Investigação de Combinações de Técnicas de
Detecção de Ruído para Dados de Expressão
Gênica

Giampaolo Luiz Libralon

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Investigação de combinações de técnicas de detecção de ruído para dados de expressão gênica

Giampaolo Luiz Libralon

Orientador: *Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional.

USP - São Carlos

Outubro/2007

Agradecimentos

Agradeço ao meu orientador André, pela amizade, confiança em meu trabalho, auxílio em todos os momentos de dificuldade e, principalmente, pela paciência e compreensão.

À Ana Carolina Lorena, que me co-orientou durante todo o trabalho, sempre pronta para solucionar minhas dúvidas e propor melhorias ao projeto de pesquisa.

À minha namorada Carla, pela alegria que traz à minha vida e por sempre ter me apoiado e estimulado na realização desse trabalho, não permitindo que eu desistisse desse sonho, nas várias vezes em que desanimei.

Agradeço a todas as pessoas que, em algum momento, me auxiliaram direta ou indiretamente, seja com discussões sobre o tema do trabalho ou compartilhando algoritmos e dados.

Meu profundo agradecimento a Deus, por nunca ter me faltado em todos os passos de minha vida, me dando força para lutar e capacidade e determinação para atingir meus sonhos. Tenho sorte por ser guiado e presenteado por Ele.

Agradeço à minha família pelo apoio oferecido, especialmente à minha mãe, pela dedicação e carinho.

Para finalizar, agradeço o suporte financeiro da FAPESP.

Resumo

Ruído pode ser definido como um exemplo em um conjunto de dados que aparentemente é inconsistente com o restante dos dados existentes, pois não segue o mesmo padrão dos demais. Ruídos em conjuntos de dados podem reduzir o desempenho das técnicas de Aprendizado de Máquina (AM) empregadas e aumentar o tempo de construção da hipótese induzida, assim como sua complexidade. Dados são geralmente coletados por meio de medições realizadas em um domínio de interesse. Nesse sentido, nenhum conjunto de dados é perfeito. Erros de medições, dados incompletos, errados, corrompidos ou distorcidos, falhas humanas ou dos equipamentos utilizados, dentre muitos outros fatores, contribuem para a contaminação dos dados, e isso é particularmente verdadeiro para dados com elevada dimensionalidade. Sendo assim, a detecção de ruídos é uma tarefa crítica, principalmente em ambientes que exigem segurança e confiabilidade, uma vez que a presença desses pode indicar situações que degradam o desempenho do sistema ou a segurança e confiabilidade das informações. Algoritmos para a detecção e remoção de ruídos podem aumentar a confiabilidade de conjuntos de dados ruidosos. Nesse âmbito, esse trabalho investiga técnicas de detecção de ruído baseadas em distância, em que a remoção de ruídos é feita em uma etapa de pré-processamento, aplicadas a problemas de classificação de dados de Expressão Gênica, caracterizados pela presença de ruídos, elevada dimensionalidade e complexidade. O objetivo é melhorar o desempenho das técnicas de AM empregadas para solucioná-los. Por fim, combinações de técnicas de detecção de ruído são implementadas de modo a analisar a possibilidade de melhorar, ainda mais, o desempenho obtido.

Palavras-chave: Ruído, Expressão Gênica, Aprendizado de Máquina, Combinação de Classificadores.

Abstract

Noise can be defined as an example which seems to be inconsistent with the remaining ones in a data set. The presence of noise in data sets can decrease the performance of Machine Learning (ML) techniques in the problem analysis and also increase the time taken to build the induced hypothesis and its complexity. Data are collected from measurements made which represent a given domain of interest. In this sense, no data set is perfect. Measurement errors, incomplete, corrupted, wrong or distorted examples, equipment problems or human fails, besides many other related factors, help contaminating the data, and this is particularly true for data sets with high dimensionality. For this reason, noise detection is a critical task, specially in domains which demand security and trustworthiness, since the presence of noise can lead to situations which degrade the system performance or the security and trustworthiness of the involved information. Algorithms to detect and remove noise may increase trustworthiness of noisy data sets. Based on that, this work evaluates distance-based noise detection techniques, in which noise removal is done by a pre-processing phase, in gene expression classification problems, characterized by the presence of noise, high dimensionality and complexity. The objective is to improve the performance of ML techniques used to solve these problems. Next, ensembles of noise detection techniques are developed in order to analyze the possibility to further improve the performance obtained.

Keywords: Noise, Gene Expression Problems, Machine Learning, Ensembles.

Sumário

Agradecimentos	i
Resumo	iii
Abstract	v
Sumário	vii
Lista de Abreviaturas e Siglas	ix
Lista de Figuras	x
Lista de Tabelas	xii
 1 Introdução	 1
 2 Aprendizado de Máquina	 5
2.1 Considerações Iniciais	5
2.2 Conceitos Básicos de AM	6
2.3 Técnicas de AM avaliadas	9
2.4 Considerações Finais	16
 3 Pré-processamento de Dados	 17
3.1 Considerações Iniciais	17
3.2 Detecção de Ruídos	19
3.3 Detecção de Ruídos baseado em distância	23

3.4	Combinação de Técnicas de Detecção de Ruído	29
3.5	Considerações Finais	30
4	Experimentos Iniciais	33
4.1	Considerações Iniciais	33
4.2	Descrição da Proposta	33
4.3	Metodologia	34
4.3.1	Conjuntos de Dados	35
4.4	Experimentos e Resultados	37
4.5	Considerações Finais	45
5	Experimentos com dados de Expressão Gênica	49
5.1	Considerações Iniciais	49
5.2	Avaliação dos Experimentos	49
5.3	Conjuntos de Dados	51
5.4	Experimentos e Resultados	53
5.5	Considerações Finais	70
6	Experimentos realizados com a Combinação	71
6.1	Considerações Iniciais	71
6.2	Combinação de Técnicas de Detecção de Ruído	71
6.3	Considerações Finais	84
7	Conclusões	87
	Referências Bibliográficas	99

Lista de Abreviaturas e Siglas

AD Árvore de Decisão

AG Algoritmo Genético

AM Aprendizado de Máquina

RN Rede Neural Artificial

SVs Support Vectors

SVMs Support Vector Machines

UCI Universidade da Califórnia em Irvine

Lista de Figuras

2.1	Conjunto de dados linearmente separável e hiperplano separador	12
2.2	Conjunto de dados com classes não-linearmente separáveis	12
2.3	Conjunto de dados não linear	13
3.1	Tipos de ruído encontrados em conjuntos de dados.	20
6.1	Exemplos removidos pelas combinações desenvolvidas pela metodologia de consenso e de votação por maioria, para o conjunto <i>Colon</i>	82
6.2	Exemplos removidos pelas combinações desenvolvidas pela metodologia de consenso e de votação por maioria, para o conjunto <i>Golub64</i>	83

Lista de Tabelas

4.1	Descrição dos conjuntos de dados utilizados nos experimentos.	38
4.2	Redução média dos conjuntos de dados originais (em porcentagem) com a aplicação das técnicas de detecção de ruído.	39
4.3	Taxas de erro das SVMs aplicadas sobre os dados originais e pré-processados.	40
4.4	Taxas de erro das ADs aplicadas sobre os dados originais e pré-processados.	41
4.5	Resumo dos resultados obtidos pelo classificador SVM.	42
4.6	Resumo dos resultados obtidos pelo classificador C4.5.	42
4.7	Número médio de SVs obtidos pelas SVMs aplicadas sobre os dados originais e pré-processados.	43
4.8	Tamanho médio das árvores produzidas pelo algoritmo C4.5 aplicado sobre os dados originais e pré-processados.	44
4.9	Tempos (em segundos) da aplicação das técnicas de eliminação de ruídos. .	45
4.10	Tempos de treinamento (em segundos) das SVMs aplicadas sobre os dados originais e pré-processados.	46
4.11	Tempos (em segundos) das ADs aplicadas sobre os dados originais e pré-processados.	47
5.1	Descrição dos conjuntos de dados utilizados nos experimentos.	53
5.2	Redução média dos conjuntos de dados originais (em porcentagem) com a aplicação das técnicas de detecção de ruído.	55

5.3	Taxas de erro das SVMs aplicadas sobre os dados originais e pré-processados.	56
5.4	Taxas de erro das ADs aplicadas sobre os dados originais e pré-processados.	57
5.5	Taxas de erro do algoritmo RIPPER aplicado sobre os dados originais e pré-processados.	58
5.6	Resumo dos resultados obtidos pelo classificador SVM.	60
5.7	Resumo dos resultados obtidos pelo classificador C4.5.	60
5.8	Resumo dos resultados obtidos pelo classificador RIPPER.	60
5.9	Resultado da análise estatística do classificador SVM pelo teste de comparações múltiplas de Dunn.	61
5.10	Resultado da análise estatística do classificador C4.5 pelo teste de comparações múltiplas de Dunn.	62
5.11	Resultado da análise estatística do classificador RIPPER pelo teste de comparações múltiplas de Dunn.	63
5.12	Tempos (em segundos) da aplicação das técnicas de detecção de ruído. . .	63
5.13	Tempos de treinamento (em segundos) das SVMs aplicadas sobre os dados originais e pré-processados.	64
5.14	Tempos (em segundos) das ADs aplicadas sobre os dados originais e pré-processados.	65
5.15	Tempos (em segundos) do algoritmo RIPPER aplicado sobre os dados originais e pré-processados.	66
5.16	Número médio de SVs obtidos pelas SVMs aplicadas sobre os dados originais e pré-processados.	67
5.17	Tamanho médio das árvores produzidas pelo algoritmo C4.5 aplicado sobre os dados originais e pré-processados.	68
5.18	Número médio de regras produzidas pelo algoritmo RIPPER aplicado sobre os dados originais e pré-processados.	69
6.1	Resultados obtidos pelas combinações desenvolvidas para o classificador SVM.	74
6.2	Resultados obtidos pelas combinações desenvolvidas para o classificador C4.5.	75

6.3	Resultados obtidos pelas combinações desenvolvidas para o classificador RIPPER.	76
6.4	Resultados obtidos pelas novas combinações desenvolvidas para o classificador SVM.	77
6.5	Resultados obtidos pelas novas combinações desenvolvidas para o classificador C4.5.	78
6.6	Resultados obtidos pelas novas combinações desenvolvidas para o classificador RIPPER.	78
6.7	Resultados obtidos pelas combinações por consenso desenvolvidas.	81
6.8	Resumo dos resultados obtidos pelas combinações desenvolvidas.	82

Introdução

Ruído pode ser definido como um exemplo (ou conjunto de exemplos) que aparentemente é inconsistente com o restante dos exemplos encontrados em um conjunto de dados, pois não segue o mesmo padrão dos demais (BARNETT e LEWIS [1994]). Vale ressaltar que exemplos considerados ruído podem, em alguns casos, representar dados corretos, que seguem padrões diferenciados dos demais exemplos presentes no conjunto de dados (JOHN [1995]).

A detecção de ruídos é uma tarefa crítica principalmente em ambientes que exigem segurança, uma vez que a presença destes pode indicar situações que degradam o desempenho do sistema ou a segurança das informações.

Trabalhos preocupados com a identificação e remoção de ruídos seguem diferentes linhas de pesquisa e se valem de diversas técnicas para esse fim. As técnicas utilizadas são provenientes de duas diferentes áreas: Estatística (técnicas paramétricas, não paramétricas, semi-paramétricas e baseadas em distância) e Aprendizado de Máquina (AM - algoritmos baseados em regras ou na geração de árvores de decisão, Redes Neurais Artificiais - RNs, dentre outros) (HODGE e AUSTIN [2004]).

A detecção de ruídos é parte integrante de uma grande variedade de aplicações práticas tais como:

- Detecção de fraudes, seja em cartões de crédito, recebimento de pensões ou mesmo

utilização de celulares;

- Empréstimos bancários, na detecção de aplicações fraudulentas ou na identificação do perfil de potenciais clientes que cometerão fraudes;
- Detecção de intrusão em redes de computadores, manutenção do desempenho ou identificação de gargalos de arquitetura;
- Diagnóstico de falhas em equipamentos como motores, turbinas, geradores, instrumentos ou sondas espaciais, dentre outros;
- Análise de imagens, na identificação de novas estruturas em imagens de satélite, remoção de ruídos, melhoria da qualidade de imagens médicas e detecção de novidades em imagens, por exemplo, provenientes de um sistema de navegação de robôs;
- Pesquisas farmacêuticas, na identificação de novas estruturas moleculares; dentre muitas outras aplicações existentes na literatura.

De acordo com HODGE e AUSTIN [2004], abordagens utilizadas para detecção de ruídos se enquadram em três categorias:

- *Identificação de ruídos sem o conhecimento prévio dos dados*: essa metodologia é análoga a problemas de agrupamento de dados, categorizando os dados ou ruídos em diferentes grupos, de acordo com suas características afins.
- *Detecção de anomalias em sistemas ou conjuntos de dados*: é análoga a problemas de classificação por aprendizado supervisionado, necessitando de conhecimento prévio sobre os dados ou problema investigado. Nessa metodologia, uma amostra representativa do problema (que cubra todo o domínio analisado) deve ser apresentada ao algoritmo de classificação, de modo a possibilitar a criação de um modelo adequado que permita novas classificações com um grau de acerto considerado satisfatório.
- *Detecção de novidades*: em que apenas dados não ruidosos ou muito poucos casos ruidosos são modelados. Essa metodologia é semelhante à tarefa de detecção ou aprendizado semi-supervisionado, pois dados corretos são apresentados ao algoritmo e esse aprende a reconhecer possíveis ruídos.

Nesse sentido, a escolha de qual metodologia utilizar deve ser feita de acordo com as características do problema e dos dados que serão analisados. Deve-se optar por algoritmos ou conjunto de algoritmos que sejam capazes de modelar o problema estudado e de

identificar possíveis ruídos de modo preciso. Os algoritmos selecionados devem também ser escaláveis aos conjuntos de dados que serão avaliados.

