o exemplo é classificado com a classe que tiver a maior probabilidade, neste caso o exemplo desconhecido seria classificada como sendo da classe V.

A.4 SMO

Sequential Minimal Optimization ou SMO é um algoritmo que treina uma máquina de vetores de suporte (conhecidos como SVM - *Support Vector Machine*). SVMs são algoritmos de aprendizado supervisionados que geram classificadores binários e lineares, ou seja, classificadores que separam exemplos em duas classes, normalmente denominadas de classe positiva e classe negativa. A separação é efetuada por meio de uma fronteira linear, ou seja, um hiperplano posicionado entre os exemplos das duas classes. Na Figura A.4, um exemplo bidimensional é mostrado. Uma fronteira (reta sólida) que rotula os exemplos à esquerda como triângulos (classe positiva) e exemplos à direita como círculos (classe negativa) é o classificador mostrado na Figura A.5. Nesse caso a distância de um exemplo à reta (hiperplano em 3 ou mais dimensões) separadora pode ser utilizada como uma métrica de confiança do classificador SVM (**?**). Os algoritmos de SVM constroem um hiperplano no espaço de exemplos que tenta separar os exemplos de treinamento da classe positiva dos exemplos de treinamento da classe negativa de forma tal que o hiperplano esteja o mais afastado possível dos exemplos de treinamento, definindo a margem máxima, conforme mostrado na Figura A.6. Os exemplos mais próximos do hiperplano são chamados de vetores de suporte, dando nome a esta família de algoritmos.

A minimização de erros de classificação no conjunto de treinamento e a maximização da margem são dois fundamentos que procuram definir um limite superior para o erro do classificador em exemplos não vistos durante o treinamento. Esses dois fundamentos, por sua vez, podem ser vistos como um problema de otimização na qual assume-se um conjunto de treinamento (x, y) linearmente separável no qual o rótulo da classe y igual a $+1$ para a classe positiva e a -1 para a classe negativa. Assumindo que $w = (w_1, w_2, ..., w_n)$ e $x = (x_1, x_2, ..., x_n)$ vetores de n elementos, pode-se mostrar que o hiperplano $w \bullet x + b$ que resolve o seguinte problema de otimização: minimizar $w \bullet w$ sujeito a $y_i(w \bullet x_i + b) \geq 1, i = 1, 2, ..., n$ minimiza

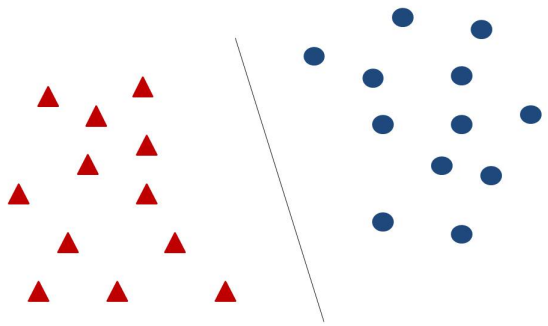

Figura A.4: *Exemplo bidimensional*

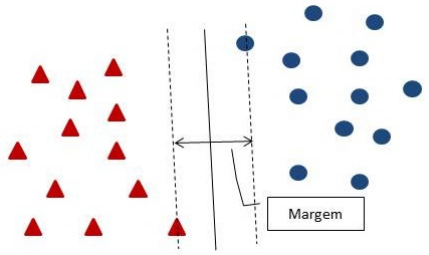

Figura A.5: *Classificador de margem qualquer*

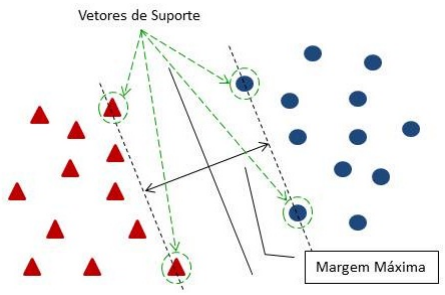

Figura A.6: *Classificador de margem máxima*

os erros no conjunto de treinamento e maximiza a margem, sendo tal problema de otimização denominado de forma primal.

Dentre os algoritmos de SVM, o SMO (*Sequential Minimal Opitmization*) é um dos menores, assim sendo ele é capaz de lidar com grandes bases de dados com um custo computacional moderado.

A.5 Árvores de Decisão

Uma árvore de decisão (ou AD) é um modelo de decisão, uma ferramenta de suporte a decisão, que mostra as possíveis consequências, incluindo as chances de sucesso em cada evento. Uma AD pode mostra qual é o caminho necessário para um determinado evento ocorrer.

Na Figura A.7 é mostrado um exemplo ilustrativo de uma árvore de decisão para o diagnóstico de um paciente para uma determinada doença genética. Na figura, cada elipse é um teste em um atributo para uma dada base de dados de pacientes. Cada retângulo representa uma classe, ou seja, o diagnóstico. Para diagnosticar (classificar) um paciente, basta começar pela raíz, seguindo cada teste até que uma folha seja alcançada. AD é uma estrutura de dados definida recursivamente como:

- um *nó folha* que corresponde a uma classe ou
- um *nó de decisão* que contém um teste sobre algum atributo. Para cada resultado do teste existe uma aresta para uma subárvore. Cada subárvore tem a mesma estrutura que a árvore.

Neste estudo, o algoritmo de indução de árvore de decisão utilizado foi o J48 implementado na biblioteca Weka (**?**)

A.5.1 Construindo uma Árvore de Decisão

O método para a construção de uma árvore de decisão a partir de uma base de treinamento T é relativamente simples. Assumindo que as classes sejam denotadas por $\{C_1, C_2, \ldots, C_k\}$, os seguintes passos devem ser seguidos:

1. T contém um ou mais exemplos, todos pertencentes à mesma classe C_j. Nesse caso, a árvore de decisão para T é um nó folha identificando a classe C_j;

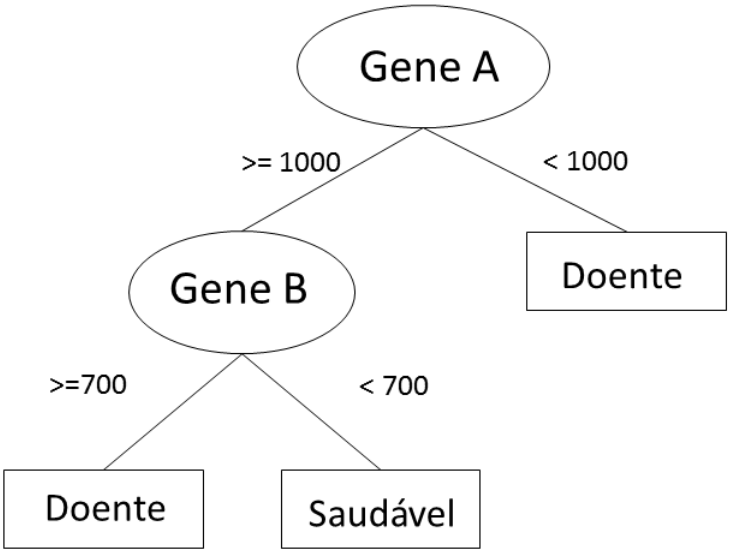

Figura A.7: *Uma árvore de decisão simples para o diagnóstico de uma doença*

2. T não contém exemplos. Novamente, nessa situação, a árvore é uma folha mas a classe associada à folha deve ser determinada a partir de informação além de T. Por exemplo, a classe mais freqüente para o nó pai desse nó pode ser utilizada;

3. T contém exemplos que pertencem a várias classes. Nesse caso á idéia é refinar T em subconjuntos de exemplos que são (ou aparentam ser) conjuntos de exemplos pertencentes a uma única classe. Normalmente, um teste é escolhido, baseado em um único atributo que possui resultados mutuamente exclusivos (na realidade, cada indutor tem sua própria forma de escolher o atributo que será utilizado no teste). Sejam os possíveis resultados do teste denotados por $\{O_1, O_2, ..., O_r\}$. T é então particionado em subconjuntos T_1, T_2, ..., T_r, nos quais cada T_i contém todos os exemplos em T que possuem como resultado daquele teste o valor O_i. A AD para T consiste em um nó interno identificado pelo teste escolhido e uma aresta para cada um dos resultados possíveis;

4. Os passos 1, 2 e 3 são aplicados recursivamente para cada subconjunto de exemplos de treinamento de forma que, em cada nó, as arestas levam para as subárvores construídas a partir do subconjunto de exemplos T_i;

5. Após a construção da AD, a poda pode ser realizada para melhorar a capacidade de generalização da AD.

A.5.2 Escolha do Melhor Atributo para Particionar

A chave para o sucesso de um algoritmo de aprendizado por AD depende do critério utilizado para escolher o atributo que particiona a base de exemplos em cada iteração. Algumas possibilidades para escolher esse atributo são: ganho máximo: seleciona o atributo que possui o maior ganho de informação esperado, isto é, seleciona o atributo que resultará no menor tamanho esperado das subárvores, assumindo que a raíz é o nó atual; índice Gini (**?**); e razão de ganho (**?**). Como exemplo, o algoritmo J48 da biblioteca Weka (**?**) utiliza a razão de ganho para a escolha do atributo.

A.5.3 Poda

Após a construção da árvore de decisão, é possível que o classificador induzido seja muito específico para a base de treinamento. Nesse caso, diz-se que o classificador super-ajustou os dados de treinamento, ou seja, ocorreu um *overfitting*. Como os exemplos de treinamento são apenas uma amostra de todos os exemplos possíveis, é possível adicionar arestas na árvore que melhoram seu desempenho nos dados de treinamento mas que pioram seu desempenho em uma base de teste.

Para tentar solucionar o problema de super-ajuste dos dados, alguns indutores *podam* a AD depois de induzí-la. Esse processo, mostrado na Figura A.8, reduz o número de nós (testes) internos, reduzindo a complexidade da árvore enquanto produz um desempenho melhor que a árvore original. Alguns indutores de AD separam a base de exemplos em uma base de exemplos de treinamento (que é utilizado para construir a AD) e uma base de exemplos de poda, o qual é efetivamente utilizado para realizar o processo de poda.