Não existe uma única abordagem ou um determinado algoritmo que seja aplicável a todos os problemas de detecção de ruídos existentes sem perda de desempenho. O que deve ser analisado é quais técnicas e metodologia empregar na resolução do problema a ser explorado, dentre a grande variedade existente na literatura.

Sendo assim, nesse trabalho optou-se pela análise de problemas em Bioinformática, que apresentam elevada dimensionalidade e grande quantidade de ruído presente nos conjuntos de dados que os descrevem. Esses problemas são freqüentes em análise de dados de expressão gênica.

Duas etapas são realizadas nos experimentos; a primeira, de pré-processamento, composta por técnicas de detecção de ruído baseadas em distância, com o objetivo de identificar e remover dados ruidosos encontrados. A segunda, de classificação, busca encontrar, por meio de técnicas de AM, um modelo adequado que represente os problemas abordados, agora livres de ruído. Essa segunda etapa visa avaliar o desempenho das técnicas de detecção de ruído investigadas, assim como as combinações de algumas dessas técnicas desenvolvidas.

As combinações de técnicas de detecção de ruído desenvolvidas são compostas pelas técnicas que apresentaram os melhores resultados individualmente e têm como objetivo complementá-las, de modo a melhorar o desempenho final encontrado.

Para facilitar a compreensão deste trabalho, a dissertação está organizada do seguinte modo. No Capítulo 2, são introduzidos conceitos sobre Aprendizado de Máquina (AM) e apresentados os algoritmos de AM utilizados no trabalho. No Capítulo 3 são apresentados conceitos sobre ruídos em conjuntos de dados, assim como trabalhos encontrados na literatura e as técnicas de detecção de ruído avaliadas. Nos capítulos 4 e 5 são descritos a metodologia do trabalho, conjuntos de dados selecionados, experimentos realizados e resultados obtidos. O Capítulo 6 apresenta as combinações de técnicas de detecção de ruído desenvolvidas, assim como os experimentos realizados. Por fim, no Capítulo 7 são apresentadas as conclusões sobre o presente trabalho e possíveis extensões a serem realizadas em trabalhos futuros.

Aprendizado de Máquina

2.1 *Considerações Iniciais*

Aprendizado de Máquina (AM) é um campo de pesquisa em Inteligência Artificial que estuda o desenvolvimento de métodos capazes de extrair conhecimento a partir de conjuntos de dados. Pesquisas de AM têm por objetivo desenvolver técnicas computacionais de aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de modo automático (MITCHELL [1997]).

Algumas das razões que fazem de AM um campo de investigações científicas importante são (LORENA e CARVALHO [2003]):

- Pesquisas em AM podem aumentar a compreensão de como ocorre o aprendizado nos seres vivos;
- Algumas tarefas são melhor definidas por meio de exemplos. A descoberta de relacionamentos entre estes dados por seres humanos é, em geral, custosa e ineficiente;
- Em grande parte dos conjuntos de dados existentes há importantes relacionamentos entre os dados, que podem ser extraídos por meio de algoritmos de AM;
- A quantidade de conhecimento disponível sobre algumas tarefas é muito grande para ser assimilada e analisada por seres humanos;

- Os preditores gerados por técnicas de AM são capazes de lidar com situações não previstas durante seu desenvolvimento, sem a necessidade de uma nova fase de projeto.

Existem diversos algoritmos de AM descritos na literatura, dentre estes é possível citar aqueles que têm inspiração nos sistemas biológicos (como as Redes Neurais - RNs - e os Algoritmos Genéticos - AGs), nos processos cognitivos (sistemas de Raciocínio Baseado em Casos), na lógica matemática (Árvores de Decisão - ADs), na Teoria de Aprendizado Estatístico (*Support Vector Machines* - SVMs), dentre outros.

Apesar de AM possibilitar o desenvolvimento de ferramentas poderosas para a aquisição automática de conhecimento, deve-se observar que não existe um único algoritmo de AM que apresente o melhor desempenho para todos os problemas. Nesse sentido, é importante compreender o poder e a limitação dos diversos algoritmos de AM por meio de uma metodologia que permita avaliar os conceitos induzidos por esses algoritmos em diferentes problemas (MONARD e BARANAUSKAS [2003]).

O uso de técnicas de AM envolve o conhecimento e uso de diversos conceitos próprios dessa área. Alguns conceitos principais são introduzidos na Seção 2.2. Na Seção 2.3 são descritas as técnicas de AM utilizadas neste trabalho. As Considerações Finais deste capítulo são apresentadas na Seção 2.4.

2.2 Conceitos Básicos de AM

De modo geral, os diferentes algoritmos de AM podem ser divididos de acordo com seu paradigma de aprendizado. Pode-se dizer que o paradigma de aprendizado determina o modo como se dará o aprendizado de um algoritmo de AM por meio dos dados analisados (HAYKIN [1999]). Nesse sentido, os três principais paradigmas de aprendizado podem ser descritos como:

- *Aprendizado Supervisionado*: existe um supervisor ou mentor externo, que representa o conhecimento do ambiente por meio de conjuntos de dados no formato (*entrada, saída*). O algoritmo de AM é então treinado a partir dos conjuntos de exemplos rotulados visando aprender um modelo que permita resolver o problema apresentado. Este é o paradigma utilizado neste trabalho, que utiliza supervisores para ajustar os parâmetros de um modelo durante o processo de aprendizado;

- *Aprendizado não-supervisionado*: Os exemplos no conjunto de dados relacionado ao problema a ser aprendido não estão rotulados. O algoritmo de AM aprende a representar ou agrupar os exemplos segundo alguma medida de similaridade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem na compreensão dos dados (SOUTO et. al. [2003]);
- *Aprendizado por reforço*: o aprendizado ocorre por meio de punições e recompensas, que são utilizadas para ajustar o algoritmo de AM dependendo do desempenho apresentado.

Existem também algoritmos de AM que seguem um paradigma de aprendizado Evolutivo, cujo principal representante são os AGs (GOLDBERG [1989]). O paradigma de aprendizado Evolutivo engloba algoritmos cujos conceitos e princípios se baseiam na Teoria da Evolução e Genética.

Os algoritmos de AM investigados neste trabalho seguem o paradigma de aprendizado supervisionado, ou seja, a partir de um conjunto de dados inicial, um classificador (modelo ou hipótese) é treinado para prever a classe de novos dados. No contexto desse paradigma, as técnicas de AM podem ser subdivididas em duas classes gerais: técnicas baseadas em instâncias e técnicas indutivas (MITCHELL [1997]).

As técnicas indutivas constroem um modelo formal do conceito a ser aprendido a partir das amostras de treinamento. As técnicas baseadas em instâncias simplesmente armazenam todos os exemplares disponíveis para uso durante a tarefa de predição. A classe de um novo exemplo é então dada de acordo com sua similaridade com os dados armazenados. Como exemplos de técnicas baseadas em instâncias destacam-se o algoritmo *k*-vizinhos mais próximos (*k-nearest neighbors* - *k-NN*) e os sistemas de Raciocínio Baseado em Casos. Para a abordagem indutiva, é possível citar as ADs e as RNs. O classificador produzido representa uma aproximação da função objetivo que melhor descreve os dados avaliados.

A generalização de um classificador é definida como sua capacidade de prever corretamente a classe de novos exemplos, sendo avaliada por seu desempenho em conjuntos de teste. No caso em que o modelo se especializa nos dados utilizados em seu treinamento, apresentando baixa taxa de acerto quando confrontado com novos dados, tem-se a ocorrência de *overfitting*. Existe também a possibilidade de se induzir hipóteses que apresentam baixa taxa de acerto mesmo no conjunto de treinamento. Esse fato é denominado *underfitting*. De acordo com MONARD e BARANAUSKAS [2003], essa situação

pode ocorrer quando os exemplos de treinamento disponíveis são pouco representativos ou o modelo obtido é demasiado simples.

Para a determinação das taxas de erro e acerto de um classificador, tem-se como regra geral a divisão do conjunto de dados original em dois subconjuntos disjuntos: de treinamento e de testes. O subconjunto de treinamento é utilizado no aprendizado do problema, e o subconjunto de testes é utilizado para medir a efetividade do aprendizado.

Dentre as metodologias existentes para obtenção das estimativas da taxa de acerto (ou da taxa de erro), os mais comumente utilizados são as metodologias *r-fold-cross-validation* e *bootstrap* (MITCHELL [1997]).

Na metodologia *r-fold-cross-validation*, os exemplos são aleatoriamente divididos em r partições mutuamente exclusivas de tamanho aproximadamente igual. Os exemplos nas $(r - 1)$ partições são utilizados para treinamento e a hipótese induzida é testada na partição remanescente. Este processo é repetido r vezes, cada vez considerando uma partição diferente para teste. O erro final é a média dos erros calculados em cada uma das r partições.

O método *bootstrap* consiste em repetir o processo de classificação diversas vezes (EFRON e TIBSHIRANI [1993]). A partir dos experimentos replicados, são estimados valores, como o erro ou o *bias*, sendo que cada experimento é conduzido com base em um novo conjunto de treinamento obtido por amostragem com reposição do conjunto original. Os exemplos não selecionados compõem os conjuntos de teste.

Para uma dada amostra no método *bootstrap*, um exemplo de treinamento tem probabilidade $1 - (1 - 1/n)^n$ de ser selecionado pelo menos uma vez em cada uma das n vezes nas quais os exemplos são aleatoriamente selecionados a partir do conjunto original (MONARD e BARANAUSKAS [2003]). O erro final é estimado como a média dos erros sobre o número de iterações (número de vezes que o processo de classificação é repetido).

Nas próximas seções são introduzidos os algoritmos de AM investigados neste trabalho na etapa de classificação e análise dos conjuntos de dados.

2.3 Técnicas de AM avaliadas

k-Vizinhos Mais Próximos - k-NN

Embora o algoritmo *k-NN* não seja explicitamente utilizado neste trabalho, todas as técnicas de detecção e eliminação de ruído avaliadas, que consideram a distância de um dado a outros exemplos no conjunto de dados, são baseadas nesse algoritmo, que é o representante mais simples da classe de técnicas de AM supervisionadas baseadas em instâncias (MITCHELL [1997]).

Seu funcionamento consiste em armazenar todas os exemplos de treinamento e produzir saídas baseadas em medidas de similaridade entre estes dados.

Um novo dado $\mathbf{x}$ apresentado a este algoritmo é classificado por meio da busca de um exemplo $\mathbf{y}$, existente no conjunto de dados, que se encontre mais próximo (apresente a menor distância) do novo dado. A classe desse novo dado $\mathbf{x}$ é considerada como sendo idêntica à classe do exemplo $\mathbf{y}$ que se encontra mais próximo dele (WILSON e MARTINEZ [1997b]).

Uma característica desse algoritmo é a possibilidade de escolher o valor de k , ou seja, o número de exemplos considerados no cálculo da distância entre um candidato a ruído e seus vizinhos mais próximos. O valor do parâmetro k é determinante no desempenho obtido pelo algoritmo *k-NN*, geralmente esse valor é um número inteiro pequeno e ímpar (por exemplo 1, 3 ou 5).

Para determinar os vizinhos mais próximos, esse algoritmo utiliza uma medida para calcular a distância entre os dados analisados. Em geral, a medida de distância empregada é a Euclidiana, representada pela Equação 2.1:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{q=1}^m (x_{iq} - x_{jq})^2} \quad (2.1)$$

em que $\mathbf{x}_i$ e $\mathbf{x}_j$ são dois exemplos pertencentes ao espaço R^m , e x_{iq} e x_{jq} correspondem aos valores de suas coordenadas (atributos).

Um problema dessa medida está na pressuposição de que os dados correspondem a pontos no espaço m -dimensional R^m , ou seja, que os atributos são numéricos (contínuos). Para a computação da distância entre dados com atributos categóricos, outros métodos são utilizados, como pode ser observado em BATISTA et. al. [2000].

O algoritmo k -NN apresenta boa capacidade de generalização para diversas aplicações, já que possui a habilidade de construir aproximações locais da função objetivo, diferentes para cada novo dado a ser classificado. Esta característica, compartilhada por todas as técnicas baseadas em instâncias, se mostra vantajosa quando a função objetivo apresenta elevada complexidade, porém pode ser descrita por uma coleção de aproximações locais de menor complexidade (MITCHELL [1997]).

Uma vez que todos os dados de treinamento são armazenados em memória, tem-se como desvantagens o alto custo de armazenamento, elevado tempo para classificação de novos dados (uma vez que todo o conjunto de dados deve ser percorrido), e a possibilidade de degradação da generalização na presença de dados ruidosos, pois esses são considerados na classificação dos novos exemplos analisados (WILSON e MARTINEZ [1997b]).

Support Vector Machines - SVMs

As SVMs (VAPNIK [1995]) englobam uma classe de algoritmos de AM que se baseia na Teoria de Aprendizagem Estatística, combinando controle de generalização com uma técnica para tratar o problema da dimensionalidade. Sua formulação resulta em um problema de otimização quadrática e envolve o princípio de Minimização do Risco Estrutural (*Structural Risk Minimization - SRM*), em vez do princípio de Minimização do Risco Empírico (*Empirical Risk Minimization - ERM*). SRM minimiza o erro de generalização, em oposição ao ERM que minimiza o erro nos dados de treinamento, conferindo ao algoritmo SVM uma maior capacidade de generalização sobre o conjunto de dados.

Basicamente, uma SVM realiza um mapeamento não linear dos dados em um espaço de dimensão elevada em que um hiperplano ótimo (com a maior margem de separação possível) possa ser construído, de modo a permitir a separação das classes positivas e negativas.

As SVMs possibilitam dois modos diferentes para abordar os problemas investigados. No primeiro, os erros de classificação não são considerados, ou seja, dados que não seguem os valores típicos de sua classe irão desviar o hiperplano de separação de forma que este dado (incorreto) seja classificado como correto. No segundo, são estabelecidas variáveis de folga, de modo que dados fora dos valores típicos da sua classe podem ser desconsiderados, dependendo da quantidade de folga que se use, diminuindo assim a probabilidade de erro de classificação.

SVMs têm obtido desempenho comparável, ou mesmo superior, a técnicas clássicas de

AM em diversos domínios. Por isso são cada vez mais utilizadas na solução de diferentes tipos de problemas (MULLER et. al. [2001]), tais como:

- *Classificação de padrões*: reconhecimento de dígitos, detecção de faces, classificação de textos, reconhecimento de objetos 3D, detecção de *spam*, dentre outros;
- *Regressão de funções*: predição de séries temporais, previsão de comportamento de bolsas de valores, dentre outros;
- *Processamento de sinais*: classificação de sinais sísmicos, estimação de densidade, classificação de DNA, etc.

Pode-se destacar também seu uso em Biologia Molecular (TAKEUCHI e COLLIER [2003]), análise de textos (JOACHIMS [2002]), bioinformática (NOBLE [2004]) e problemas de visão computacional (ZHENG et. al. [2004]).

As SVMs podem ser distinguidas em três diferentes tipos funcionais, de acordo com suas características e o problema abordado:

- *SVMs Lineares para Dados de Classes Linearmente Separáveis*: Originalmente, as SVMs são classificadores que conseguem discriminar apenas dados de classes linearmente separáveis, por meio da construção de um hiperplano ótimo. Classificadores lineares são aqueles capazes de separar exemplares de classes distintas por meio da utilização de um hiperplano. Nesse caso, as classes são ditas linearmente separáveis. A Figura 2.1 ilustra um conjunto de pontos bidimensionais linearmente separáveis e o hiperplano separador. É importante ressaltar que, para um conjunto de treinamento S , linearmente separável, podem existir diversos hiperplanos separadores. Nesse sentido, a SVM optará pelo hiperplano que maximize a margem de separação das classes, de modo a conferir a melhor generalização, ou seja, aquele hiperplano cujo Risco Estrutural seja mínimo.
- *SVMs Lineares para Dados de Classes Não-linearmente Separáveis*: Para a maioria dos problemas reais, os dados não são bem comportados, com elementos de classes distintas sendo perfeitamente discriminados por um hiperplano. De fato, em muitas aplicações, seja pela natureza do domínio ou por ruídos presentes nos dados, ocorre a sobreposição de classes, com elementos de uma classe confundindo-se com elementos de outra. Nesses casos, a utilização de um hiperplano que separe perfeitamente os pontos não é possível. Assim, diz-se que os dados são não-linearmente separáveis.

Um exemplo desse tipo de conjunto de dados é apresentado na Figura 2.2. Uma forma de contornar este problema é a introdução de um conjunto de variáveis de folga permitindo que erros de classificação possam ocorrer. Uma variável de folga β_i mede o quão afastado um ponto x_i encontra-se do hiperplano contendo os vetores de suporte de sua classe. Desse modo, quanto maior seu valor, pior classificado o ponto estará.

- *SVMs Não-lineares*: Para lidar com dados não linearmente separáveis, as SVMs não-lineares transformam os elementos de seu espaço de entrada original (os dados de entrada) para um novo, de alta dimensionalidade (potencialmente infinita), conhecido como espaço de características (*feature space*) e tenta projetar um classificador linear neste espaço. Um conjunto de pontos não linearmente separáveis é representado na Figura 2.3.

Figura 2.1: Conjunto de dados linearmente separável e hiperplano separador.

Figura 2.2: Conjunto de dados com classes não-linearmente separáveis.

De acordo com SMOLA e SCHOOLKOPF [2002], as principais características presentes nas SVMs que tornam seu uso atrativo são:

- *Boa capacidade de generalização*: como as SVMs implementam princípios da Teoria do Aprendizado Estatístico, conseguem minimizar a probabilidade de erros quando

Figura 2.3: Conjunto de dados não linear.

confrontadas com dados não vistos durante a fase de treinamento, dessa forma, os classificadores gerados por uma SVM alcançam, em geral, bons resultados em termos de generalização;

- *Robustez em dados de elevada dimensionalidade:*, as SVMs são tolerantes a problemas comumente encontradas em outros algoritmos de AM quando expostos a dados de grande dimensão, como *overfitting* e degradação acentuada de desempenho;
- *Convergência da função objetivo:* a aplicação das SVMs implica na otimização de uma função quadrática, que possui apenas um mínimo global, não apresentando mínimos locais;
- *Teoria bem definida:* a base teórica das SVMs é bem estabelecida dentro da Matemática e da Estatística.

Suas principais deficiências encontram-se em sua sensibilidade na escolha dos parâmetros pelo usuário e na dificuldade de interpretação do modelo produzido por esta técnica (LORENA e CARVALHO [2003]).

Algoritmo C4.5

O algoritmo C4.5 (QUINLAN [1993]) é um dos sucessores do algoritmo ID3. O algoritmo ID3 é um dos membros de uma família mais geral de algoritmos de AM indutivos denominada *Top Down Induction of Decision Trees*. Nesse sentido, o algoritmo C4.5 é um algoritmo de aprendizado simbólico que gera árvores de decisão (ADs) a partir de um conjunto de treinamento.

As ADs são estruturas recursivas que organizam suas informações em uma árvore composta por nós e ramificações (BARANAUSKAS [2001]). Cada nó representa um

atributo e está associado a um teste relevante para classificação dos dados, ou a uma classe, no caso do nó ser um nó folha. As ramificações representam cada um dos possíveis resultados do teste aplicado. A complexidade de uma AD é avaliada de acordo com seu tamanho, quanto maior o tamanho da árvore, maior a complexidade.

Uma AD está sujeita a *overfitting* (MITCHELL [1997]). Para evitar este efeito, uma etapa de poda, que elimina os ramos considerados pouco representativos (nós redundantes ou pouco expressivos), é geralmente aplicada à árvore treinada. O algoritmo C4.5 implementa esse procedimento.

Uma AD é produzida do seguinte modo: um atributo é escolhido e, de acordo com seu valor, o conjunto de treinamento é dividido. Para cada subconjunto criado, um novo atributo é escolhido para dividi-lo. Esse processo continua até que um subconjunto contendo somente exemplos de uma mesma classe seja obtido. Quando isso ocorre, um nó folha é criado e rotulado com o mesmo nome da respectiva classe. A classificação de um exemplo se dá seguindo os nós e as ramificações existentes até que um nó folha seja alcançado, ou seja, para classificar um novo exemplo, o algoritmo C4.5 percorre a AD produzida, começando pela raiz até alcançar um nó folha. A classe do novo exemplo analisado será a mesma do rótulo do nó folha alcançado.

Algumas vantagens das ADs são a compreensibilidade das regras de classificação produzidas, a facilidade de manutenção, flexibilidade e velocidade de treinamento. Em contrapartida, são pouco robustas a exemplos de elevada dimensionalidade (com grande quantidade de atributos).

Algoritmo RIPPER

O algoritmo RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*) (COHEN [1995]) é um algoritmo de indução de regras que foi projetado para ser eficiente (obter baixas taxas de erro) em conjuntos de dados que apresentam ruídos e possuem elevada dimensionalidade.

Algoritmos de indução de regras são mais flexíveis e incrementais que algoritmos geradores de ADs, uma vez que, com a inclusão de novos dados, novas regras podem ser adicionadas ou modificadas sem afetar as demais existentes no conjunto. No caso das ADs, uma nova árvore precisa ser produzida para representar os novos dados, e a anterior é eliminada (HODGE e AUSTIN [2004]).

O algoritmo RIPPER se baseia no algoritmo IREP (*Incremental Reduce Error Pruning*) (FÜRNKRANZ e WIDMER [1994]). Seu funcionamento consiste em chamar repetidas vezes o algoritmo IREP até a obtenção de um conjunto final de regras que possua pequena taxa de erro de classificação.

O algoritmo IREP constrói o conjunto de regras, uma regra por vez, e apresenta duas etapas distintas: a de crescimento e a de poda. Primeiro, os dados são inicialmente divididos em dois subconjuntos, o de crescimento e o de poda. A fase de crescimento, em que as regras são produzidas, cria um conjunto inicial de regras, iniciando a partir de uma regra vazia e adicionando, repetidamente, condições à regra existente. Essas condições maximizam a cobertura da regra criada, representando uma maior gama de estados possíveis.

O critério de parada para que as condições deixem de ser adicionadas é a cobertura completa pela regra de todos os estados de entrada ou se a adição de uma condição não melhorar a taxa de acerto da regra sendo produzida. Após a etapa de crescimento ou construção de uma regra, essa vai ser imediatamente otimizada na etapa de poda do algoritmo.

Essa etapa visa prevenir que as regras criadas se tornem demasiado específicas, evitando *overfitting*. É a etapa de otimização do algoritmo. Nela, cada condição que contribua para a diminuição da taxa de acerto do algoritmo é eliminada. Para isso, o conjunto de treinamento completo é utilizado de modo a, inicialmente, criar uma regra de substituição e, em seguida, uma regra revisada (após a etapa de poda), para cada uma das regras existentes no conjunto de regras inicialmente produzido.

Para cada regra do conjunto de regras inicialmente produzido, o algoritmo avalia a qualidade dessa regra, e das regras de substituição e revisada criadas, mantendo apenas aquela que confira melhor taxa de acerto ao algoritmo. Esse processo é repetido até que todas as classes sejam cobertas ou as regras apresentem uma taxa de erro muito reduzida. Por fim, o algoritmo RIPPER busca criar, a partir de regras residuais existentes, novas regras que cubram os dados ainda não representados pelas regras já produzidas e, em seguida, remove quaisquer regras existentes que estejam deteriorando a taxa de acerto final do mesmo.

Um dos problemas do algoritmo RIPPER está no fato dele eliminar condições idênticas que fazem parte de regras distintas pertencentes a classes diferentes, mesmo que essas condições representem características semelhantes entre as classes existentes, com o objetivo de reduzir sua taxa de erro final.

2.4 Considerações Finais

Neste capítulo foram apresentados os principais conceitos relacionados à área de AM, uma linha de pesquisa que estuda o desenvolvimento de métodos capazes de extrair conhecimento a partir de conjuntos de dados.

Diferentes técnicas de AM são encontradas na literatura, sendo que apenas as utilizadas direta ou indiretamente neste trabalho, a saber, os algoritmos *k-NN*, *C4.5*, *RI-PPER* e *SVMs*, foram descritas de modo detalhado.

No próximo capítulo são apresentados conceitos sobre detecção de ruídos, assim como as técnicas de detecção e eliminação de ruído (baseadas no algoritmo *k-NN*) avaliadas neste trabalho.

Pré-processamento de Dados

3.1 *Considerações Iniciais*

Dados são geralmente coletados por meio de medições realizadas em um domínio de interesse, associando o valor de uma variável com uma determinada propriedade existente no ambiente ou problema em estudo. As relações existentes entre os diversos elementos do problema em análise são representadas por relacionamentos numéricos entre as variáveis existentes no conjunto de dados.

Nesse sentido, nenhum conjunto de dados é perfeito, erros de medições, dados incompletos, amostras distorcidas, falhas humanas ou dos equipamentos utilizados, dentre muitos outros fatores, contribuem para a contaminação dos dados, e isso é particularmente verdadeiro para conjuntos que apresentam dados com elevada dimensionalidade.

Conjuntos de dados de elevada dimensionalidade possuem maior probabilidade de apresentar uma maior gama de problemas. O principal deles diz respeito à dificuldade de manipulação e entendimento dos dados, intrínseco às características desses conjuntos, que inviabilizam uma análise por especialistas ou mesmo pela utilização de técnicas de visualização que facilitem a interpretação dos dados. Além desses problemas, outros advêm do modo com que os dados são coletados e armazenados nesses conjuntos. Problemas com os dados coletados geralmente têm como causa a existência de diferentes fontes de ruídos,

como dados incompletos, errados ou corrompidos, que distorcem o problema em análise e dificultam ainda mais a indução de hipóteses por algoritmos de Aprendizado de Máquina (AM).

Dados de expressão gênica geralmente se caracterizam pela presença de ruídos. A própria natureza dos experimentos laboratoriais realizados na área de Biologia é afetada pela ocorrência de erros de diversas categorias. Exemplos comuns são a presença de contaminações nas amostras laboratoriais ou erros na calibragem dos equipamentos utilizados nos experimentos.

Ruídos presentes em conjuntos de dados podem reduzir o desempenho da técnica de AM e aumentar o tempo de construção da hipótese induzida pelo classificador, assim como o tamanho final desse classificador (ZHU e WU [2004]).

Embora grande parte das técnicas de AM apresentem uma robustez diante de dados ruidosos, a presença destes exemplos pode afetar o processo de indução da hipótese pelo classificador (LAVRAC e GAMBERGER [2001]). Isso pode ser devido à ocorrência de *overfitting*, uma vez que um algoritmo de AM, quando confrontado com um conjunto de dados ruidoso, pode gerar uma hipótese muito complexa e específica, na tentativa de acomodar a presença dos ruídos, que não se ajuste realmente à realidade das amostras existentes no conjunto de dados (GAMBERGER et al. [1999]). Uma hipótese induzida a partir de um conjunto de dados livre de ruídos possui maior probabilidade de ser menos complexa, além de ser mais acurada na classificação de dados desconhecidos.

A qualidade de um conjunto de dados pode ser determinada basicamente por fatores internos e externos. O fator interno indica se os atributos e as classes foram bem selecionados e definidos para caracterizar esse conjunto, ao passo que o externo se refere a erros adicionados aos atributos e classes, seja de modo artificial e voluntário ou àqueles introduzidos de maneira acidental.