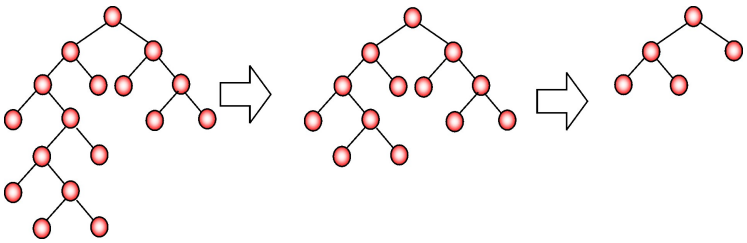

Figura A.8: *Uma árvore grande é induzida de forma a super-ajustar os exemplos e então ela é podada até obter uma árvore menor (mais simples)*

A.5.4 Classificando Novos Exemplos

A AD, após construída, pode ser utilizada para classificar novos exemplos iniciando-se pela raíz da árvore e caminhando através de cada nó de decisão até que uma folha seja encontrada. Quando uma folha é encontrada, a classe do novo exemplo é dada pela classe daquela folha.

Apêndice

B

Bases de Dados Utilizadas nos Experimentos

Na Tabela B.1 é mostrada uma visão resumida das bases de dados, nenhuma delas possui valores ausentes para o atributo classe.

Os números de atributos e exemplos em cada base de dados pode influenciar nos resultados, portanto foi utilizada uma métrica de densidade proposta por (**?**) que divide as bases de dados em 5 bases de baixa densidade (Densidade $\leq 0,75$) e 25 bases de alta densidade (Densidade $\geq 0,75$). Essa densidade é calculada como Density $\triangleq \log_a N$, onde N representa o número de exemplos e a é o número de atributos.

Segue uma breve descrição de cada base de dados. *Breast Cancer*, *Lung Cancer*, *CNS* (Central Nervous System Tumour Outcome), *Colon*, *Lymphoma*, *Leukemia*, *Leukemia nom.*, *WBC* (Wisconsin Breast Cancer), *WDBC* (Wisconsin Diagnostic Breast Cancer), *Lymphography* e *H. Survival* (*H.* representa *Haberman's*) todos estão relacionados a câncer e seus atributos consistem de dados laboratoriais, clínicos ou de expressão gênica. *Leukemia* e *Leukemia nom.* representam os mesmos dados, porém a segunda base teve seus atributos discretizados (**?**). *C. Arrhythmia* (*C.* representa *Cardiac*), *Heart Statlog*, *HD Cleveland*, *HD Hungarian* e *HD Switz.* (*Switz.* é *Switzerland*) todos relacionados com problemas cardiológicos e seus atributos

Tabela B.1: *Resumo das bases utilizadas nos experimentos. ATTR representa o número de atributos em cada base; MISS representa a porcentagem de atributos com valores ausentes, sem considerar o atributo classe. As bases de dados estão em ordem crescente de densidade.*

#	Base	N	c	ATTR	MISS	Densidade
1	CNS	60	2	7129	0.00%	0.46
2	Leukemia	72	2	7129	0.00%	0.48
3	Leukemia nom.	72	2	7129	0.00%	0.48
4	Colon	62	2	2000	0.00%	0.54
5	Lymphoma	96	9	4026	5.09%	0.55
6	Lung Cancer	32	3	56	0.28%	0.86
7	C. Arrhythmia	452	16	279	0.32%	1.09
8	Hepatitis	155	2	19	5.67%	1.71
9	WDBC	569	2	30	0.00%	1.87
10	Dermatology	366	6	34	0.06%	1.67
11	Lymphography	148	4	18	0.00%	1.73
12	Splice Junction	3190	3	60	0.00%	1.97
13	Heart Statlog	270	2	13	0.00%	2.18
14	HD Switz.	123	5	13	17.07%	1.88
15	Sick	3772	2	29	5.54%	2.45
16	P. Patient	90	3	8	0.42%	2.16
17	Hypothyroid	3163	2	25	6.74%	2.50
18	HD Hungarian	294	5	13	20.46%	2.22
19	HD Cleveland	303	5	13	0.18%	2.23
20	Allhypo	3772	4	29	5.54%	2.45
21	Breast Cancer	286	2	9	0.35%	2.98
22	Allhyper	3772	5	29	5.54%	2.41
23	ANN Thyroid	7200	3	21	0.00%	2.92
24	WBC	699	2	9	0.25%	2.98
25	Pima Diabetes	768	2	8	0.00%	3.19
26	Liver Disorders	345	2	6	0.00%	3.26
27	Thyroid 0387	9172	32	29	5.50%	2.71
28	C. Method	1473	3	9	0.00%	3.32
29	Ecoli	482	13	280	1.07%	2.99
30	H. Survival	306	2	3	0.00%	5.21

consistem de dados clínicos e laboratoriais. *Allhyper*, *Allhypo*, *ANN Thyroid*, *Hypothyroid*, *Sick* and *Thyroid 0387* são uma série de bases relacionadas com a condição da tireóide. *Hepatitis* e *Liver Disorders* sobre doenças do figado, onde *C. Method* (*C.* representa *Contraceptive*), *Dermatology*, *Pima Diabetes* (Pima Indians Diabetes) e *P. Patient* (*P.* representa *Postoperative*) são outras bases de dados relacionadas com a condição humana. *Splice Junction* é relacionado com a tarefa de predizer fronteiras entre *exons* e *introns*. *E.Coli* sobre sitio de localização de proteinas. A maioria das bases foram retiradas do repositório da UCI (**?**), *Leukemia* e *Leukemia nom.* foram obtidos do (**?**).

Apêndice

C

Experimento I

O primeiro experimento realizado, denominado Experimento I, utilizou-se a primeira versão do algoritmo na qual a o critério de parada — número de iterações — não foi utilizado. O segundo experimento, denominado Experimento II, utiliza a versão mais nova do algoritmo que inclui o número de iterações como um critério de parada. Este experimento está explicado no Apêndice D.

Na Figura C.1 é mostrada a capacidade de compressão para todas as bases (topo esquerda), 5 de baixa densidade (topo direita) e 25 de alta densidade (centro inferior) bases de dados e na Tabela C.1 os valores médios dos experimentos. É possível observar que m075 teve a pior capacidade de compressão. Isto pode ser explicado pelo fato de que uma vez que é um limiar menor mais atributos serão selecionados. Para bases de dados de alta densidade, a capacidade de compressão média do filtro é inferior a 40% para todas as três configurações. Considerando-se as bases de dados de baixa densidade as duas primeiras configurações, m100 e m095, foram capazes de comprimir os dados em aproximadamente 80% enquanto m075 não chegou a 50%, em média.

Os valores AUC para todas as bases de dados e as configurações do filtro podem ser observados nas Tabelas C.2, C.3, C.4, C.5, C.6. Para analisar o desempenho do filtro em cada base de dados foi utilizado o Teste não-paramétrico de Friedman (**?**) considerando um nível de

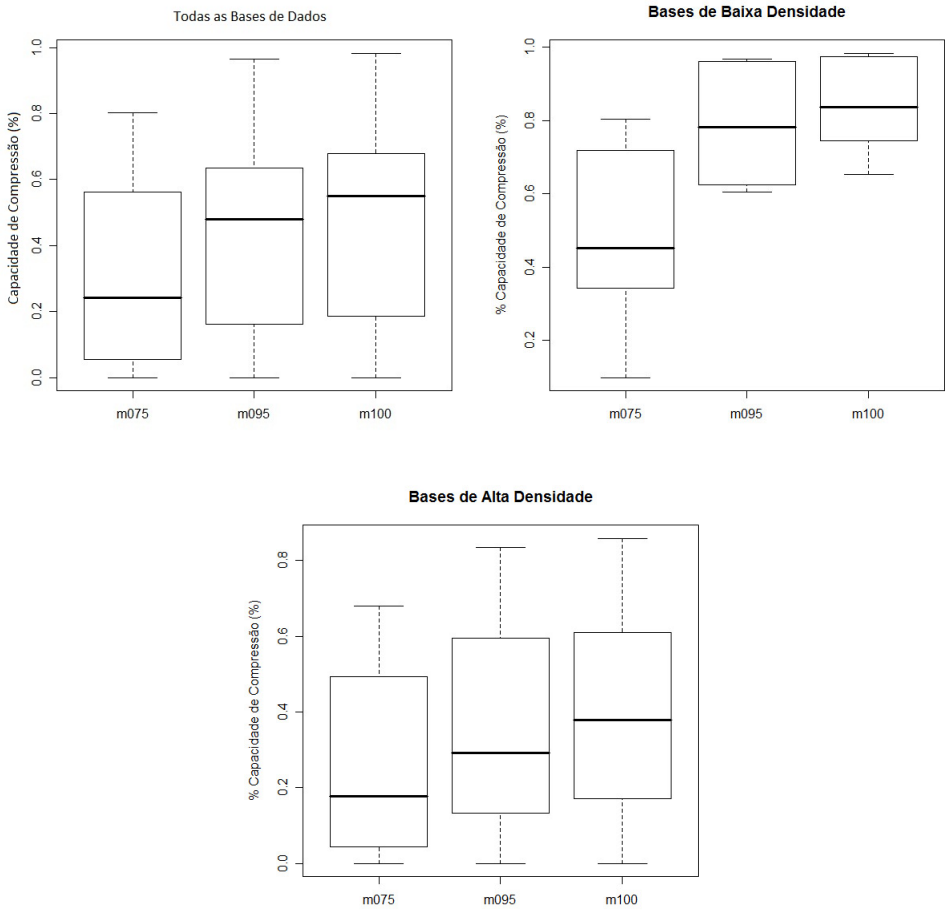

Figura C.1: *Compressão para cada configuração*

Tabela C.1: *Capacidade de Compressão Média do Algoritmo 1*

Base de Dados	m100	m095	m075
Todos	55.07%	48.06%	24.10%
Baixa-Densidade	83.82%	78.73%	48.24%
Alta-Densidade	38.98%	34.10%	25.22%

significância de 5%; a hipotese nula assume que todos os classificadores têm o mesmo desempenho. Se a hipotese nula é rejeitada, o teste Benjamini-Hochberg *post-hoc* (**?**) é usado para detectar qualquer diferença estatisticamente significante entre os classificadores.