Nesse sentido, em AM é possível caracterizar ruídos em dois grupos: ruídos presentes nos atributos e nas classes. O primeiro é representado por erros que são introduzidos nos valores dos atributos. O segundo, por erros nas classes, seja por um mesmo dado com mais de uma classificação ou por dados que são rotulados por classes incorretas (ZHU e WU [2004]).

Diferentes são os tipos de ruídos que podem ser encontrados nos conjuntos de dados, especialmente aqueles que manipulam dados reais, como os de Bioinformática (Figura 3.1 - (LORENA e CARVALHO [2004])):

- *Dados com rótulo incorreto*: Compreendem dados incorretamente classificados, cujos padrões são considerados ruídos;
- *Dados redundantes*: Exemplos que podem ser representadas por outros já existentes no conjunto de treinamento. Estes dados geralmente formam agrupamentos no conjunto de dados. A remoção de dados redundantes pode ser vantajosa no balanceamento do número de exemplos pertencentes a diferentes classes em conjuntos de dados desbalanceados;
- *Outliers*: Exemplos muito distintos dos demais existentes no conjunto de dados. Estes exemplos podem ser ruídos ou casos particulares e sua influência na indução da fronteira de decisão deve ser minimizada;
- *Dados próximos à borda de decisão*: Esses dados podem ser considerados relativamente inseguros, já que uma pequena quantidade de ruído pode movê-los para o lado incorreto da fronteira de decisão. A sua remoção pode suavizar a borda de decisão, possibilitando uma redução no efeito de *overfitting*;
- *Dados "seguros"*: Demais exemplos existentes no conjunto de dados. Estes são os dados mais típicos de cada classe, devendo ser mantidos no treinamento dos algoritmos de AM. Deve-se salientar que dados redundantes também podem ser considerados seguros, porém a remoção de alguns deles não necessariamente propicia prejuízos à generalização do classificador, uma vez que os dados pertencentes a um mesmo agrupamento redundante representam informações similares.

Na próxima seção são apresentadas algumas das técnicas de detecção de ruído existentes na literatura.

3.2 Detecção de Ruídos

Existem diversas técnicas de pré-processamento que podem ser aplicadas na detecção e remoção de ruídos. Modelos estatísticos foram os primeiros utilizados na detecção de ruídos, sendo que os primeiros modelos desenvolvidos eram aplicáveis somente a conjuntos de dados unidimensionais (HODGE e AUSTIN [2004]).

Em Estatística, o problema de detecção de ruídos é comumente solucionado por meio de técnicas baseadas em modelos da distribuição dos dados (BARNETT e LEWIS [1994]), em que os ruídos são identificados como observações que diferem de uma distribuição

Figura 3.1: Tipos de ruído encontrados em conjuntos de dados: (a) Exemplo de conjunto de dados; (b) Dados com rótulos incorretos; (c) Dados redundantes; (d) Outliers; (e) Dados próximos da borda de decisão; (f) Dados "seguros" (LORENA e CARVALHO [2004]).

utilizada na modelagem dos dados. O maior problema dessa abordagem está em assumir que a distribuição dos dados é previamente conhecida, fato que não reflete a realidade em grande parte das aplicações reais.

Modelos estatísticos são adequados para conjuntos de dados numéricos ou para aqueles em que os dados podem facilmente ser convertidos para representações numéricas. Isso limita sua aplicabilidade e pode elevar muito o tempo de processamento se forem aplicados a dados que necessitem de conversões complexas de modo a viabilizar sua utilização por esses modelos.

Técnicas de agrupamento de dados (clusterização) também vêm sendo aplicadas na identificação de ruídos (CORNEY [2002]; HE e XU [2002]; HE et al. [2003]; JIN et al. [2001]). Em HE et al. [2003], esta estratégia é empregada na identificação de ruídos, em que pequenos agrupamentos de dados são considerados possíveis ruídos. Uma medida baseada nesta observação é determinada por cada autor em seu respectivo trabalho, de modo a determinar o grau com que cada exemplo pode ser considerado ou não um ruído.

De acordo com outra linha de pesquisa, podem ser encontrados diversos trabalhos que exploram o uso de técnicas de AM na obtenção de filtros para ruídos (JOHN [1995]; BRODLEY e FRIEDL [1996]; STOLFO et. al. [1997]; GAMBERGER e LAVRAC [1997];

BRODLEY e FRIEDL [1999]; GAMBERGER et al. [1999]; AGGARWAL e YU [2001]; HAWKINS et al. [2002]; VERBAETEN [2002]; VERBAETEN e ASSCHE [2003]; KHOSHGOFTAAR e REBOURS [2004]).

Em HAWKINS et al. [2002], por exemplo, uma RN auto-associativa é utilizada na detecção desses exemplos. RNs auto-associativas são modelos neurais que possuem a camada de saída com mesmo tamanho que a de entrada de forma a reproduzir cada exemplo apresentado em seu treinamento. Seguindo esta metodologia, os dados para os quais a RN comete maiores erros na reprodução podem ser considerados ruídos.

Já VERBAETEN [2002] aborda o problema de identificação e remoção de ruídos (ou pré-processamento) do conjunto de treinamento, mesmo método utilizado neste projeto. De acordo com GAMBERGER et al. [2000]; ZHU e WU [2004], a eliminação de ruídos antes da análise dos dados por técnicas da AM, ou seja, antes da determinação da hipótese induzida, possui a vantagem de que os exemplares ruidosos não influenciam na construção dessa hipótese pelo algoritmo de AM.

Para a detecção e eliminação de ruídos, VERBAETEN [2002] utilizou o algoritmo *Robust Tilde* (JOHN [1995]), uma versão do algoritmo C4.5 que trabalha com Árvores de Decisão (ADs) robustas, valendo-se também de filtros que seguem a abordagem de consenso ou votação por maioria.

Em AGGARWAL e YU [2001] é utilizado um Algoritmo Genético (AG) para resolver o problema de detecção de ruídos em conjuntos de dados com elevada dimensionalidade. Neste trabalho, projeções de menor dimensionalidade que representam o conjunto de dados original são determinadas e o AG é responsável por encontrar possíveis dados ruidosos.

No trabalho de GAMBERGER et al. [1999], o objetivo foi a remoção de ruídos do conjunto de dados original antes da apresentação dos dados aos classificadores utilizados, mesma metodologia utilizada neste trabalho. Entretanto, os autores do primeiro trabalho investigaram técnicas diferentes para detecção e remoção de exemplares ruidosos. Os autores utilizaram-se de quatro tipos diferentes de filtros de ruído em seus experimentos, sendo denominados de filtro de saturação, filtro de classificação, filtro combinado de classificação-saturação, que é a utilização em conjunto dos dois filtros já mencionados, e filtro de saturação por consenso, que é a utilização do filtro de saturação adicionado a um método de classificação por consenso, em que um dado somente é considerado ruído se todos os classificadores utilizados o classificarem como tal.

Em BRODLEY e FRIEDL [1996], a mesma metodologia de remoção de ruídos utilizada em GAMBERGER et al. [1999] é empregada, com o objetivo de obter um conjunto

de treinamento melhorado após a remoção dos dados ruidosos. Nesse trabalho uma combinação de classificadores foi utilizada como filtro de ruídos, sendo que todos os exemplos considerados ruídos pelos n classificadores da combinação são removidos, ou seja, foi utilizada a metodologia de eliminação por consenso. As técnicas de AM utilizadas nesse filtro foram o algoritmo C4.5 (QUINLAN [1993]), o algoritmo k -NN com o valor de $k=1$ e uma função linear discriminante (NILSSON [1965]). Como classificador final, foi utilizada uma combinação de classificadores segundo a metodologia de votação por consenso.

Outros trabalhos que também se valem de combinação de classificadores para identificação e remoção de ruídos do conjunto de dados original são os de KHOSHGOFTAAR e REBOURS [2004] e VERBAETEN e ASSCHE [2003]. Nesses dois trabalhos, o conjunto de dados inicial é pré-processado por técnicas de remoção de ruídos e somente é apresentado ao classificador final o conjunto de dados reduzido, do qual os dados ruidosos foram previamente eliminados.

VERBAETEN e ASSCHE [2003] utiliza cinco diferentes filtros de ruído, baseados na metodologia de combinação classificadores, que são descritos a seguir:

- Comitês do método *cross-validation*, valendo-se dos métodos de combinação por consenso e votação por maioria;
- Algoritmo de *bagging*, também se valendo dos métodos de combinação por consenso e por votação por maioria; e
- Algoritmo de *boosting Adaboost* (FREUND e SCHAPIRE [1996]), em que são removidos os dados que obtiveram peso elevado.

Para todos os filtros desenvolvidos, o classificador utilizado foi induzido pelo algoritmo *Tilde* (BLOCKEEL e RAEDT [1998]).

Já em KHOSHGOFTAAR e REBOURS [2004], o autor desenvolve um filtro de ruídos que implementa diferentes filtros comumente encontrados na literatura, bastando apenas determinar um conjunto de parâmetros de modo a especializá-lo em um filtro de classificação; um de *ensembles*; um multi-particionado ou um iterativamente-particionado.

O filtro de classificação considera como ruído os dados que são incorretamente classificados por um classificador, que utiliza o método *cross-validation* em seu treinamento. Já o filtro de *ensembles* é idêntico ao filtro de classificação, porém combinando o resultado de m classificadores diferentes. Esse filtro pode seguir a metodologia de consenso ou votação por maioria.

O filtro multi-particionado segue a idéia do filtro de particionamento descrito em ZHU et. al. [2003]. Entretanto, ele combina os resultados de cinco diferentes classificadores induzidos em cada partição do conjunto de dados existente, ao passo que o filtro iterativamente-particionado produz apenas um modelo para cada sub-conjunto do conjunto de dados original, mas itera o processo de filtragem dos dados até que uma determinada condição seja obtida.

É importante notar que todos os filtros apresentados são baseados na idéia de que quanto mais um exemplo é incorretamente classificado pelos classificadores que o avaliam, maior a probabilidade dele ser um ruído.

Diversos trabalhos, em que a identificação de ruídos é realizada considerando a distância desses a outros dados no conjunto de dados podem ser encontrados na literatura (WILSON [1972]; TOMÉK [1976a;b]; WETTSCHERECK [1994]; KNORR e NG [1998]; KNORR et al. [2000]; RAMASWAMY et. al. [2000]; TANG et. al. [2002]). Neste trabalho utilizou-se representantes desta classe, os quais são descritos em maiores detalhes na Seção 3.3.

Essas técnicas se destacam pela simplicidade de implementação e por não fazerem suposições, a priori, sobre o modelo de distribuição dos dados investigados. Um de seus problemas está no fato de consumir elevada memória e tempo na computação das distâncias entre todos os exemplos analisados. Nesse sentido, a complexidade computacional é diretamente proporcional à dimensionalidade dos dados e ao número de registros existente (HODGE e AUSTIN [2004]).

É possível notar que, na área de detecção de ruídos, os autores cada vez mais se valem de sistemas híbridos e de combinação de classificadores com o objetivo de diminuir as deficiências dos algoritmos envolvidos e, conseqüentemente, melhorar o desempenho final obtido.

3.3 Detecção de Ruídos baseado em distância

Algoritmos baseados em distância como, por exemplo, aqueles desenvolvidos com base no algoritmo *k-nearest neighbour* (*k-NN*), que foram os selecionados para avaliação nesse trabalho, assim como algoritmos de agrupamento, dentre outros, baseiam-se em medidas de similaridade existentes entre os dados analisados.

Alguns dos mais conhecidos algoritmos baseados em distância encontrados na literatura, assim como aqueles selecionados para realização deste trabalho, são apresentados de

modo mais detalhado a seguir.

Para todos os algoritmos apresentados, seja T o conjunto de treinamento original e S o subconjunto de T , obtido com a aplicação de uma das técnicas de eliminação de ruídos baseadas em distância, supor que T contém n dados $\mathbf{x}_1, \dots, \mathbf{x}_n$. Cada dado $\mathbf{x}$ de T (e também de S) possui k vizinhos mais próximos.

Define-se como "inimigo mais próximo" de $\mathbf{x}$ o exemplo de T (ou S), com classe oposta à de $\mathbf{x}$, mais próximo a este dado. Exemplos que têm $\mathbf{x}$ como um de seus k vizinhos mais próximos são denominados associados de $\mathbf{x}$. Para encontrar os vizinhos mais próximos a um determinado exemplo e realizar classificações eventualmente necessárias, todas as técnicas de detecção de ruído consideradas utilizam como base o algoritmo k -NN. Este algoritmo determina a classificação de um dado de acordo com a classe da maioria de seus k vizinhos mais próximos.

Outra consideração importante na descrição dos algoritmos apresentados diz respeito à definição da métrica a ser utilizada na computação das distâncias entre os dados. É demonstrado que, para espaços com grandes dimensões, várias métricas, como a Euclidiana, são pouco significativas (BEYER et. al. [1999]; AGGARWAL et. al. [2001]; HINNEBURG et. al. [2000]). Isto se deve ao fato de que os dados tendem a ser esparsos. Como os dados de expressão gênica avaliados neste trabalho possuem, em geral, alta dimensionalidade e apresentam atributos nominais e contínuos, WILSON e MARTINEZ [1997a] sugerem o uso da métrica de distância denominada *HVDM* (*Heterogeneous Value Difference Metric*) para lidar com tais características. Esta métrica se baseia na distribuição dos atributos por classe em todo o conjunto de dados, e não apenas em valores pontuais, como ocorre com a medida Euclidiana. A métrica *HVDM* é apresentada pela Equação 3.1.

$$HVDM(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_{a=1}^m (d_a^2(x_a, z_a))} \quad (3.1)$$

em que $\mathbf{x}$ e $\mathbf{z}$ são dois exemplos com m atributos. A função $d_a(x, z)$, que mede a distância entre os atributos de $\mathbf{x}$ e $\mathbf{z}$ é dada pela Equação 3.2.

$$d_a(x_a, z_a) = \begin{cases} 1, & \text{se } x_a \text{ ou } z_a \text{ é desconhecido;} \\ VDM_a(x_a, z_a), & \text{se } a \text{ é nominal;} \\ \frac{|x_a - z_a|}{\rho_a}, & \text{se } a \text{ é numérico.} \end{cases} \quad (3.2)$$

em que $VDM_a(x_a, z_a)$ é a distância VDM (*Value Difference Metric*) (STANFILL e WALTZ [1986]), definida pela Equação 3.3, e ρ_a é o desvio padrão do atributo a no conjunto de dados.

$$VDM_a(x_a, z_a) = \sum_{i=1}^c \left| \frac{C_{xa_i}}{C_{xa}} - \frac{C_{za_i}}{C_{za}} \right|^l \quad (3.3)$$

Na Equação 3.3, c é o número de classes presentes em um conjunto de dados, C_{xa_i} é o número de dados da classe i com valor de atributo x_a . C_{xa} é o número total de exemplos com atributo igual a x_a e l é uma constante, freqüentemente igual a 1 ou 2. Pela interpretação dessa equação, tem-se que o VDM considera dois valores como similares se eles ocorrem com freqüência relativa aproximada em todas as classes.