Analisando o *Rank* Médio nas Tabelas C.2, C.3, C.4, C.5, C.6 é possível observar que m075, para todos os paradigmas e todas as bases de dados, obteve o pior desempenho. Apenas SMO teve um desempenho melhor do que todas as configurações do filtro. Sobre as bases de baixa densidade J48 também teve um melhor desempenho do que todas as três configurações. Nas bases de alta densidade para todos os indutores, exceto SMO, m100 e m095 foram melhores, exceto para J48, onde m095 obteve um desempenho um pouco melhor do que m100. Dado que nenhum dos classificadores obteve o mesmo desempenho, a hipótese nula é rejeitada e um teste *post-hoc* é necessário.

O resultado do teste *post-hoc* pode ser encontrado na Tabela C.7 considerando as três configurações. Na tabela o símbolo $\triangle$ (▲) significa que o classificador na linha, usando todos os atributos, é melhor (significativamente) que o mesmo classificador, utilizando apenas os atributos selecionados pelo filtro, na coluna; o símbolo $\triangledown$ (▼) significa que o classificador na linha, usando todos os atributos, é pior (significativamente) que o mesmo classificador, utilizando apenas os atributos selecionados pelo filtro, na coluna.

Observando a Tabela C.7 é possível confirmar os resultados já mencionados nas Tabelas C.2, C.3, C.4, C.5, C.6. A configuração m075 é sempre pior que a base original; para SMO e J48 ela é significativamente pior. SMO é melhor que qualquer configuração do filtro proposta. IBk3, NB e PART o filtro obteve um resultado melhor, porém não significantivamente, para todas as bases e também considerando separadamente as bases de alta e baixa densidade. Para as bases de baixa densidade J48 obteve um resultado melhor que as três configurações. De forma geral, analisando todas as bases de dados o Algoritmo 1 obteve um desempenho melhor em quatro dos cinco paradigmas testados, tendo um desempenho pior somente que SMO. Trabalhos futuros podem ser feitos para tentar analisar esse comportamento.

Tabela C.2: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o J48*

Base de Dados	J48	m100	m095	m075
CNS	0.50	0.49	0.49	0.46
Leukemia	0.75	0.76	0.76	0.70
Leukemia nom.	0.92	0.92	0.92	0.83
Colon	0.80	0.73	0.81	0.73
Lymphoma	0.90	0.90	0.90	0.80
Lung Cancer	0.68	0.63	0.64	0.63
C. Arrhythmia	0.72	0.74	0.73	0.65
Hepatitis	0.70	0.71	0.72	0.63
WDBC	0.93	0.94	0.94	0.84
Dermatology	0.97	0.98	0.98	0.88
Lymphography	0.79	0.79	0.79	0.72
Splice Junction	0.96	0.96	0.96	0.87
Heart Statlog	0.76	0.74	0.75	0.69
HD Switz.	0.55	0.55	0.55	0.49
Sick	0.95	0.95	0.95	0.86
P. Patient	0.49	0.49	0.49	0.44
Hypothyroid	0.95	0.95	0.95	0.86
HD Hungarian	0.77	0.73	0.73	0.66
HD Cleveland	0.80	0.82	0.80	0.73
Allhypo	1.00	1.00	1.00	0.90
Breast Cancer	0.63	0.62	0.63	0.56
Allhyper	0.90	0.89	0.89	0.81
ANN Thyroid	0.99	0.99	0.99	0.89
WBC	0.95	0.96	0.95	0.86
Pima Diabetes	0.75	0.75	0.75	0.67
Liver Disorders	0.67	0.67	0.67	0.60
Thyroid 0387	0.98	0.98	0.98	0.89
C. Method	0.66	0.66	0.66	0.60
Ecoli	0.92	0.93	0.93	0.83
H. Survival	0.58	0.58	0.58	0.51
Todos				
Média	0.80	0.79	0.80	0.72
Mediana	0.79	0.78	0.79	0.72
Rank Médio	2.13	2.07	1.85	3.95
Baixa-Densidade				
Média	0.77	0.76	0.78	0.70
Mediana	0.80	0.76	0.81	0.73
Rank Médio	1.66	2.66	1.91	3.75
Alta-Densidade				
Média	0.80	0.80	0.80	0.72
Mediana	0.79	0.79	0.79	0.72
Rank Médio	2.25	1.92	1.83	4.00

Tabela C.3: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o IBk3*

Base de Dados	IBk3	m100	m095	m075
CNS	0.56	0.56	0.56	0.53
Leukemia	0.85	0.88	0.88	0.78
Leukemia nom.	0.99	0.99	0.99	0.89
Colon	0.83	0.87	0.87	0.77
Lymphoma	0.92	0.92	0.92	0.83
Lung Cancer	0.68	0.78	0.78	0.68
C. Arrhythmia	0.66	0.67	0.67	0.59
Hepatitis	0.79	0.76	0.76	0.68
WDBC	0.98	0.98	0.98	0.88
Dermatology	0.99	0.99	0.99	0.89
Lymphography	0.89	0.88	0.88	0.80
Splice Junction	0.94	0.95	0.95	0.85
Heart Statlog	0.83	0.83	0.83	0.75
HD Switz.	0.49	0.54	0.54	0.49
Sick	0.88	0.92	0.92	0.82
P. Patient	0.31	0.50	0.50	0.44
Hypothyroid	0.85	0.87	0.87	0.79
HD Hungarian	0.87	0.86	0.86	0.77
HD Cleveland	0.85	0.84	0.84	0.77
Allhypo	0.74	0.76	0.76	0.69
Breast Cancer	0.66	0.64	0.64	0.58
Allhyper	0.85	0.90	0.90	0.81
ANN Thyroid	0.74	0.78	0.78	0.70
WBC	0.99	0.99	0.99	0.89
Pima Diabetes	0.74	0.74	0.74	0.67
Liver Disorders	0.64	0.64	0.64	0.58
Thyroid 0387	0.84	0.85	0.85	0.76
C. Method	0.62	0.62	0.62	0.56
Ecoli	0.94	0.94	0.94	0.84
H. Survival	0.63	0.63	0.63	0.58
Todos				
Média	0.78	0.80	0.80	0.72
Mediana	0.84	0.84	0.84	0.76
Rank Médio	2.33	1.88	1.88	3.90
Baixa-Densidade				
Média	0.81	0.83	0.83	0.75
Mediana	0.84	0.84	0.84	0.76
Rank Médio	2.83	1.67	1.67	3.83
Alta-Densidade				
Média	0.78	0.80	0.80	0.72
Mediana	0.84	0.83	0.83	0.75
Rank Médio	2.21	1.94	1.94	3.92

Tabela C.4: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o NB*

Base de Dados	NB	m100	m095	m075
CNS	0.58	0.59	0.59	0.53
Leukemia	0.98	1.00	1.00	0.90
Leukemia nom.	0.96	0.99	0.99	0.90
Colon	0.61	0.63	0.63	0.54
Lymphoma	0.76	0.77	0.77	0.68
Lung Cancer	0.71	0.71	0.71	0.61
C. Arrhythmia	0.81	0.81	0.81	0.72
Hepatitis	0.86	0.86	0.86	0.79
WDBC	0.98	0.98	0.98	0.88
Dermatology	1.00	1.00	1.00	0.90
Lymphography	0.91	0.91	0.91	0.82
Splice Junction	0.99	0.99	0.99	0.89
Heart Statlog	0.90	0.90	0.90	0.82
HD Switz.	0.53	0.54	0.54	0.47
Sick	0.93	0.93	0.93	0.83
P. Patient	0.39	0.46	0.46	0.43
Hypothyroid	0.97	0.98	0.98	0.89
HD Hungarian	0.90	0.89	0.89	0.80
HD Cleveland	0.90	0.90	0.90	0.82
Allhypo	0.93	0.94	0.94	0.84
Breast Cancer	0.72	0.71	0.71	0.63
Allhyper	0.97	0.97	0.97	0.88
ANN Thyroid	0.93	0.93	0.93	0.84
WBC	0.99	0.99	0.99	0.89
Pima Diabetes	0.82	0.82	0.82	0.73
Liver Disorders	0.65	0.65	0.65	0.58
Thyroid 0387	0.93	0.93	0.93	0.83
C. Method	0.69	0.69	0.69	0.63
Ecoli	0.97	0.97	0.97	0.87
H. Survival	0.67	0.67	0.67	0.60
Todos				
Média	0.83	0.84	0.84	0.75
Mediana	0.90	0.90	0.90	0.82
Rank Médio	2.40	1.82	1.82	3.97
Baixa-Densidade				
Média	0.77	0.78	0.78	0.69
Mediana	0.74	0.74	0.74	0.65
Rank Médio	2.83	1.58	1.58	4.00
Alta-Densidade				
Média	0.85	0.85	0.85	0.77
Mediana	0.91	0.91	0.91	0.82
Rank Médio	2.29	1.88	1.88	3.96