A partir dessas definições, seguem descrições de diversas técnicas de eliminação de ruídos baseadas em medidas de distância encontradas na literatura.

A regra *Edited Nearest Neighbour (ENN)* foi proposta por WILSON [1972]. De acordo com o algoritmo ENN, tem-se inicialmente $S = T$. Um exemplo do conjunto de dados S é removido se sua classe não é igual à da maioria de seus k vizinhos mais próximos, em que k tipicamente é igual a 3. Este algoritmo elimina ruídos e dados próximos à borda de decisão, utilizada na separação dos dados em classes. Dados de borda podem ser considerados relativamente inseguros, uma vez que mesmo uma pequena quantidade de ruídos pode movê-los para o lado incorreto da borda de decisão. Sua remoção pode também suavizar a fronteira de separação, possibilitando redução no efeito de *overfitting* na posterior aplicação de técnicas de AM. Essa técnica também se caracteriza por manter todos os pontos internos do conjunto de dados (por reter dados redundantes).

Existe uma variante do algoritmo ENN, denominada *Repeated ENN (RENN)*, em que o algoritmo ENN é aplicado repetidamente ao conjunto de dados até que todos os exemplos remanescentes tenham a maioria de seus vizinhos pertencentes à mesma classe. Isso permite que as classes sejam mais bem definidas (aumenta a margem de separação entre as classes) e também suaviza (define) a fronteira de decisão, garantindo uma melhor generalização.

TOMEK [1976b] propôs também outra extensão do algoritmo ENN, denominada *All-KNN*, que pode ser aplicada na detecção e remoção de ruídos e dados da borda. Esse algoritmo procede da seguinte maneira: para $i = (1, \dots, k)$, marcar como "indesejado" qualquer exemplo classificado incorretamente por seus i vizinhos mais próximos. Depois de completos os ciclos, remover todos os exemplos previamente marcados. Essa técnica

pode deixar os pontos internos inalterados, e proporciona pouca redução no conjunto de dados original, assim como as técnicas ENN e RENN. Essas três técnicas (ENN, RENN e AllKNN) são conhecidas, de acordo com WILSON e MARTINEZ [2000], como filtros de ruídos.

LOWE [1995] propôs o algoritmo *Variable Similarity Metric (VSM)*, que é capaz de garantir uma boa generalização para os dados analisados. Para garantir a eliminação de ruídos de modo a evitar perda da qualidade de generalização, um exemplo somente é removido se todos os seus k vizinhos pertencem a uma mesma classe, mesmo que a classe de seus k vizinhos seja diferente da classe do exemplo analisado. Desse modo, não apenas dados ruidosos são eliminados, mas também pontos internos, ao passo que pontos das bordas são mantidos. Porém, um dado somente é removido se seus vizinhos estão pelo menos 60% certos de sua classificação. Esse algoritmo tipicamente apresenta um valor elevado de k (superior a 10), pois utiliza o método de votação baseado nos pesos das distâncias (*distance-weighted voting*).

Outro algoritmo existente, o *Reduced Nearest Neighbour Rule (RNN)*, foi proposto por GATES [1972]. Esse algoritmo inicia sua execução considerando $S = T$, e remove um exemplo de S se a remoção não fizer com que quaisquer outros exemplos existentes em T sejam erroneamente classificados pelos exemplos remanescentes em S . Esse algoritmo elimina ruídos e dados internos, enquanto mantém pontos localizados nas bordas de decisão.

Uma outra alternativa, o algoritmo *IB3 (Instance Based Learning Algorithm 3)*, foi proposto por AHA et. al. [1991; 1992], que investigou diversos algoritmos baseados em instâncias. O *IB3* é um algoritmo incremental que possui a vantagem de considerar os dados que são incluídos no conjunto de dados após o término do treinamento. Contudo, tem a desvantagem de ser sensível à ordem de apresentação desses dados. Ele é uma melhoria do algoritmo *IB2*, uma vez que elimina dados ruidosos utilizando, para isso, uma métrica que mantém apenas dados classificados erroneamente que são considerados "aceitáveis".

Os autores AHA et. al. [1991; 1992] propuseram ainda dois outros algoritmos, denominados *IB4* e *IB5*, que são capazes de eliminar ruídos de conjuntos de dados. O algoritmo *IB4* é uma extensão do algoritmo *IB3* que lida com atributos irrelevantes, construindo, para isso, um conjunto de pesos para os atributos pertencentes a cada classe. Já o algoritmo *IB5* é uma extensão do algoritmo *IB4* de modo a permitir a adição de novos dados ao problema analisado depois que o treinamento do algoritmo já tenha sido iniciado.

ZHANG [1992] propôs um algoritmo baseado em instâncias denominado *Typical Instance Based Learning (TIBL)*, que objetiva manter dados localizados próximos aos centros dos clusters em detrimento dos localizados próximos às bordas de decisão. Isso garante robustez frente aos ruídos e suaviza a fronteira de decisão. Inicialmente, esse algoritmo seleciona um exemplo $\mathbf{x}$, existente em $T - S$, que é incorretamente classificado pelos dados presentes em S . Em seguida, seleciona um dado $\mathbf{y}$ existente em $T - S$ que faz com que $\mathbf{x}$ seja corretamente classificado, e o adiciona a S , definindo um peso a esse dado $\mathbf{y}$, de acordo com a relação entre sua similaridade média com os dados de mesma classe e sua similaridade média com os exemplos de outras classes. Esse cálculo é realizado de acordo com uma função específica desse algoritmo. Esse processo é repetido até que todos os dados em T sejam corretamente classificados. Durante esse processo, o algoritmo *TIBL* evita manter dados considerados ruidosos no novo conjunto de dados por ele produzido. Esse é um algoritmo pseudo incremental, pois é executado de modo incremental, mas utiliza todo o conjunto de treinamento no cálculo dos pesos dos dados.

Já WILSON e MARTINEZ [2000] propuseram um conjunto de algoritmos de eliminação de ruídos baseados em distância que visam apresentar tolerância aos ruídos, elevada taxa de acerto na generalização, não sensibilidade à ordem em que os dados são apresentados e alta eliminação de ruídos com conseqüente redução do conjunto de dados original. Esses algoritmos, descritos a seguir, são denominados *Decremental Reduction Optimization Procedures*, que incluem cinco variações dessa categoria de algoritmos (*DROPs 1 a 5*) e o algoritmo *Decremental Encoding Length*. Eles apresentam uma etapa de identificação e remoção de ruídos e uma etapa de detecção e remoção de dados redundantes presentes no conjunto de dados analisado. Nesse sentido, o conjunto de dados resultante será constituído apenas de exemplos que não sejam considerados redundantes ou ruídos pelo algoritmo avaliado.

- *Decremental Reduction Optimization Procedure 1 (DROP1)*: Inicialmente, tem-se $S = T$ e constrói-se uma lista de vizinhos mais próximos para cada dado, assim como uma lista de associados. Um exemplo $\mathbf{x}$ é removido de S se a remoção não afetar a classificação da maioria dos exemplos associados a $\mathbf{x}$ presentes em S . Esse algoritmo é capaz de remover ruídos e centros de agrupamentos (que podem ser considerados dados redundantes), mantendo a maioria dos pontos localizados nas bordas. Seu método de remoção dos dados pode prejudicar a generalização do classificador, uma vez que muitas vezes agrupamentos inteiros são removidos.
- *DROP2*: Eventualmente, o algoritmo DROP1 pode não remover todo o ruído pre-

sente nos dados. Se os vizinhos de um dado ruidoso são eliminados antes dele, seus associados podem vir a incluir dados da mesma classe do ruído, resultando que a eliminação deste leve à classificação incorreta de seus associados. Com isso, o exemplo ruidoso é mantido.

Para lidar com esta situação, o algoritmo DROP2 considera o efeito da remoção de um determinado dado em todos os exemplos existentes no conjunto original T , e não somente nos dados existentes no subconjunto S , proveniente da eliminação de ruídos, apenas eliminando um dado $\mathbf{x}$ de T se a maioria de seus associados em S (não mais em T) são classificados corretamente sem $\mathbf{x}$. Também há uma mudança na ordem em que os dados são removidos. Primeiro, os exemplos em T são ordenados pela distância em relação a seu inimigo mais próximo. A ordem de checagem para a remoção é então dada pelos exemplos mais afastados de seus inimigos mais próximos. Nesse sentido, é evitada a remoção de agrupamentos inteiros e a tendência é remover os dados mais centrais, mantendo maior quantidade de dados pertencentes às bordas.

- *DROP3*: De acordo com o algoritmo DROP2, pontos centrais são removidos antes de pontos de borda. Com isto, pontos de borda ruidosos podem ser mantidos. O algoritmo DROP3 lida com essa situação realizando um passo de filtragem de ruídos antes de ordenar os dados. Esse processo de filtragem consiste em uma regra similar à do algoritmo ENN: qualquer dado classificado incorretamente por seus k vizinhos mais próximos é removido. Desse modo, a regra tende a remover primeiro os dados internos dos agrupamentos e os ruídos e dados localizados próximos das bordas, definindo-a melhor. Possui um drástico problema de redução, que muitas vezes elimina todos ou uma grande quantidade de dados existentes no conjunto de dados original.
- *DROP4*: Esse algoritmo é semelhante ao algoritmo DROP3, porém não aplica indiscriminadamente o filtro de ruído ENN. O passo de filtragem de ruídos somente remove um dado se este é classificado incorretamente por seus k vizinhos mais próximos e se sua remoção não prejudicar a classificação de outros dados. Essa regra corrige o problema de remoção de agrupamentos inteiros e melhora a generalização, com o custo de menor redução do conjunto de dados original.
- *DROP5*: Esse algoritmo modifica o algoritmo DROP2, de modo que a ordem para remoção é iniciada com os dados mais próximos de seu inimigo mais próximo e assim segue. Esse procedimento funciona como um passo na redução de ruídos. Pontos de borda também são removidos primeiro, o que suaviza a fronteira de decisão.

Depois desse passo, o algoritmo convencional DROP2 é aplicado até que não seja mais possível obter melhorias na generalização.

- *Decremental Encoding Length (DEL)*: Pode ser visto como uma versão decremental do algoritmo ELH, sendo similar ao algoritmo DROP3, com exceção do fato de também empregar uma heurística na decisão sobre quais dados remover na etapa de filtragem de ruídos. Essa heurística baseia-se em um cálculo do custo acarretado pela remoção do dado. Nesse sentido, um dado somente é removido se for classificado erroneamente por seus k vizinhos mais próximos e se sua remoção não provocar aumento no custo da função heurística utilizada. Após essa fase, os dados resultantes são sorteados de acordo com a distância que possuem de seu inimigo mais próximo. Enquanto for possível obter melhorias na generalização, esses dados são removidos (começando a partir daquele que esteja mais distante de seu inimigo mais próximo), desde que essa remoção não acarrete em aumento no custo da função heurística utilizada.

O algoritmo *ELH (Encoding Length)* (JANKOWSKI e GROCHOWSKI [2004]), a partir do qual o algoritmo *DEL* foi implementado, faz parte de um grupo de algoritmos que possuem uma heurística de codificação de tamanho que serve como base para determinação de quais dados são ruídos e devem, portanto, ser eliminados. Essa heurística é baseada no cálculo do custo acarretado pela remoção de um determinado exemplo. Trata-se de um algoritmo incremental, que inicia sua execução a partir de um conjunto de dados T vazio, e adiciona dados a T (que será o conjunto reduzido de dados) somente se o dado adicionado não acarretar em aumento no custo da função heurística utilizada.

3.4 Combinação de Técnicas de Detecção de Ruído

Diferentes classificadores, com base em um mesmo conjunto de treinamento, seguem diferentes princípios de funcionamento e possuem diferentes capacidades de generalização. A abordagem tradicional para se obter o melhor classificador é selecionar aquele que oferece o melhor resultado para o problema que está sendo analisado. Contudo, essa pode não ser a opção mais adequada, uma vez que informações valiosas descobertas pelos demais classificadores podem estar sendo ignoradas. Sendo assim, uma abordagem alternativa é combinar, de algum modo, as saídas de todos os classificadores (TUMER e GOSH [1996]).

Nesse sentido, para o desenvolvimento de combinações de técnicas de detecção de ruído

deve-se exaltar as boas características de cada técnica empregada e reduzir as falhas. Não existem vantagens na combinação para a solução de problemas para os quais já existe um algoritmo específico capaz de solucioná-los satisfatoriamente. Uma das principais desvantagens da combinação é a possibilidade de aumento da complexidade do problema.

De acordo com PRAMPERO e CARVALHO [1999], o desenvolvimento de uma combinação tem como objetivo resolver problemas mais gerais, que envolvem diferentes tipos de dados, ou aqueles que não são resolvidos satisfatoriamente por uma única técnica e/ou em que o desempenho apresentado pelo módulo de classificação deva ser estável.