Tabela C.5: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o PART*

Base de Dados	J48	m100	m095	m075
CNS	0.00	0.59	0.59	0.53
Leukemia	0.75	0.76	0.76	0.70
Leukemia nom.	0.88	0.86	0.86	0.78
Colon	0.79	0.83	0.83	0.75
Lymphoma	0.81	0.82	0.82	0.73
Lung Cancer	0.69	0.69	0.69	0.61
C. Arrhythmia	0.74	0.75	0.75	0.67
Hepatitis	0.83	0.80	0.80	0.71
WDBC	0.94	0.94	0.94	0.84
Dermatology	0.98	0.98	0.98	0.88
Lymphography	0.79	0.79	0.79	0.73
Splice Junction	0.96	0.96	0.96	0.87
Heart Statlog	0.74	0.75	0.75	0.66
HD Switz.	0.54	0.56	0.56	0.50
Sick	0.96	0.97	0.97	0.87
P. Patient	0.46	0.46	0.46	0.41
Hypothyroid	0.96	0.95	0.95	0.85
HD Hungarian	0.87	0.84	0.84	0.75
HD Cleveland	0.81	0.84	0.84	0.76
Allhypo	1.00	1.00	1.00	0.90
Breast Cancer	0.58	0.58	0.58	0.51
Allhyper	0.94	0.95	0.95	0.87
ANN Thyroid	0.99	1.00	1.00	0.90
WBC	0.97	0.97	0.97	0.88
Pima Diabetes	0.79	0.79	0.79	0.71
Liver Disorders	0.67	0.67	0.67	0.60
Thyroid 0387	0.98	0.98	0.98	0.88
C. Method	0.64	0.65	0.65	0.59
Ecoli	0.92	0.92	0.92	0.83
H. Survival	0.55	0.55	0.55	0.49
Todos				
Média	0.78	0.81	0.81	0.73
Mediana	0.81	0.82	0.82	0.74
Rank Médio	2.43	1.80	1.80	3.97
Baixa-Densidade				
Média	0.65	0.76	0.76	0.68
Mediana	0.77	0.79	0.79	0.72
Rank Médio	2.67	1.75	1.75	3.83
Alta-Densidade				
Média	0.82	0.82	0.82	0.74
Mediana	0.85	0.84	0.84	0.76
Rank Médio	2.38	1.81	1.81	4.00

Tabela C.6: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o SMO*

Base de Dados	J48	m100	m095	m075
CNS	0.63	0.63	0.63	0.56
Leukemia	0.98	0.98	0.98	0.90
Leukemia nom.	0.97	0.97	0.97	0.89
Colon	0.84	0.82	0.82	0.74
Lymphoma	0.98	0.98	0.98	0.88
Lung Cancer	0.63	0.63	0.63	0.55
C. Arrhythmia	0.79	0.79	0.79	0.70
Hepatitis	0.75	0.70	0.70	0.62
WDBC	0.97	0.97	0.97	0.88
Dermatology	0.98	0.98	0.98	0.89
Lymphography	0.87	0.85	0.85	0.78
Splice Junction	0.96	0.96	0.96	0.87
Heart Statlog	0.84	0.84	0.84	0.75
HD Switz.	0.57	0.56	0.56	0.50
Sick	0.50	0.50	0.50	0.45
P. Patient	0.47	0.49	0.49	0.44
Hypothyroid	0.75	0.74	0.74	0.67
HD Hungarian	0.80	0.78	0.78	0.70
HD Cleveland	0.84	0.84	0.84	0.75
Allhypo	0.59	0.59	0.59	0.54
Breast Cancer	0.59	0.57	0.57	0.51
Allhyper	0.66	0.63	0.63	0.57
ANN Thyroid	0.59	0.59	0.59	0.53
WBC	0.97	0.97	0.97	0.87
Pima Diabetes	0.72	0.72	0.72	0.65
Liver Disorders	0.50	0.50	0.50	0.45
Thyroid 0387	0.98	0.98	0.98	0.88
C. Method	0.63	0.63	0.63	0.57
Ecoli	0.95	0.95	0.95	0.85
H. Survival	0.51	0.50	0.50	0.45
Todos				
Média	0.76	0.76	0.76	0.68
Mediana	0.77	0.76	0.76	0.68
Rank Médio	1.73	2.13	2.13	4.00
Baixa-Densidade				
Média	0.84	0.84	0.84	0.76
Mediana	0.91	0.89	0.89	0.81
Rank Médio	1.83	2.08	2.08	4.00
Alta-Densidade				
Média	0.74	0.74	0.74	0.66
Mediana	0.75	0.73	0.73	0.66
Rank Médio	1.71	2.15	2.15	4.00

Tabela C.7: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus indutores*

	M100 (TODOS/ALTA/BAIXA)	M095 (TODOS/ALTA/BAIXA)	M075 (TODOS/ALTA/BAIXA)
J48	▽/▽/△	▽/▽/△	▲/▲/▲
IBK3	▽/▽/▽	▽/▽/▽	▲/▲/△
NB	▽/▽/▽	▽/▽/▽	▲/▲/△
PART	▽/▽/▽	▽/▽/▽	▲/▲/△
SMO	△/△/△	△/△/△	▲/▲/▲

Apêndice

D

Experimento II

Utilizando o algoritmo na sua segunda versão, incluindo o novo critério de parada, que considera o número de iterações, foram realizados os mesmos testes que o experimento anterior, para avaliar se a implementação do critério de parada, que reduziu o custo computacional, modificou o desempenho e a capacidade de compressão do Algoritmo 1

Na Figura D.1 é mostrada a capacidade de compressão para todas as bases de dados (topo esquerda), para as bases com baixa densidade (topo direita) e para bases com alta densisdade (centro inferior) e na Tabela D.6 os valores médios dos experimentos referentes a figura. É possível observar que, assim como no experimento anterior o m075 teve a pior capacidade de compressão. Isto pode ser explicado pelo fato de que uma vez que é um limiar menor mais atributos serão selecionados. Para bases de dados de alta densidade, a capacidade de compressão média do filtro é inferior a 40% para todas as três configurações. Considerando-se as bases de dados de baixa densidade todas as configurações obtiveram compressão acima de 98%. Essa diferença em relação ao experimento anterior onde a média de compressão estava perto dos 80% pode ser explicada pelo critério de parada, pois este faz com que o algoritmo pare antes de avaliar todos os atributos, como as bases de baixa densidade são as que possuem um grande número de atributos, boa parte deles não são observados pelo algoritmo. Em relação aos dados de alta densidade não houve muita diferença, dado que estas bases possuem um

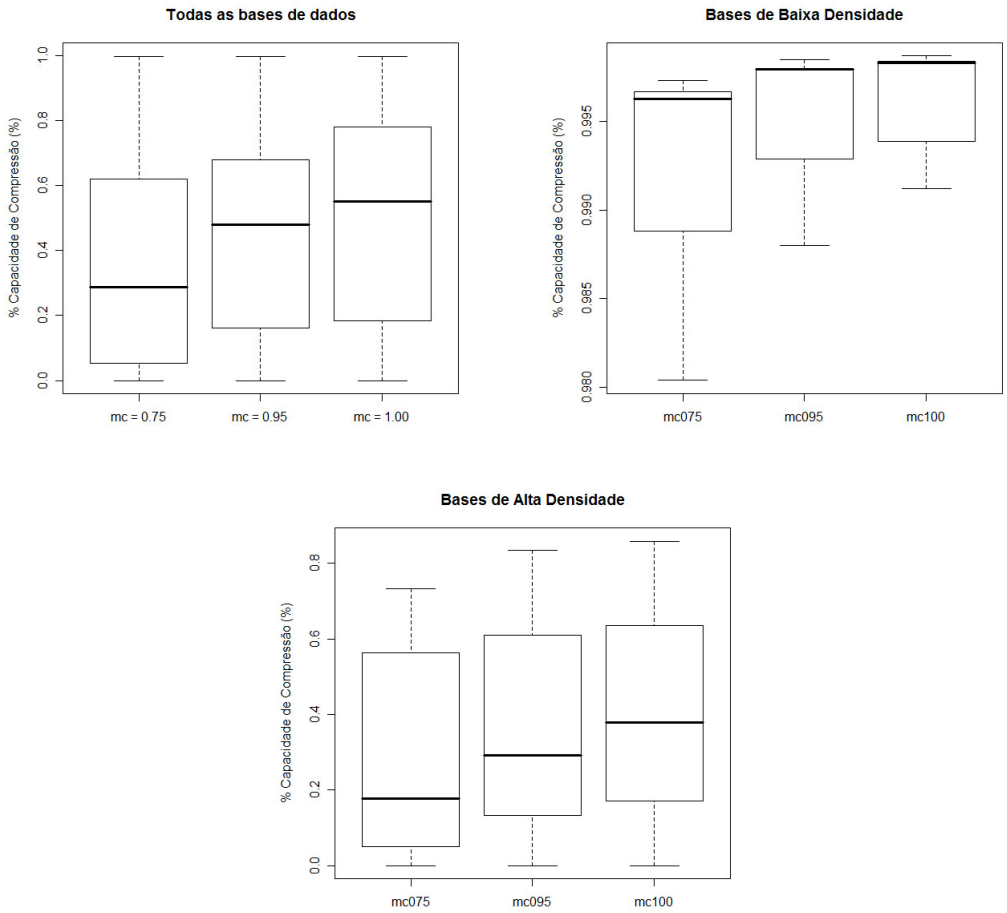

Figura D.1: *Compressão para cada configuração*

número pequeno de atributos e geralmente não atingem o novo critério de parada.

Outra diferença entre o Experimento 1 e o Experimento 2 foi a utlização de uma quarta métrica de avaliação, além de m100, m095 e m075 a métrica *Unique Tree* ou UT foi adicionada, essa métrica significa que o algoritmo fez apenas 1 iteração, assim sendo gerou apenas uma árvore. Como podemos ver, em termos de capacidade de compressão, mostrado na Tabela D.6 a métrica UT obteve ótimos resultados, porém a avaliação dela na analise estatística provou que apesar de comprimir bem os dados essa métrica não possui boa precisão.