A escolha de quantas e quais técnicas serão combinadas depende do problema em análise. Para que a combinação seja bem sucedida é aconselhável utilizar técnicas de diferentes características e que possam se complementar, ou seja, a ênfase da combinação deve estar na exploração da similaridade ou da diferença de cada uma das técnicas envolvidas. Da similaridade porque oferece mais garantias de que a decisão indicada por uma técnica foi correta, da diferença porque dados classificados incorretamente por uma técnica podem ser classificados corretamente por outra(s).

Uma combinação pode ser feita paralelamente, em cascata ou de forma hierárquica (PRAMPERO [1998]). Em um sistema em cascata, os algoritmos são dispostos de maneira seqüencial em fila, sendo que as saídas do anterior servem como entrada para o seguinte. Em sistemas paralelos, os algoritmos trabalham independentemente e o processo de decisão integra os resultados obtidos (somente suas saídas são combinadas para serem apresentadas a um classificador final). Já em sistemas hierárquicos, a estratégia de controle é uma mistura entre processamento em cascata e em paralelo.

Uma combinação deve ser feita de maneira apropriada, de modo que seja tão simples quanto possível e com um mínimo de redundância, uma vez que módulos desnecessários ocupam recursos, aumentam o tempo de processamento e a complexidade do sistema (HODGE e AUSTIN [2004]).

3.5 *Considerações Finais*

Uma grande quantidade de conjuntos de dados, principalmente aqueles relacionados a problemas reais, apresentam ruídos em sua composição. Dados de expressão gênica, utilizados neste trabalho, são um exemplo clássico de conjuntos de dados que apresentam não apenas ruído, mas também muitas vezes elevada dimensionalidade, fator que dificulta a análise desses por técnicas de AM.

Neste capítulo foram apresentados os conceitos básicos sobre ruídos em conjuntos de dados e os problemas que esses acarretam a um classificador durante a modelagem da hipótese induzida, no intuito de resolver o problema analisado de maneira satisfatória.

Alguns dos trabalhos existentes na literatura, que abordam o tema analisado, foram apresentados, assim como algumas técnicas de detecção e eliminação de ruído baseadas em distância foram explicadas de maneira mais detalhada, uma vez que essa é a abordagem sobre ruídos empregada neste trabalho. A combinação de técnicas de detecção de ruído também foi abordada, apresentando os conceitos básicos para análise e desenvolvimento dos módulos de combinação.

Experimentos Iniciais

4.1 *Considerações Iniciais*

Este capítulo apresenta os experimentos iniciais realizados neste trabalho de Mestrado, assim como os resultados obtidos. Na Seção 4.2 é descrita a proposta desse trabalho. Na Seção 4.3 é apresentada a metodologia empregada na realização dos experimentos. A Seção 4.4, apresenta os resultados inicialmente obtidos, ao passo que a Seção 4.5 discute aspectos observados nos experimentos realizados e a motivação para a realização de uma nova etapa de experimentos.

4.2 *Descrição da Proposta*

Neste trabalho, a identificação e eliminação de ruídos em conjuntos de dados é analisada. Para esse fim, foram utilizados algoritmos de detecção de ruído baseados em distância.

Ruídos presentes em conjuntos de dados prejudicam o desempenho obtido pelos classificadores que objetivam analisar o problema envolvido, assim como tornam a análise destes mais complexa. Nesse âmbito, a remoção de ruídos dos conjuntos de dados em uma etapa de pré-processamento, antes de sua análise pelos classificadores selecionados, tende a melhorar o esforço computacional, assim como os resultados obtidos.

Os conjuntos inicialmente selecionados são, em sua maioria, amplamente utilizados em trabalhos encontrados na literatura, sendo a maioria deles pertencente ao repositório da Universidade da Califórnia em Irvine (UCI) (BLAKE e MERZ [1998]) de Aprendizado de Máquina (AM). Esses são descritas em detalhes nas seções subseqüentes.

Algoritmos de detecção de ruído baseados em distância foram selecionados e aplicados aos conjuntos de dados utilizados, com o objetivo de avaliar se a etapa de eliminação de ruídos melhora o desempenho dos classificadores de AM analisados e também reduz a complexidade da hipótese induzida.

Foram então desenvolvidas combinações das técnicas de detecção de ruído utilizadas, visando obter resultados superiores aos previamente obtidos pelas mesmas quando avaliadas individualmente. Essas combinações foram compostas pelas técnicas de detecção de ruído que apresentaram melhor desempenho para cada um dos conjuntos de dados investigados, explorando a similaridade ou diferença existente entre as mesmas e se elas se complementam de modo a obter um melhor desempenho.

4.3 Metodologia

Os algoritmos para a detecção de ruído utilizados neste trabalho são aqueles na qual a identificação dos ruídos se dá considerando a distância entre exemplos que podem representar provável ruído e os demais existentes no conjunto de dados. Nesse sentido, foram utilizados os seguintes algoritmos de detecção de ruído: *Edited Nearest Neighbor (ENN)*, *Repeated ENN (RENN)*, *AllKNN*, *Decremental Reduction Optimization Procedures 1 to 5 (DROPs 1 a 5)* e *Decremental Encoding Length (DEL)*. Os algoritmos DEL e DROPs 1 a 5 se caracterizam por detectar não apenas ruídos mas também dados redundantes presentes nos conjuntos de dados analisados.

Uma característica desses algoritmos é a possibilidade de definir o valor do parâmetro k a ser utilizado, ou seja, o número de exemplos considerados no cálculo da distância entre um candidato a ruído e seus vizinhos mais próximos. Como o valor *default* das técnicas utilizadas é $k=3$, todos os experimentos foram realizados com os valores de k como 1, 3 ou 9, de modo a seguir uma progressão geométrica que inclua o número 3, o valor *default* dos algoritmos utilizados.

Os conjuntos de dados previamente selecionados (conjuntos originais - antes da etapa de eliminação de ruídos) e os conjuntos de dados pré-processados, ou seja, já eliminados os dados considerados ruídos por cada uma das técnicas de detecção de ruído utilizadas,

foram submetidos a classificadores induzidos pelos algoritmos *Support Vector Machines* (SVMs), C4.5, gerador de Árvores de Decisão (ADs), e RIPPER, de modo a terem seu efeito analisado.

Em seguida, para os conjuntos de dados que apresentaram resultados superiores para os dados pré-processados, foram desenvolvidas combinações das técnicas de detecção de ruído anteriormente avaliadas. O objetivo foi analisar se a combinação das diferentes técnicas poderia resultar em desempenho superior à aplicação delas individualmente.

Em uma combinação por consenso, todas as n técnicas empregadas devem classificar determinado dado como ruído para que ele seja eliminado do conjunto de dados. Na combinação de votação por maioria, m técnicas, dentre n investigadas, devem classificar o dado como ruído para que esse seja eliminado, em que m deve ser um número maior que $n/2$.

Quanto maior o valor atribuído a m , mais conservador se torna o método, uma vez que mais técnicas devem classificar um dado como ruído para que ele possa ser eliminado. Nesse sentido, utilizar combinações por consenso em vez de combinações de votação por maioria resultará em uma combinação mais conservadora, em que menos dados serão eliminados do conjunto de dados final.

De acordo com BRODLEY e FRIEDL [1999], combinações por consenso evitam que dados considerados não ruidosos sejam eliminados, com o prejuízo de que uma maior quantidade de dados ruidosos serão conservados, ao passo que combinações de votação por maioria são melhores em detectar ruídos, mas correm o risco de eliminar também dados não ruidosos.

Uma vez que manter dados ruidosos pode degradar o desempenho final da combinação, o método de votação por maioria se torna, muitas vezes, mais adequado, uma vez que mais dados ruidosos podem ser detectados e eliminados do conjunto de dados final, resultante da combinação desenvolvida (BRODLEY e FRIEDL [1999]).

4.3.1 Conjuntos de Dados

Para a realização dos experimentos, cinco conjuntos de dados foram inicialmente selecionados:

- *Splice* (BLAKE e MERZ [1998]) do repositório UCI;
- *Ionosphere* (SIGILLITO et. al. [1989]) do repositório UCI;

- *Wisconsin Breast Cancer* (WOLBERG et al. [1995]) do repositório UCI;
- *Expressão Gênica* disponibilizada por BROWN et al. [1999]; e
- *Golub* (GOLUB et al. [1999]).

O objetivo de utilizar conjuntos de dados do repositório UCI está no fato de eles serem amplamente utilizados em diversos trabalhos encontrados na literatura, facilitando assim a comparação dos resultados obtidos nos experimentos realizados.

Nesse sentido, o conjunto *Splice* apresenta seqüências de nucleotídeos (A, T, C, G) de tamanho fixo para a análise do problema de reconhecimento de junções de *splice*. Os genes dos organismos eucariotos são compostos por segmentos alternados de *exons*, ou regiões que codificam proteínas, e *introns*, ou porções não codificantes. O problema de reconhecimento de junções de *splice* consiste em identificar se uma determinada região da seqüência (frequentemente seu centro) possui uma fronteira do tipo *intron-exon* (IE), *exon-intron* (EI) ou se nenhum desses sítios está presente (N). Nesse conjunto, os elementos com atributos de valor diferente para os nucleotídeos A, T, C ou G foram eliminados.

Como as SVMs lidam somente com atributos numéricos, os dados utilizados para o treinamento e teste desse conjunto foram codificados para um formato numérico. Utilizou-se para isso uma codificação canônica sugerida no trabalho de PEDERSEN e NIELSEN [1997]. De acordo com essa codificação, cada nucleotídeo (atributo) das seqüências é representado por quatro bits, como segue $A = (0\ 0\ 0\ 1)$, $C = (0\ 0\ 1\ 0)$, $G = (0\ 1\ 0\ 0)$ e $T = (1\ 0\ 0\ 0)$.

O conjunto *Ionosphere* consiste em dados provenientes de medições da ionosfera terrestre por radares. O objetivo é a detecção ou não da presença de estruturas na ionosfera terrestre. Os sinais recebidos pelos radares foram processados por meio de uma função de autocorrelação, que analisa o tempo (duração) de um pulso e seu valor associado. Os exemplos desse conjunto de dados são descritos por dois atributos para cada valor associado a um determinado pulso.

O conjunto *Wisconsin Breast Cancer* possui dados referentes ao diagnóstico de câncer de mama, que pode ser maligno ou benigno. Os dados foram obtidos por meio da análise de imagens da mama e extração das características consideradas de maior relevância. Os exemplos são descritos por 30 características que representam cada uma das imagens analisadas, sendo dez delas medidas originais, obtidas por meio da extração de características da imagem e as demais resultantes do cálculo de médias ou desvios padrões sobre os

dados originais. Esse conjunto será descrito em todos os experimentos realizados como *wBCancer*.

O conjunto *Expressão Gênica* apresenta medidas do nível de expressão para classificação funcional de genes. Essa aplicação consiste em categorizar um gene em uma classe que indique sua função no meio celular. Para isso, recorre-se à análise do nível de expressão desse gene em diferentes experimentos laboratoriais. Obtém-se assim valores numéricos que precisam ser analisados e que seguem algum padrão de acordo com a classe do tecido. Esse conjunto apresenta medidas de expressão de 2467 genes obtidas em 79 experimentos laboratoriais distintos. Destes genes, 207 encontram-se categorizados em cinco classes funcionais e foram, então, selecionados para a realização dos experimentos deste trabalho. Em todos os experimentos realizados, esse conjunto será denominado *ExpGen*.

Por fim, o conjunto de dados *Golub* consiste de dados de expressão gênica de amostras de pacientes com leucemia aguda (GOLUB et al. [1999]). Esses dados de expressão foram obtidos de 72 imagens de *microarrays*, e medem os níveis de expressão de 6817 genes humanos. Duas classes são consideradas, que representam dois tipos diferentes de leucemia aguda analisadas nesse conjunto: uma com as amostras de leucemia linfóide aguda, ou ALL (*Acute Lymphoid Leukemia*), e outra com as amostras de leucemia mielóide aguda, ou AML (*Acute Myeloid Leukemia*).

Uma etapa de pré-processamento foi aplicada de modo a simplificar os dados existentes nesse conjunto, a mesma utilizada em DUDOIT et al. [2000]: todos os dados foram convertidos para um valor entre 100 e 16000 e foi realizada uma filtragem nos dados originais, sendo selecionados apenas os genes que apresentavam valor de expressão tal que $\text{mínimo/máximo} > 5$ e $(\text{máximo} - \text{mínimo}) > 500$, resultando em uma redução total para apenas 3571 genes no conjunto pré-processado. Em seguida, para suavizar os dados, foi aplicado logaritmo na base 10.

Na Tabela 4.1 tem-se a descrição das principais características de cada um dos conjuntos de dados descritos, em que é apresentado o conjunto de dados, o número total de exemplos existentes nele (*Exemplos*), o número de atributos ou a dimensionalidade dos dados (*Atributos*) e a descrição das classes presentes (*Classes*).

4.4 Experimentos e Resultados

Para prover estimativas dos erros dos classificadores induzidos nos estudos realizados, os conjuntos de dados foram divididos segundo o método *10-fold cross validation* (MIT-

Tabela 4.1: Descrição dos conjuntos de dados utilizados nos experimentos.

Conjunto	Exemplos	Atributos	Classes
Splice	3175	60	IE, EI, N
Ionosphere	354	34	good, bad
wBCancer	569	30	M, B
ExpGen	207	79	B, H, T, R, P
Golub	72	3571	ALL, AML

CHELL [1997]). Nesse método, o conjunto de dados é dividido em dez partições de tamanho aproximadamente igual. Enquanto nove dessas partições são utilizadas no treinamento do classificador, a partição restante é utilizada no teste do mesmo. Tem-se, desse modo, dez pares de conjuntos de treinamento e teste. O erro total do classificador é dado pela média dos erros individuais verificados em cada partição de teste. Cada uma das técnicas de detecção de ruído avaliadas foram aplicadas dez vezes, uma para cada partição do conjunto de dados produzida pelo método *10-fold cross validation*.