Na Tabela D.1 são expostos os resultados do segundo experimento para o algoritmo J48. Podemos observar que considerando todos as bases de dados ou somente as de alta densidade a metodologia UT obteve o melhor Rank Médio. Já analisando apenas as bases de dados com baixa densidade o J48 obteve o melhor Rank Médio, porém a metodologia M095 obteve a melhor Média.

Na Tabela D.2 são mostrados os resultados do segundo experimento com o algoritmo IBk3. Analisando todas as bases de dados as metodologias m100 e m095 obtiveram os melhores

Tabela D.1: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o J48*

Base de Dados	J48	m100	m095	UT
CNS	0.50	0.63	0.63	0.60
Leukemia	0.75	0.82	0.82	0.90
Leukemia nom.	0.92	0.90	0.90	0.83
Colon	0.80	0.75	0.75	0.73
Lymphoma	0.90	0.85	0.85	0.83
Lung Cancer	0.68	0.81	0.81	0.69
C. Arrhythmia	0.72	0.73	0.73	0.73
Hepatitis	0.70	0.69	0.69	0.98
WDBC	0.93	0.94	0.94	0.70
Dermatology	0.97	0.97	0.97	0.79
Lymphography	0.79	0.76	0.76	0.94
Splice Junction	0.96	0.96	0.96	0.56
Heart Statlog	0.76	0.75	0.75	0.96
HD Switz.	0.55	0.55	0.55	0.49
Sick	0.95	0.95	0.95	0.75
P. Patient	0.49	0.49	0.49	0.73
Hypothyroid	0.95	0.95	0.95	0.83
HD Hungarian	0.77	0.73	0.73	0.89
HD Cleveland	0.80	0.79	0.79	1.00
Allhypo	1.00	1.00	1.00	0.95
Breast Cancer	0.63	0.63	0.63	0.96
Allhyper	0.90	0.90	0.90	0.63
ANN Thyroid	0.99	0.99	0.99	0.98
WBC	0.95	0.95	0.95	0.99
Pima Diabetes	0.75	0.75	0.75	0.97
Liver Disorders	0.67	0.67	0.67	0.93
Thyroid 0387	0.98	0.98	0.98	0.75
C. Method	0.66	0.66	0.66	0.67
Ecoli	0.92	0.92	0.92	0.66
H. Survival	0.58	0.58	0.58	0.58
Todos				
Média	0.80	0.79	0.80	0.80
Mediana	0.79	0.78	0.79	0.81
Rank Médio	2.57	2.72	2.72	2.00
Baixa-Densidade				
Média	0.77	0.76	0.78	0.76
Mediana	0.80	0.76	0.81	0.78
Rank Médio	2.20	2.30	2.30	3.20
Alta-Densidade				
Média	0.80	0.80	0.80	0.80
Mediana	0.79	0.79	0.79	0.79
Rank Médio	2.64	2.80	2.80	1.76

Tabela D.2: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o IBk3*

Base de Dados	IBk3	m100	m095	UT
CNS	0.56	0.60	0.60	0.63
Leukemia	0.85	0.92	0.92	0.91
Leukemia nom.	0.99	0.98	0.98	0.88
Colon	0.83	0.85	0.85	0.70
Lymphoma	0.92	0.95	0.95	0.90
Lung Cancer	0.68	0.69	0.69	0.58
C. Arrhythmia	0.66	0.67	0.67	0.73
Hepatitis	0.79	0.76	0.76	0.98
WDBC	0.98	0.98	0.98	0.78
Dermatology	0.99	0.99	0.99	0.87
Lymphography	0.89	0.88	0.88	0.97
Splice Junction	0.94	0.95	0.95	0.53
Heart Statlog	0.83	0.83	0.83	0.96
HD Switz.	0.49	0.54	0.54	0.50
Sick	0.88	0.92	0.92	0.84
P. Patient	0.31	0.50	0.50	0.85
Hypothyroid	0.85	0.87	0.87	0.85
HD Hungarian	0.87	0.86	0.86	0.88
HD Cleveland	0.85	0.84	0.84	0.78
Allhypo	0.74	0.76	0.76	0.93
Breast Cancer	0.66	0.64	0.64	0.95
Allhyper	0.85	0.90	0.90	0.68
ANN Thyroid	0.74	0.78	0.78	0.85
WBC	0.99	0.99	0.99	0.82
Pima Diabetes	0.74	0.74	0.74	0.99
Liver Disorders	0.64	0.64	0.64	0.94
Thyroid 0387	0.84	0.85	0.85	0.74
C. Method	0.62	0.62	0.62	0.64
Ecoli	0.94	0.94	0.94	0.62
H. Survival	0.63	0.63	0.63	0.63
Todos				
Média	0.78	0.80	0.80	0.80
Mediana	0.84	0.84	0.84	0.84
Rank Médio	2.87	2.32	2.32	2.50
Baixa-Densidade				
Média	0,83	0.86	0.86	0.78
Mediana	0,85	0.92	0.92	0.79
Rank Médio	3.00	1.90	1.90	3.20
Alta-Densidade				
Média	0.78	0.79	0.79	0.80
Mediana	0.83	0.83	0.83	0.84
Rank Médio	2.84	2.40	2.40	2.36

Rank Médio, porém com a mesma média do UT. Observando somente as bases de dados de alta densidade o UT obteve o melhor *Rank* Médio assim como a melhor média, nas bases de baixa densidade novamente m100 e m095 aparecem com o menor *Rank* Médio, dessa vez com média superior a UT.

Na Tabela D.3 os resultados utilizando o algoritmo *Naive Bayes* são expostos, estes não diferem muito dos obtidos pelo IBk3 onde as metodologias m100 e m095 obtiveram um Rank Médio menor analisando todas as bases de dados ou analisando somente as bases de dados de baixa densidade.

Analisando os resultados para o algoritmo PART mostrados na Tabela D.4 é possivel observar que o algoritmo sozinho obteve os melhores resultados tanto para todas as base de dados e para as de alta densidade. A metodologia UT ficou com o maior Rank Médio para todos os conjuntos. Analisando somente as de baixa densidade as metodologias m100 e m095 obtiveram um *Rank* Médio menor.

O último algoritmo analisado foi o SMO, cujo resultados aparecem na Tabela D.5. Para todos os conjuntos de base de dados as metodologias m100 e m095 obtiveram o menor *Rank* Médio, enquanto a UT ficou com o maior também para todos os conjuntos.

Na Tabela D.6 é possível ver a capacidade de compressão média para as 4 metodologias propostas utlizando agora o novo critério de parada. Como é possível observar a melhor capacidade de compressão ficou com a UT comprimindo 53.07% em segundo aparece a mc100 comprimindo 99.45% dos dados. De modo geral é possível percerber que para grandes bases de dados, ou seja, as de baixa densidade, a metodologia proposta consegue comprimir mais de 99% ou seja em uma base de dados de 1000 atributos a metodologia seleciona apenas 10. Já para bases com poucos atributos é o resultado não chegou a 40% exceto na UT.

Ao analisarmos a Tabela D.7 é possível observar que as metodologias m100 e m095 não possuem nenhum tipo de diferença significativa em relação aos algoritmos sem passarem pelo filtro proposto. Isso mostra que o filtro proposto faz uma seleção de atributos reduzindo o tamanho da base de dados porém sem nenhuma perde significativa de informação. Já a metodologia UT apresentou um comportamento variável. Para o J48 ela conseguiu obter, para o conjunto de bases de dados de alta densidade uma melhora significativa mesmo com a redução da base. Isso pode ter sido causado dado que a metodologia UT e o J48 deveriam ser a mesma árvore, porém como o filtro proposto faz um bootstrap antes de passar pela metodologia essa nova base de dados fez com que atributos mais relevantes fossem selecionados.

Tabela D.3: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o NaiveBayes*

Base de Dados	IBk3	m100	m095	UT
CNS	0.58	0.55	0.55	0.63
Leukemia	0.98	1.00	1.00	0.90
Leukemia nom.	0.96	1.00	1.00	0.88
Colon	0.61	0.87	0.87	0.75
Lymphoma	0.76	0.91	0.91	0.91
Lung Cancer	0.71	0.74	0.74	0.61
C. Arrhythmia	0.81	0.81	0.81	0.83
Hepatitis	0.86	0.86	0.86	0.99
WDBC	0.98	0.98	0.98	0.89
Dermatology	1.00	1.00	1.00	0.89
Lymphography	0.91	0.91	0.91	0.98
Splice Junction	0.99	0.99	0.99	0.56
Heart Statlog	0.90	0.90	0.90	0.99
HD Switz.	0.53	0.54	0.54	0.46
Sick	0.93	0.93	0.93	0.90
P. Patient	0.39	0.46	0.46	0.89
Hypothyroid	0.97	0.98	0.98	0.90
HD Hungarian	0.90	0.89	0.89	0.97
HD Cleveland	0.90	0.90	0.90	0.94
Allhypo	0.93	0.94	0.94	0.92
Breast Cancer	0.72	0.71	0.71	0.99
Allhyper	0.97	0.97	0.97	0.71
ANN Thyroid	0.93	0.93	0.93	0.93
WBC	0.99	0.99	0.99	0.93
Pima Diabetes	0.82	0.82	0.82	0.99
Liver Disorders	0.65	0.65	0.65	0.97
Thyroid 0387	0.93	0.93	0.93	0.82
C. Method	0.69	0.69	0.69	0.65
Ecoli	0.97	0.97	0.97	0.69
H. Survival	0.67	0.67	0.67	0.67
Todos				
Média	0.78	0.80	0.80	0.84
Mediana	0.84	0.84	0.84	0.89
Rank Médio	2.77	2.47	2.47	2.30
Baixa-Densidade				
Média	0.78	0.86	0.86	0.79
Mediana	0.76	0.91	0.91	0.81
Rank Médio	3.20	2.10	2.10	2.60
Alta-Densidade				
Média	0.84	0.85	0.85	0.84
Mediana	0.90	0.90	0.90	0.90
Rank Médio	2.68	2.54	2.54	2.24