Inicialmente, os experimentos foram realizados com os conjuntos *Splice*, *Ionosphere*, *Wisconsin Breast Cancer*, *Expressão Gênica* e *Golub*. Cada um desses conjuntos foi pré-processado de acordo com o procedimento apresentado na Seção 4.3.1, de modo a adequá-los para utilização nesse projeto. Em seguida, cada um deles foi submetido às técnicas de detecção de ruído *AllKNN*, *ENN*, *RENN*, *DROPs 1 a 5* e *DEL* avaliadas. Os conjuntos de dados pré-processados resultantes, reduzidos pelas técnicas de detecção de ruído, foram então apresentados aos classificadores SVMs e algoritmo C4.5.

Cada um dos conjuntos de dados utilizados foi também apresentado diretamente aos classificadores, de modo a comparar o desempenho obtido por esses com os dados originais e pré-processados (livres de ruídos).

Todos os experimentos foram realizados em um computador com processador dual Intel Pentium 4 de 3.0GHz e 1.0GB de memória RAM. Para a execução das técnicas de detecção de ruído, foi utilizado o código disponibilizado pelos autores WILSON e MARTINEZ [2000]. Diferentes valores do parâmetro k das técnicas utilizadas, que determina o número de vizinhos mais próximos no k -NN, foram avaliados. Esses seguem uma progressão geométrica que contém o valor 3 (valor *default* do código utilizado). Os valores testados foram $k = 1, 3$ ou 9 .

Para treinamento das SVMs, utilizou-se o software *SVM-Torch II* (COLLOBERT e BENGIO [2001]). Para efeito de comparação, os valores dos parâmetros das SVMs foram mantidos iguais em todos experimentos, sendo estes os valores *default* do software utilizado. Para o treinamento das ADs, foi utilizado o algoritmo C4.5 (QUINLAN [1993]) e,

assim como para as SVMs, os valores dos parâmetros das ADs foram mantidos iguais em todos experimentos, sendo estes os valores *default* do software utilizado.

Para conversão dos diversos conjuntos de dados gerados durante os experimentos para os formatos exigidos pelo código de WILSON e MARTINEZ [2000], pela ferramenta *SVM-Torch II* e pelo algoritmo C4.5, assim como para a determinação dos melhores modelos de SVMs e ADs para cada conjunto avaliado e para a criação dos diferentes modelos de combinação apresentados, foram implementados *scripts* em linguagem *Perl*.

A redução do conjunto de dados original após a aplicação de cada uma das técnicas de detecção de ruído mencionadas pode ser observada na Tabela 4.2, em que é possível notar que as técnicas AllKNN, ENN e RENN reduziram entre 22% e 42% os conjuntos de dados originais, ao passo que as técnicas DROPs 1 a 5 e DEL removeram uma maior quantidade de dados do conjunto inicial, eliminando entre 72% e 97% dos dados existentes nos conjuntos originais.

A maior eliminação observada pelas técnicas DEL e DROPs 1 a 5 ocorre pelo fato delas também eliminarem prováveis dados redundantes e não apenas ruídos.

Tabela 4.2: Redução média dos conjuntos de dados originais (em porcentagem) com a aplicação das técnicas de detecção de ruído.

	<i>k</i>	Splice	Ionosphere	Golub	ExpGen	wBCancer
AllKNN	1	23.38	30.82	25.55	26.23	23.19
	3	24.30	34.45	29.30	27.53	24.06
	9	25.32	37.86	33.34	29.08	24.78
ENN	1	22.86	31.50	25.28	25.56	22.67
	3	22.80	32.68	27.78	25.65	22.04
	9	23.30	33.22	33.20	26.76	22.42
RENN	1	23.02	32.00	25.28	25.84	22.72
	3	23.26	33.19	28.19	25.94	22.09
	9	24.08	36.21	42.23	27.92	23.21
DROP1	1	96.86	94.73	91.67	94.97	96.67
	3	96.40	94.36	87.78	92.99	95.73
	9	94.65	93.59	74.86	88.02	93.70
DROP2	1	92.15	91.96	86.53	93.28	92.19
	3	91.71	91.08	82.23	90.91	91.05
	9	89.18	90.23	73.20	85.84	89.84
DROP3	1	94.48	94.45	87.36	94.10	94.62
	3	93.85	94.45	86.53	92.41	92.76
	9	92.62	92.25	82.08	87.00	91.83
DROP4	1	94.30	93.27	87.36	93.91	94.60
	3	93.00	92.50	83.61	91.50	92.62
	9	90.67	90.00	75.40	86.23	90.84
DROP5	1	95.14	93.84	87.90	94.15	94.53
	3	94.03	92.30	83.89	92.12	94.04
	9	92.15	90.08	72.78	88.30	92.04
DEL	1	94.62	86.55	91.80	91.12	96.76
	3	96.54	91.90	88.89	90.96	97.03
	9	95.30	90.37	76.67	87.73	96.62

Nas tabelas 4.3 e 4.4 são apresentadas as taxas de erro e desvio padrão médios obtidos

pelos classificadores induzidos pelos algoritmos SVMs e C4.5, quando aplicados sobre os conjuntos de dados originais, antes de serem apresentados às técnicas de detecção de ruído, e pré-processados, obtidos após a aplicação das técnicas de detecção de ruído aos dados originais.

Para todos os resultados apresentados, em *itálico* estão destacados aqueles superiores ou semelhantes aos obtidos para os dados originais. Em **negrito** é apresentado o melhor resultado obtido, dentre todos os demais encontrados nos experimentos realizados, para cada conjunto de dados avaliado.

Tabela 4.3: Taxas de erro das SVMs aplicadas sobre os dados originais e pré-processados.

	k	Splice	Ionosphere	Golub	ExpGen	wBCancer
Originais		3.59+-1.13	6.83+-4.88	29.28+-10.75	7.66+-6.82	36.57+-8.58
AllKNN	1	3.75+-1.65	9.68+-4.88	32.14+-13.88	<i>7.69+-5.13</i>	<i>36.74+-9.00</i>
	3	3.78+-1.24	12.82+-5.44	34.82+-13.91	8.19+-3.90	<i>36.74+-9.00</i>
	9	3.97+-1.29	14.54+-4.38	34.82+-13.91	8.69+-3.76	<i>36.74+-9.00</i>
ENN	1	<i>3.68+-1.58</i>	11.68+-5.63	32.14+-13.88	8.67+-4.34	<i>36.74+-9.00</i>
	3	<i>3.71+-1.19</i>	12.82+-5.92	34.82+-13.91	<i>7.71+-3.32</i>	<i>36.57+-8.66</i>
	9	3.84+-1.46	15.12+-7.99	34.82+-13.91	9.64+-5.48	<i>36.74+-9.00</i>
RENN	1	3.87+-1.58	11.98+-5.54	32.14+-13.88	8.67+-4.34	<i>36.74+-9.00</i>
	3	3.78+-1.21	12.53+-6.19	34.82+-13.91	<i>7.71+-3.32</i>	<i>36.57+-8.66</i>
	9	3.93+-1.39	16.53+-7.61	56.25+-20.15	9.17+-4.16	<i>36.74+-9.00</i>
DROP1	1	21.13+-2.59	34.54+-14.02	46.61+-20.90	8.64+-7.02	38.15+-10.97
	3	20.47+-3.54	18.52+-6.07	39.46+-25.00	8.21+-5.58	53.41+-14.21
	9	17.29+-2.04	16.23+-7.10	31.78+-19.52	9.64+-5.00	52.53+-11.41
DROP2	1	11.56+-2.58	17.67+-6.29	31.96+-19.68	<i>7.19+-5.12</i>	37.27+-9.10
	3	12.00+-2.99	24.24+-6.69	<i>24.82+-17.70</i>	8.62+-6.67	37.27+-9.19
	9	11.50+-1.93	16.24+-6.48	<i>23.57+-19.06</i>	9.21+-6.25	43.06+-9.82
DROP3	1	15.90+-1.98	16.82+-6.55	31.96+-16.42	8.19+-5.13	44.63+-12.61
	3	14.52+-2.49	17.10+-6.34	20.71+-15.13	8.17+-5.96	48.68+-13.80
	9	13.29+-1.83	19.12+-9.07	48.03+-22.21	8.69+-4.92	52.89+-11.11
DROP4	1	15.78+-1.77	25.24+-16.95	31.96+-16.42	<i>7.69+-5.69</i>	45.17+-12.64
	3	14.52+-2.30	14.56+-5.81	<i>21.07+-13.72</i>	9.59+-6.79	46.05+-12.71
	9	12.22+-1.19	14.82+-8.71	33.93+-25.49	8.19+-5.50	49.38+-11.85
DROP5	1	13.42+-1.03	17.97+-4.53	<i>22.32+-19.29</i>	<i>7.67+-6.40</i>	39.19+-11.02
	3	11.97+-1.81	18.52+-4.92	29.46+-21.97	<i>6.71+-5.10</i>	46.22+-11.81
	9	11.24+-1.65	14.27+-8.32	<i>21.25+-15.53</i>	8.19+-5.03	37.81+-9.92
DEL	1	18.96+-2.78	24.18+-6.70	45.36+-26.19	<i>6.24+-6.78</i>	38.32+-9.03
	3	28.50+-2.98	19.35+-11.01	36.43+-26.60	5.74+-4.37	42.19+-12.57
	9	26.24+-2.67	15.40+-8.01	36.43+-27.44	8.17+-5.04	43.77+-13.52

De acordo com os resultados apresentados nas tabelas 4.3 e 4.4, é possível notar que, para os conjuntos de dados *Splice* e *Ionosphere*, os resultados obtidos pelo classificador SVM após a etapa de eliminação de ruídos foram inferiores aos observados pelo mesmo classificador quando aplicado aos dados originais. O mesmo ocorreu para o classificador C4.5, exceto quando da aplicação da técnica ENN com $k=1$ para o conjunto *Ionosphere*, em que o resultado obtido por esse classificador foi levemente superior para o conjunto de dados pré-processado.

Para o conjunto de dados *Golub*, pode-se observar que os dois classificadores avalia-

Tabela 4.4: Taxas de erro das ADs aplicadas sobre os dados originais e pré-processados.

	k	Splice	Ionosphere	Golub	ExpGen	wBCancer
Originais		5.86+-1.16	<i>9.70+-4.30</i>	16.44+-8.12	8.19+-4.60	5.62+-3.47
AllKNN	1	6.55+-0.84	10.25+-4.50	17.87+-8.91	<i>8.16+-7.46</i>	7.92+-3.82
	3	6.38+-1.12	11.39+-4.68	<i>13.58+-12.69</i>	<i>8.65+-4.91</i>	7.74+-4.48
	9	6.36+-0.99	12.26+-5.06	<i>11.97+-14.24</i>	12.52+-5.07	<i>6.87+-3.93</i>
ENN	1	6.55+-1.09	9.39+-3.57	<i>16.44+-10.54</i>	9.18+-6.10	7.57+-3.13
	3	<i>6.04+-1.13</i>	11.68+-4.34	<i>13.58+-12.69</i>	12.03+-7.47	<i>7.04+-3.21</i>
	9	6.39+-1.30	11.99+-6.84	<i>11.08+-12.66</i>	12.49+-7.07	7.40+-3.80
RENN	1	6.60+-1.06	10.54+-5.06	<i>16.44+-10.54</i>	9.66+-6.32	7.56+-2.99
	3	6.33+-1.10	11.40+-4.02	<i>13.58+-12.69</i>	12.50+-7.10	7.57+-2.99
	9	6.51+-1.14	13.39+-5.88	29.48+-18.04	12.49+-7.07	<i>6.87+-3.88</i>
DROP1	1	31.76+-2.97	47.03+-14.32	41.96+-24.57	37.46+-24.40	13.88+-5.63
	3	35.16+-4.39	29.33+-11.92	31.63+-13.63	31.45+-6.35	19.32+-13.25
	9	33.16+-3.40	24.54+-10.23	16.98+-13.19	19.64+-10.31	14.07+-6.56
DROP2	1	27.40+-3.40	31.90+-9.85	37.70+-20.99	26.88+-21.09	9.13+-4.27
	3	28.37+-3.21	43.32+-8.59	20.55+-15.32	42.20+-22.49	9.84+-4.29
	9	25.60+-3.31	37.05+-12.24	10.54+-13.05	27.40+-14.36	10.04+-5.41
DROP3	1	31.25+-5.74	53.04+-12.68	33.77+-22.37	36.14+-20.67	11.96+-4.87
	3	32.03+-3.21	25.97+-11.83	24.66+-11.69	35.74+-14.42	8.79+-3.79
	9	26.79+-5.85	19.07+-7.45	19.48+-14.55	21.07+-16.43	12.31+-11.03
DROP4	1	31.94+-5.19	45.53+-14.74	33.77+-22.37	34.64+-22.06	11.96+-4.87
	3	30.92+-2.83	44.43+-13.80	19.12+-11.78	45.06+-14.69	10.01+-4.04
	9	28.26+-5.19	29.94+-17.68	26.44+-14.19	22.97+-17.41	10.55+-5.12
DROP5	1	29.57+-1.90	35.35+-12.74	26.63+-15.22	38.52+-21.15	11.60+-5.16
	3	29.10+-3.46	41.62+-9.88	31.62+-20.30	44.83+-23.98	11.24+-4.60
	9	29.25+-3.48	35.04+-16.50	<i>11.44+-11.28</i>	25.53+-15.22	10.91+-4.28
DEL	1	31.84+-3.66	39.88+-12.73	30.73+-16.10	69.19+-23.59	11.60+-4.42
	3	32.63+-4.28	49.58+-7.19	34.84+-19.97	40.20+-21.03	15.30+-7.19
	9	32.70+-3.48	23.97+-14.10	21.45+-18.15	29.87+-19.44	14.96+-7.23

dos apresentaram resultados superiores quando aplicados aos dados pré-processados. O mesmo ocorreu para o conjunto *ExpGen* com o classificador SVM. O algoritmo C4.5, para o conjunto *ExpGen*, quando aplicado sobre os dados pré-processados, apresentou resultados melhores em apenas um caso, para a técnica AllKNN com $k=1$.