Tabela D.4: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o PART*

Base de Dados	IBk3	m100	m095	UT
CNS	0.63	0.48	0.48	0.50
Leukemia	0.00	1.00	1.00	0.67
Leukemia nom.	0.97	0.97	0.97	0.88
Colon	0.84	0.82	0.82	0.63
Lymphoma	0.98	0.97	0.97	0.77
Lung Cancer	0.63	0.67	0.67	0.57
C. Arrhythmia	0.79	0.79	0.79	0.74
Hepatitis	0.75	0.70	0.70	0.99
WDBC	0.97	0.97	0.97	0.69
Dermatology	0.98	0.98	0.98	0.82
Lymphography	0.87	0.85	0.85	0.95
Splice Junction	0.96	0.96	0.96	0.54
Heart Statlog	0.84	0.84	0.84	0.97
HD Switz.	0.57	0.56	0.56	0.49
Sick	0.50	0.50	0.50	0.83
P. Patient	0.47	0.49	0.49	0.78
Hypothyroid	0.75	0.74	0.74	0.84
HD Hungarian	0.80	0.78	0.78	0.63
HD Cleveland	0.84	0.84	0.84	0.59
Allhypo	0.59	0.59	0.59	0.50
Breast Cancer	0.59	0.57	0.57	0.66
Allhyper	0.66	0.63	0.63	0.56
ANN Thyroid	0.59	0.59	0.59	0.68
WBC	0.97	0.97	0.97	0.58
Pima Diabetes	0.72	0.72	0.72	0.96
Liver Disorders	0.50	0.50	0.50	0.95
Thyroid 0387	0.68	0.68	0.68	0.72
C. Method	0.63	0.63	0.63	0.50
Ecoli	0.95	0.95	0.95	0.63
H. Survival	0.51	0.50	0.50	0.50
Todos				
Média	0.72	0.74	0.74	0.70
Mediana	0.73	0.73	0.73	0.68
Rank Médio	1.95	2.35	2.35	3.35
Baixa-Densidade				
Média	0.69	0.85	0.85	0.67
Mediana	0.79	0.82	0.82	0.65
Rank Médio	2.80	2.40	2.40	3.40
Alta-Densidade				
Média	0.72	0.72	0.72	0.71
Mediana	0.72	0.70	0.70	0.68
Rank Médio	1.95	2.34	2.34	3.34

Tabela D.5: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o SMO*

Base de Dados	IBk3	m100	m095	UT
CNS	0.51	0.55	0.55	0.63
Leukemia	0.75	0.82	0.82	0.90
Leukemia nom.	0.88	0.88	0.88	0.82
Colon	0.79	0.77	0.77	0.71
Lymphoma	0.81	0.84	0.84	0.78
Lung Cancer	0.69	0.71	0.71	0.63
C. Arrhythmia	0.74	0.75	0.75	0.73
Hepatitis	0.83	0.80	0.80	0.98
WDBC	0.94	0.94	0.94	0.79
Dermatology	0.98	0.98	0.98	0.81
Lymphography	0.79	0.79	0.79	0.95
Splice Junction	0.96	0.96	0.96	0.55
Heart Statlog	0.74	0.75	0.75	0.96
HD Switz.	0.54	0.56	0.56	0.46
Sick	0.96	0.97	0.97	0.83
P. Patient	0.46	0.46	0.46	0.85
Hypothyroid	0.96	0.95	0.95	0.80
HD Hungarian	0.87	0.84	0.84	0.95
HD Cleveland	0.81	0.84	0.84	0.99
Allhypo	1.00	1.00	1.00	0.97
Breast Cancer	0.58	0.58	0.58	0.95
Allhyper	0.94	0.95	0.95	0.59
ANN Thyroid	0.99	1.00	1.00	0.98
WBC	0.97	0.97	0.97	1.00
Pima Diabetes	0.79	0.79	0.79	0.98
Liver Disorders	0.67	0.67	0.67	0.92
Thyroid 0387	0.98	0.98	0.98	0.79
C. Method	0.64	0.65	0.65	0.67
Ecoli	0.92	0.92	0.92	0.65
H. Survival	0.55	0.55	0.55	0.55
Todos				
Média	0.83	0.85	0.85	0.80
Mediana	0.90	0.91	0.91	0.81
Rank Médio	2.70	2.27	2.27	2.77
Baixa-Densidade				
Média	0.75	0.77	0.77	0.76
Mediana	0.79	0.82	0.82	0.76
Rank Médio	2.80	2.20	2.20	2.80
Alta-Densidade				
Média	0.81	0.81	0.81	0.81
Mediana	0.83	0.84	0.84	0.83
Rank Médio	2.68	2.28	2.28	2.76

Tabela D.6: *Capacidade de Compressão Média do Algoritmo 1 com os 3 critérios de parada*

Base de Dados	mc100	mc095	mc075	UT
Todos	49.77%	45.83%	39.06%	53,07%
Baixa-Densidade	99.45%	99.30%	98.85%	99,89%
Alta-Densidade	39.81%	35.09%	27.03%	43,69%

Tabela D.7: *Benjamini-Hochberg post-hoc Test - Algoritmo 1 + indutor versus indutores*

	M100 (TODOS/ALTA/BAIXA)	M095 (TODOS/ALTA/BAIXA)	UT (TODOS/ALTA/BAIXA)
J48	△/△/△	△/△/△	▽/▼/△
IBK3	▽/▽/▽	▽/▽/▽	▽/▽/△
NB	▽/▽/▽	▽/▽/▽	▽/▽/▽
PART	▽/▽/▽	▽/▽/▽	△/△/*o*
SMO	△/△/△	△/△/△	▲/▲/△

Apêndice

E

Experimento III

A primeira avaliação foi feita visando comparar o tempo de seleção de nosso algoritmos com alguns da literatura. O cálculo de tempo foi feito utilizando o algoritmo de AM conhecido como Zero Rule, esse algoritmo apenas calcula a média para bases com classes com valores contínuos ou a moda para bases com classes nominais, pois este praticamente não utiliza tempo computacional.

Como podemos observar pelos dados mostrados na Tabela E.1 o seletor *Gain Ratio* apresentou o menor tempo de seleção dentre todos os seletores. Em segundo lugar ficou o*CFS Subset Eval*, a metodologia utilizando uma única árvore (UT) em terceiro. O *Relief F* apresentou o pior tempo. O *CFS Subset Eval* apresentou o pior tempo para grandes bases, baixa densidade, isso se deu devido ao fato que ele trabalha com subconjuntos, logo quanto menor a base menos subconjuntos podem ser gerados. A metodologia proposta obteve o quarto tempo na média com média de 6400 milisegundos por base. Podemos considerar esse tempo como um bom tempo para analisar uma grande bases de dados, dado que o processamento em média, de uma grande base, dura apenas 6 segundos e consegue reduzir mais de 99% dos dados. A metodologia proposta fica apenas alguns segundos de um seletor rápido como o *Gain Ratio* que processou as grandes bases, em média, em 3 segundos.

Na Tabela E.2 é mostrada a porcentagem de atributos selecionados. É possível observar que o

Tabela E.1: *Valor do Tempo Computacional gasto por cada Seletor escala* log_10

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	2.93	1.86	3.74	3.66	3.67	3.33
Leukemia	0.15	-0.15	1.80	0.99	1.01	0.90
Leukemia nom.	0.80	0.65	3.21	1.97	1.96	1.96
Colon	1.19	0.32	2.58	1.58	1.57	1.26
Lymphoma	0.48	0.38	1.95	1.31	1.31	1.26
Lung Cancer	-0.15	-0.30	1.54	0.89	0.89	0.86
C. Arrhythmia	0.64	0.04	1.60	1.18	1.18	1.05
Hepatitis	0.28	0.20	1.88	1.30	1.45	1.26
WDBC	1.10	0.15	0.90	1.21	1.20	0.78
Dermatology	0.48	0.00	1.58	1.11	1.14	1.03
Lymphography	0.75	0.74	2.64	1.67	1.67	1.65
Splice Junction	-0.30	-0.40	0.88	0.65	0.65	0.59
Heart Statlog	0.54	0.32	1.93	1.34	1.34	1.30
HD Switz.	0.80	0.67	2.61	1.49	1.44	1.38
Sick	1.48	1.30	2.92	2.12	2.15	1.94
P. Patient	5.80	2.51	3.44	3.91	3.90	3.18
Hypothyroid	4.41	1.96	2.92	3.26	3.26	2.62
HD Hungarian	6.33	2.74	3.71	4.12	4.12	3.39
HD Cleveland	6.08	2.61	3.60	3.83	3.82	3.06
Allhypo	5.44	2.14	3.58	3.38	3.39	2.45
Breast Cancer	0.59	0.28	2.02	1.32	1.29	1.22
Allhyper	0.43	0.15	1.99	1.27	1.28	1.22
ANN Thyroid	0.18	-0.15	1.36	1.09	1.08	1.02
WBC	2.10	1.22	4.61	2.51	2.53	2.39
Pima Diabetes	1.74	1.54	4.45	2.47	2.47	2.41
Liver Disorders	1.81	1.44	4.45	2.48	2.49	2.38
Thyroid 0387	2.10	1.79	4.90	2.50	2.51	2.43
C. Method	1.58	1.30	4.25	2.39	2.38	2.27
Ecoli	1.71	1.40	4.45	2.48	2.48	2.41
H. Survival	2.55	2.09	5.25	3.35	3.34	3.31
Todos						
Média	1.80	0.96	2.89	2.09	2.10	1.88
Mediana	1.15	0.71	2.78	1.82	1.82	1.79
Rank	2	1	6	4	5	3
Baixa-Densidade						
Média	5.61	2.39	3.45	3.70	3.70	2.94
Mediana	5.80	2.51	3.58	3.83	3.82	3.06
Rank	6	1	3	5	4	2
Alta-Densidade						
Média	1.04	0.67	2.78	1.77	1.78	1.66
Mediana	0.80	0.38	2.58	1.49	1.45	1.30
Rank	2	1	6	4	5	3

Tabela E.2: *Porcentagem de Atributos Selecionados*

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	0.61%	1.17%	52.34%	0.43%	0.43%	0.03%
Leukemia	1.21%	12.88%	82.02%	0.29%	0.29%	0.02%
Leukemia nom.	0.57%	69.25%	38.98%	0.42%	0.42%	0.02%
Colon	1.21%	6.36%	75.96%	1.21%	1.21%	0.11%
Lymphoma	8.25%	51.11%	99.72%	2.00%	2.00%	0.16%
Lung Cancer	19.29%	99.82%	56.79%	37.50%	37.50%	6.61%
C. Arrhythmia	9.71%	47.99%	81.29%	76.59%	76.59%	10.25%
Hepatitis	55.59%	100.00%	99.41%	95.59%	95.59%	20.00%
WDBC	51.58%	87.37%	99.47%	82.63%	82.63%	32.11%
Dermatology	53.89%	100.00%	98.33%	95.56%	95.56%	41.11%
Lymphography	37.00%	88.00%	100.00%	95.00%	95.00%	24.00%
Splice Junction	23.08%	54.62%	56.92%	68.46%	68.46%	56.15%
Heart Statlog	35.17%	100.00%	100.00%	99.83%	99.83%	35.67%
HD Switz.	73.75%	100.00%	93.75%	43.75%	43.75%	40.00%
Sick	56.92%	74.62%	99.23%	98.46%	98.46%	66.92%
P. Patient	46.92%	69.23%	92.31%	76.15%	76.15%	63.08%
Hypothyroid	50.77%	84.62%	95.38%	96.92%	96.92%	66.92%
HD Hungarian	16.90%	89.31%	92.76%	41.38%	41.38%	32.41%
HD Cleveland	17.24%	87.59%	90.34%	48.62%	48.62%	28.62%
Allhypo	18.97%	85.86%	92.07%	38.62%	38.62%	32.41%
Breast Cancer	20.40%	92.00%	95.60%	57.20%	57.20%	24.00%
Allhyper	45.56%	100.00%	100.00%	86.67%	86.67%	64.44%
ANN Thyroid	38.97%	100.00%	99.66%	92.76%	92.76%	81.72%
WBC	23.33%	99.05%	95.24%	48.57%	48.57%	36.67%
Pima Diabetes	100.00%	100.00%	100.00%	95.56%	95.56%	71.11%
Liver Disorders	85.71%	85.71%	98.57%	85.71%	85.71%	74.29%
Thyroid 0387	48.75%	87.50%	100.00%	100.00%	100.00%	90.00%
C. Method	20.00%	20.00%	93.33%	100.00%	100.00%	90.00%
Ecoli	34.44%	100.00%	100.00%	100.00%	100.00%	90.00%
H. Survival	66.67%	66.67%	100.00%	100.00%	100.00%	90.00%
Todos						
Média	35.41%	75.36%	89.32%	65.53%	65.53%	42.29%
Mediana	34.81%	87.43%	95.79%	79.61%	79.61%	36.17%
Rank	1	5	6	3.5	3.5	2
Baixa-Densidade						
Média	2.37%	28.15%	69.80%	0.87%	0.87%	0.07%
Mediana	1.21%	12.88%	75.96%	0.43%	0.43%	0.03%
Rank	4	5	6	2.5	2.5	1
Alta-Densidade						
Média	42.02%	84.80%	93.22%	78.46%	78.46%	50.74%
Mediana	38.97%	88.00%	98.33%	86.67%	86.67%	41.11%
Rank	1	5	6	3.5	3.5	2

CFS Subset Eval obteve, na média a menor porcentagem de atributos selecionados (34.41%) enquanto que o *Relief F* ficou com a maior média (89.32%). Em relação ao foco do estudo que são as grandes bases de dados, ou seja as de baixa densidade, a metodologia proposta ficou atrás apenas da árvore única (UT), o que era esperado dada que a UT faz apenas uma iteração, enquanto que a metodologia faz mais de uma. Mesmo assim a metodologia proposta conseguiu compactar os dados acima de 99%. Ou seja em uma base de dados com dez mil genes, ela selecionaria, como relevantes, apenas 87. Isso vai a favor do objetivo desta proposta, que visa reduzir a base de dados de modo que um especialista consiga focar apenas em um pequeno conjunto de atributos.

Nas Tabelas E.3, E.4, E.5, E.6 e E.7 são mostrados AUC, média, mediana e *rank* médio para cada seletor em relação a cada indutor. Utilizando J48 como parâmetro, a metodologia proposta obteve os menores *ranks* médios em relação a todas as bases de dados, enquanto que a UT ficou com a pior. Para o IBk3 e Naive Bayes, CFS Subset Eval ficou em primeiro lugar seguido pela metodologia, mais uma vez o pior de todos foi a UT; no PART o resultado se repetiu, porém o resultado foi mais próximo entre os primeiros. Já para o SMO o resultado foi completamente diferente. O primeiro lugar ficou com o Relief F com o Gain Ratio em segundo muito próximo dele. A metodologia proposta ficou com o terceiro melhor rank. A UT ficou em ultimo em todas as bases testadas considerando o indutor SMO.

Tabela E.3: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o J48*

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	0.57	0.47	0.50	0.59	0.59	0.56
Leukemia	0.82	0.75	0.76	0.82	0.82	0.80
Leukemia nom.	0.92	0.92	0.89	0.90	0.90	0.75
Colon	0.85	0.88	0.84	0.75	0.75	0.66
Lymphoma	0.86	0.91	0.90	0.85	0.85	0.74
Lung Cancer	0.77	0.68	0.68	0.83	0.83	0.59
C. Arrhythmia	0.75	0.72	0.73	0.73	0.73	0.66
Hepatitis	0.98	0.97	0.97	0.97	0.97	0.88
WDBC	0.68	0.70	0.70	0.69	0.69	0.61
Dermatology	0.76	0.80	0.79	0.75	0.75	0.71
Lymphography	0.93	0.93	0.94	0.94	0.94	0.84
Splice Junction	0.45	0.55	0.55	0.55	0.55	0.50
Heart Statlog	0.97	0.96	0.96	0.96	0.96	0.87
HD Switz.	0.50	0.49	0.49	0.49	0.49	0.44
Sick	0.81	0.75	0.75	0.75	0.75	0.68
P. Patient	0.76	0.77	0.77	0.73	0.73	0.66
Hypothyroid	0.80	0.79	0.79	0.79	0.79	0.76
HD Hungarian	0.90	0.90	0.89	0.90	0.90	0.89
HD Cleveland	0.99	1.00	1.00	1.00	1.00	0.90
Allhypo	0.93	0.95	0.95	0.95	0.95	0.86
Breast Cancer	0.96	0.95	0.95	0.95	0.95	0.87
Allhyper	0.63	0.63	0.63	0.63	0.63	0.56
ANN Thyroid	0.98	0.98	0.98	0.98	0.98	0.89
WBC	0.99	0.99	0.99	0.99	0.99	0.89
Pima Diabetes	0.95	0.95	0.95	0.95	0.95	0.87
Liver Disorders	0.92	0.92	0.92	0.92	0.92	0.83
Thyroid 0387	0.76	0.75	0.75	0.75	0.75	0.67
C. Method	0.61	0.67	0.67	0.67	0.67	0.60
Ecoli	0.71	0.66	0.66	0.66	0.66	0.60
H. Survival	0.54	0.57	0.57	0.58	0.58	0.51
Todos						
Média	0.80	0.79	0.78	0.78	0.78	0.70
Mediana	0.85	0.88	0.84	0.82	0.82	0.74
Rank Médio	3.10	3.22	3.55	2.73	2.73	5.67
Baixa-Densidade						
Média	0.80	0.80	0.80	0.80	0.80	0.73
Mediana	0.80	0.79	0.79	0.79	0.79	0.71
Alta-Densidade						
Média	0.80	0.80	0.80	0.80	0.80	0.73
Mediana	0.80	0.79	0.79	0.79	0.79	0.71

Tabela E.4: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o IBk3*

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	0.61	0.56	0.56	0.60	0.60	0.56
Leukemia	0.97	0.85	0.85	0.92	0.92	0.81
Leukemia nom.	0.98	0.99	0.99	0.98	0.98	0.79
Colon	0.83	0.83	0.83	0.85	0.85	0.60
Lymphoma	0.96	0.92	0.92	0.95	0.95	0.80
Lung Cancer	0.77	0.68	0.68	0.69	0.69	0.48
C. Arrhythmia	0.74	0.66	0.66	0.67	0.67	0.64
Hepatitis	0.99	0.99	0.99	0.99	0.99	0.88
WDBC	0.74	0.79	0.79	0.76	0.76	0.70
Dermatology	0.89	0.89	0.89	0.88	0.88	0.78
Lymphography	0.98	0.98	0.98	0.98	0.98	0.88
Splice Junction	0.57	0.49	0.49	0.54	0.54	0.48
Heart Statlog	0.97	0.94	0.94	0.95	0.95	0.86
HD Switz.	0.37	0.31	0.31	0.50	0.50	0.44
Sick	0.83	0.83	0.83	0.83	0.83	0.76
P. Patient	0.90	0.87	0.87	0.86	0.86	0.76
Hypothyroid	0.87	0.85	0.85	0.84	0.84	0.77
HD Hungarian	0.89	0.85	0.85	0.90	0.90	0.79
HD Cleveland	0.78	0.74	0.74	0.76	0.76	0.70
Allhypo	0.90	0.88	0.88	0.92	0.92	0.83
Breast Cancer	0.92	0.85	0.85	0.87	0.87	0.86
Allhyper	0.65	0.66	0.66	0.64	0.64	0.60
ANN Thyroid	0.74	0.84	0.84	0.85	0.85	0.76
WBC	0.95	0.74	0.74	0.78	0.78	0.73
Pima Diabetes	0.99	0.99	0.99	0.99	0.99	0.89
Liver Disorders	0.94	0.94	0.94	0.94	0.94	0.84
Thyroid 0387	0.74	0.74	0.74	0.74	0.74	0.67
C. Method	0.58	0.64	0.64	0.64	0.64	0.58
Ecoli	0.68	0.62	0.62	0.62	0.62	0.56
H. Survival	0.59	0.63	0.63	0.63	0.63	0.58
Todos						
Média	0.81	0.78	0.78	0.80	0.80	0.71
Mediana	0.85	0.84	0.84	0.84	0.84	0.76
Rank Médio	2.48	3.52	3.40	2.93	2.93	5.73
Baixa-Densidade						
Média	0.87	0.83	0.83	0.86	0.86	0.71
Mediana	0.96	0.85	0.85	0.92	0.92	0.79
Alta-Densidade						
Média	0.80	0.78	0.78	0.79	0.79	0.71
Mediana	0.83	0.83	0.83	0.83	0.83	0.76

Tabela E.5: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o NaiveBayes*

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	0.59	0.58	0.58	0.55	0.55	0.56
Leukemia	0.99	0.97	0.98	1.00	1.00	0.80
Leukemia nom.	1.00	0.96	0.96	1.00	1.00	0.79
Colon	0.88	0.61	0.64	0.87	0.87	0.65
Lymphoma	0.86	0.80	0.80	0.91	0.91	0.81
Lung Cancer	0.74	0.71	0.71	0.74	0.74	0.54
C. Arrhythmia	0.85	0.81	0.81	0.81	0.81	0.74
Hepatitis	1.00	1.00	1.00	1.00	1.00	0.89
WDBC	0.87	0.86	0.86	0.86	0.86	0.82
Dermatology	0.89	0.91	0.91	0.91	0.91	0.80
Lymphography	0.99	0.98	0.98	0.98	0.98	0.88
Splice Junction	0.49	0.53	0.53	0.54	0.54	0.49
Heart Statlog	0.99	0.99	0.99	0.99	0.99	0.89
HD Switz.	0.39	0.39	0.39	0.46	0.46	0.43
Sick	0.88	0.90	0.90	0.90	0.90	0.81
P. Patient	0.91	0.90	0.90	0.89	0.89	0.80
Hypothyroid	0.89	0.90	0.90	0.90	0.90	0.81
HD Hungarian	0.98	0.97	0.97	0.97	0.97	0.87
HD Cleveland	0.94	0.93	0.93	0.94	0.94	0.85
Allhypo	0.94	0.93	0.93	0.93	0.93	0.83
Breast Cancer	0.98	0.97	0.97	0.98	0.98	0.89
Allhyper	0.70	0.72	0.72	0.71	0.71	0.63
ANN Thyroid	0.93	0.93	0.93	0.93	0.93	0.83
WBC	0.95	0.93	0.93	0.93	0.93	0.84
Pima Diabetes	0.99	0.99	0.99	0.99	0.99	0.89
Liver Disorders	0.97	0.97	0.97	0.97	0.97	0.87
Thyroid 0387	0.83	0.82	0.82	0.82	0.82	0.73
C. Method	0.60	0.65	0.65	0.65	0.65	0.58
Ecoli	0.70	0.69	0.69	0.69	0.69	0.63
H. Survival	0.68	0.67	0.67	0.67	0.67	0.60
Todos						
Média	0.85	0.83	0.83	0.85	0.85	0.75
Mediana	0.89	0.90	0.90	0.91	0.91	0.80
Rank Médio	2.33	3.72	3.68	2.78	2.78	5.7
Baixa-Densidade						
Média	0.86	0.79	0.79	0.86	0.86	0.72
Mediana	0.88	0.80	0.80	0.91	0.91	0.79
Alta-Densidade						
Média	0.84	0.84	0.85	0.85	0.85	0.76
Mediana	0.89	0.90	0.90	0.90	0.90	0.81

Tabela E.6: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o PART*

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	0.58	0.50	0.52	0.52	0.52	0.58
Leukemia	0.84	0.75	0.76	0.82	0.82	0.80
Leukemia nom.	0.84	0.75	0.76	0.82	0.82	0.80
Colon	0.82	0.88	0.83	0.78	0.78	0.63
Lymphoma	0.86	0.85	0.84	0.82	0.82	0.69
Lung Cancer	0.71	0.69	0.69	0.73	0.73	0.57
C. Arrhythmia	0.75	0.75	0.74	0.75	0.75	0.66
Hepatitis	0.98	0.98	0.98	0.98	0.98	0.88
WDBC	0.76	0.83	0.83	0.80	0.80	0.70
Dermatology	0.80	0.79	0.79	0.79	0.79	0.73
Lymphography	0.94	0.94	0.94	0.94	0.94	0.86
Splice Junction	0.48	0.54	0.54	0.56	0.56	0.49
Heart Statlog	0.97	0.96	0.96	0.96	0.96	0.86
HD Switz.	0.46	0.44	0.46	0.46	0.46	0.41
Sick	0.82	0.75	0.75	0.75	0.75	0.73
P. Patient	0.86	0.87	0.87	0.84	0.84	0.76
Hypothyroid	0.85	0.83	0.83	0.84	0.84	0.74
HD Hungarian	0.94	0.94	0.94	0.95	0.95	0.86
HD Cleveland	0.99	1.00	1.00	1.00	1.00	0.89
Allhypo	0.93	0.96	0.96	0.97	0.97	0.87
Breast Cancer	0.97	0.96	0.96	0.95	0.95	0.86
Allhyper	0.56	0.59	0.59	0.58	0.58	0.52
ANN Thyroid	0.98	0.98	0.98	0.98	0.98	0.88
WBC	0.99	0.99	0.99	1.00	1.00	0.90
Pima Diabetes	0.97	0.97	0.97	0.97	0.97	0.88
Liver Disorders	0.92	0.92	0.92	0.92	0.92	0.84
Thyroid 0387	0.78	0.79	0.79	0.79	0.79	0.71
C. Method	0.61	0.67	0.67	0.67	0.67	0.60
Ecoli	0.69	0.65	0.65	0.65	0.65	0.59
H. Survival	0.58	0.54	0.54	0.55	0.55	0.49
Todos						
Média	0.81	0.80	0.80	0.80	0.80	0.73
Mediana	0.84	0.83	0.83	0.82	0.82	0.74
Rank Médio	2.63	3.50	3.47	2.87	2.87	5.67
Baixa-Densidade						
Média	0.79	0.74	0.74	0.75	0.75	0.70
Mediana	0.84	0.75	0.76	0.82	0.82	0.69
Alta-Densidade						
Média	0.81	0.81	0.81	0.81	0.81	0.73
Mediana	0.85	0.83	0.83	0.84	0.84	0.74

Tabela E.7: *Valores de AUC, média, mediana e Rank médio obtido pelos experimentos para o SMO*

Base de Dados	CFS	Gain Ratio	Relief F	m100	m095	UT
CNS	0.62	0.63	0.63	0.46	0.46	0.45
Leukemia	0.97	0.98	0.98	0.90	0.90	0.62
Leukemia nom.	0.97	0.97	0.97	0.97	0.97	0.79
Colon	0.79	0.84	0.84	0.82	0.82	0.55
Lymphoma	0.98	0.98	0.98	0.97	0.97	0.69
Lung Cancer	0.75	0.63	0.63	0.67	0.67	0.51
C. Arrhythmia	0.76	0.79	0.79	0.79	0.79	0.66
Hepatitis	0.99	0.98	0.98	0.98	0.98	0.89
WDBC	0.65	0.75	0.75	0.70	0.70	0.61
Dermatology	0.85	0.87	0.87	0.85	0.85	0.74
Lymphography	0.96	0.97	0.97	0.97	0.97	0.86
Splice Junction	0.54	0.58	0.57	0.55	0.55	0.49
Heart Statlog	0.97	0.96	0.96	0.96	0.96	0.87
HD Switz.	0.48	0.47	0.48	0.49	0.49	0.44
Sick	0.83	0.84	0.84	0.84	0.84	0.75
P. Patient	0.79	0.80	0.80	0.78	0.78	0.70
Hypothyroid	0.83	0.84	0.84	0.84	0.84	0.76
HD Hungarian	0.90	0.90	0.89	0.90	0.90	0.89
HD Cleveland	0.58	0.59	0.59	0.59	0.59	0.53
Allhypo	0.50	0.50	0.50	0.50	0.50	0.45
Breast Cancer	0.65	0.75	0.75	0.74	0.74	0.59
Allhyper	0.56	0.59	0.58	0.58	0.58	0.50
ANN Thyroid	0.67	0.68	0.68	0.68	0.68	0.61
WBC	0.58	0.59	0.59	0.59	0.59	0.53
Pima Diabetes	0.97	0.97	0.97	0.97	0.97	0.87
Liver Disorders	0.95	0.95	0.95	0.95	0.95	0.85
Thyroid 0387	0.71	0.72	0.72	0.72	0.72	0.65
C. Method	0.50	0.50	0.50	0.50	0.50	0.45
Ecoli	0.63	0.63	0.63	0.63	0.63	0.57
H. Survival	0.50	0.51	0.51	0.50	0.50	0.45
Todos						
Média	0.75	0.76	0.76	0.75	0.75	0.64
Mediana	0.75	0.77	0.77	0.76	0.76	0.62
Rank Médio	4.08	2.40	2.35	3.08	3.08	6.00
Baixa-Densidade						
Média	0.72	0.73	0.73	0.73	0.73	0.65
Mediana	0.71	0.75	0.75	0.72	0.72	0.61
Alta-Densidade						
Média	0.87	0.88	0.88	0.82	0.82	0.62
Mediana	0.97	0.97	0.97	0.90	0.90	0.62