Com relação ao conjunto de dados *wBCancer*, nota-se que, para o classificador SVM, foram obtidos resultados semelhantes quando esse foi aplicado aos dados pré-processados para as técnicas ENN e RENN para $k=3$, com deterioração dos resultados após a aplicação das técnicas avaliadas. Para o algoritmo C4.5, os resultados para os dados pré-processados foram sempre inferiores.

Nas tabelas 4.5 e 4.6 é apresentado um resumo dos resultados obtidos pelos classificadores investigados. São apresentados apenas os conjuntos de dados e técnicas em que se obteve resultados superiores ou semelhantes aos obtidos para os dados originais. Nesse sentido, Δ indica resultados semelhantes, sejam levemente superiores ou inferiores; $\blacktriangle$ indica resultados superiores e $\square$ indica resultados iguais.

Os tempos computacionais, assim como a complexidade dos modelos propostos pelos classificadores induzidos pelos algoritmos SVMs e C4.5, também foram analisados. Para

Tabela 4.5: Resumo dos resultados obtidos pelo classificador SVM.

	k	Splice	Golub	ExpGen	wBCancer
AllKNN	1			Δ	Δ
	3				Δ
	9				Δ
ENN	1	Δ			Δ
	3	Δ		Δ	$\square$
	9				Δ
RENN	1				Δ
	3			Δ	$\square$
	9				Δ
DROP2	1			$\blacktriangle$	
	3		$\blacktriangle$		
	9		$\blacktriangle$		
DROP3	3		$\blacktriangle$		
DROP4	1			Δ	
	3		$\blacktriangle$		
DROP5	1		$\blacktriangle$	Δ	
	3		Δ	$\blacktriangle$	
	9		$\blacktriangle$		
DEL	1			$\blacktriangle$	
	3			$\blacktriangle$	

Tabela 4.6: Resumo dos resultados obtidos pelo classificador C4.5.

	k	Splice	Ionosphere	Golub	ExpGen	wBCancer
AllKNN	1				$\blacktriangle$	
	3			$\blacktriangle$	Δ	
	9			$\blacktriangle$		Δ
ENN	1		$\blacktriangle$	Δ		
	3	Δ		$\blacktriangle$		Δ
	9			$\blacktriangle$		
RENN	1			Δ		
	3			$\blacktriangle$		
	9					Δ
DROP2	9			$\blacktriangle$		
DROP5	9			$\blacktriangle$		

a determinação da complexidade foi considerado, para as SVMs, o número de vetores de suporte (*Support Vectors* - *SVs*) do modelo induzido. Um número menor de SVs indica menor complexidade do modelo obtido, uma vez que esses são os dados que determinam a fronteira de decisão induzida pelas SVMs.

Para as ADs, a determinação da complexidade se deu pela análise do tamanho médio das árvores produzidas pelo algoritmo C4.5. Reduções nas árvores refletem em ganhos de compreensibilidade dos modelos obtidos, uma vez que árvores menores podem ser consideradas mais fáceis de serem examinadas.

As tabelas 4.7 e 4.8 apresentam a complexidade dos modelos obtidos pelos classificadores induzidos pelos algoritmos SVMs e C4.5. Nessas tabelas é possível notar que, para os dois classificadores, os modelos foram simplificados após a aplicação das técnicas de detecção de ruído sendo que, tanto o tamanho médio das árvores produzidas pelo algoritmo C4.5, como o número médio de SVs obtidos pelas SVMs, foram reduzidos.

Tabela 4.7: Número médio de SVs obtidos pelas SVMs aplicadas sobre os dados originais e pré-processados.

	<i>k</i>	Splice	Ionosphere	Golub	ExpGen	wBCancer
Originais		734.14+-5.71	66.85+-3.05	64.80+-0.42	35.72+-1.09	505.80+-1.40
AllKNN	1	566.44+-3.66	31.80+-3.08	54.60+-0.70	26.96+-1.78	434.60+-2.95
	3	548.80+-3.83	24.10+-1.85	51.90+-1.91	24.72+-2.25	429.70+-3.80
	9	533.40+-5.87	22.50+-1.90	49.00+-2.16	22.56+-1.85	425.60+-3.24
ENN	1	572.50+-3.85	31.35+-3.74	54.80+-0.92	27.30+-1.90	437.40+-3.24
	3	565.54+-4.87	24.30+-2.94	53.00+-1.56	25.36+-2.88	441.00+-3.53
	9	555.34+-7.21	24.60+-1.58	49.10+-2.51	22.90+-1.07	438.80+-1.99
RENN	1	570.87+-5.73	29.10+-2.89	54.80+-0.92	26.34+-2.18	437.10+-3.48
	3	557.04+-5.40	22.30+-2.31	52.70+-1.89	24.14+-2.30	440.70+-3.56
	9	547.70+-4.10	22.70+-1.49	15.00+-24.21	22.28+-2.11	434.30+-1.34
DROP1	1	88.97+-4.89	11.40+-2.46	7.00+-1.15	9.74+-1.26	19.90+-2.23
	3	97.74+-7.68	10.60+-2.22	9.80+-2.15	11.44+-1.74	25.30+-6.23
	9	132.84+-9.98	14.10+-2.28	19.10+-2.13	14.42+-2.26	36.90+-6.22
DROP2	1	173.07+-3.55	18.50+-2.46	10.70+-1.49	11.06+-1.38	45.40+-3.66
	3	184.97+-8.91	19.00+-1.70	13.80+-1.68	13.66+-0.64	51.90+-5.74
	9	221.20+-9.18	21.60+-2.67	20.30+-1.49	16.98+-1.26	58.80+-4.47
DROP3	1	136.44+-5.33	13.30+-2.40	10.10+-1.59	9.82+-1.77	31.60+-3.66
	3	147.84+-7.78	11.80+-1.55	10.70+-2.06	10.66+-1.61	42.20+-5.00
	9	166.20+-6.46	11.80+-1.75	13.90+-1.73	12.58+-1.43	47.50+-5.27
DROP4	1	140.37+-6.13	16.00+-1.88	10.10+-1.59	10.20+-1.43	31.80+-3.29
	3	162.70+-8.15	16.10+-2.51	12.80+-1.62	12.18+-2.10	43.00+-5.62
	9	197.84+-5.15	14.90+-2.68	18.70+-1.94	14.76+-1.56	53.10+-4.41
DROP5	1	123.70+-3.91	16.10+-1.37	9.70+-1.49	11.66+-1.46	32.10+-6.21
	3	141.17+-6.85	18.00+-3.77	12.60+-1.43	13.42+-1.57	34.90+-2.51
	9	166.84+-2.99	18.25+-3.39	20.60+-2.36	16.34+-2.33	46.30+-4.67
DEL	1	131.97+-6.38	25.50+-2.46	6.90+-2.33	15.26+-0.74	19.40+-1.90
	3	90.64+-3.47	17.35+-1.86	9.00+-2.00	14.80+-1.12	17.90+-3.72
	9	112.37+-2.91	13.80+-2.61	17.80+-3.39	14.46+-3.15	20.20+-2.74

Os respectivos tempos de execução das técnicas de detecção de ruído utilizadas foram calculados objetivando avaliar a complexidade destas. Eles são apresentados na Tabela

Tabela 4.8: Tamanho médio das árvores produzidas pelo algoritmo C4.5 aplicado sobre os dados originais e pré-processados.

	k	Splice	Ionosphere	Golub	ExpGen	wBCancer
Originais		417.00+-10.83	28.00+-3.43	4.40+-0.96	14.00+-1.41	23.40+-2.27
AllKNN	1	233.00+-11.47	13.60+-2.32	4.00+-1.05	9.00+-0.94	18.00+-1.41
	3	204.60+-12.43	10.20+-1.93	3.40+-0.84	7.80+-1.03	17.40+-1.84
	9	173.00+-7.54	8.20+-2.34	3.40+-0.84	7.00+-0.00	14.60+-1.84
ENN	1	245.00+-14.36	11.40+-1.58	4.00+-1.05	8.60+-0.84	18.60+-2.27
	3	226.20+-18.19	10.20+-2.15	3.40+-0.84	8.60+-0.84	16.60+-2.46
	9	195.80+-17.29	9.40+-2.46	3.40+-0.84	8.20+-1.40	15.80+-1.40
RENN	1	240.60+-14.41	11.80+-1.68	4.00+-1.05	8.60+-0.84	18.00+-2.36
	3	208.60+-17.12	10.20+-2.15	3.40+-0.84	8.40+-0.96	16.40+-2.32
	9	161.00+-8.00	8.40+-2.12	2.00+-1.70	8.00+-1.41	13.40+-1.26
DROP1	1	55.80+-8.65	3.80+-1.03	3.00+-0.00	8.60+-0.84	4.40+-1.64
	3	59.40+-6.85	4.80+-1.13	3.00+-0.00	8.20+-1.03	5.60+-2.32
	9	80.60+-10.74	5.80+-1.68	3.00+-0.00	7.80+-1.03	5.80+-2.70
DROP2	1	115.40+-10.36	6.60+-1.26	3.00+-0.00	8.80+-0.63	5.80+-1.40
	3	130.20+-6.54	7.20+-1.47	3.00+-0.00	9.00+-0.00	7.60+-1.64
	9	153.40+-8.10	7.80+-1.68	3.00+-0.00	8.40+-0.96	8.80+-1.75
DROP3	1	87.80+-10.33	4.60+-1.84	3.00+-0.00	8.20+-1.03	4.80+-1.13
	3	98.20+-9.80	3.60+-0.96	3.00+-0.00	7.40+-0.84	5.60+-1.35
	9	101.80+-5.60	3.60+-1.35	3.00+-0.00	6.80+-0.63	7.00+-1.34
DROP4	1	93.40+-6.09	5.20+-1.47	3.00+-0.00	8.20+-1.03	4.80+-1.13
	3	116.20+-6.19	5.60+-0.96	3.00+-0.00	8.20+-1.03	6.40+-0.96
	9	127.80+-7.55	5.20+-1.47	3.00+-0.00	7.60+-0.96	7.60+-1.35
DROP5	1	81.80+-4.91	6.20+-1.40	3.00+-0.00	8.60+-0.84	5.20+-1.47
	3	93.40+-8.10	6.60+-1.26	3.00+-0.00	8.40+-0.96	4.40+-1.64
	9	109.00+-9.80	6.00+-1.41	3.00+-0.00	8.20+-1.03	6.20+-1.40
DEL	1	81.40+-6.91	8.40+-2.99	2.80+-0.63	7.40+-0.84	3.60+-0.96
	3	56.20+-8.17	5.20+-0.63	3.00+-0.00	8.80+-0.63	4.20+-1.03
	9	67.00+-8.48	5.00+-0.94	3.00+-0.00	7.80+-1.03	4.20+-1.03

4.9, em que é possível verificar um maior tempo computacional na execução das técnicas DROP1 a DROP5 e DEL, que foram as que eliminaram uma maior quantidade de dados. Os tempos são apresentados em segundos.

Embora seja possível considerar os tempos de execução indicados nessa tabela relativamente altos para o conjunto de dados *Splice*, esses não são excessivamente elevados para os demais conjuntos estudados. Além disso, deve-se destacar que a etapa de eliminação de ruídos é aplicada uma única vez sobre os conjuntos de treinamento, gerando dados que podem ser utilizados por diversas técnicas de AM.

Nas tabelas 4.10 e 4.11 são apresentados os tempos computacionais necessários para treinamento dos classificadores SVMs e algoritmo C4.5 para os dados originais e pré-processados.

Como pode ser observado nessas tabelas, os tempos computacionais obtidos pelo classificador SVM, quando aplicado sobre os dados pré-processados, são inferiores para todos os conjuntos estudados e também para a maioria das técnicas de detecção de ruído investigadas. O fato desse tempo ser superior para algumas das técnicas de detecção

Tabela 4.9: Tempos (em segundos) da aplicação das técnicas de eliminação de ruídos.

	<i>k</i>	Splice	Ionosphere	Golub	ExpGen	wBCancer
AllKNN	1	82.06	0.45	2.52	0.39	1.03
	3	82.11	0.45	2.50	0.38	1.03
	9	82.21	0.45	2.48	0.39	1.04
ENN	1	135.06	0.70	2.80	0.48	1.43
	3	151.22	0.77	3.05	0.51	1.44
	9	176.82	0.93	5.45	0.62	1.80
RENN	1	136.91	0.70	2.79	0.48	1.43
	3	159.26	0.78	3.07	0.51	1.44
	9	191.06	0.93	6.59	0.75	2.05
DROP1	1	273.93	1.27	5.10	1.12	2.94
	3	438.77	1.98	7.41	1.72	4.68
	9	923.92	3.97	13.27	3.63	9.58
DROP2	1	329.87	1.29	4.19	0.92	3.46
	3	564.21	1.92	5.57	1.41	5.54
	9	1162.04	3.23	9.82	2.84	10.37
DROP3	1	446.56	1.67	4.39	0.97	3.71
	3	722.49	2.38	6.15	1.43	5.62
	9	1409.66	3.70	13.15	2.67	10.59
DROP4	1	445.91	1.55	4.39	0.97	3.69
	3	722.43	2.29	5.86	1.43	5.65
	9	1410.44	3.58	10.22	2.75	10.24
DROP5	1	410.48	2.62	6.54	1.90	4.33
	3	621.81	4.07	9.53	2.92	6.73
	9	1182.44	7.43	16.15	5.55	12.56
DEL	1	351.60	1.60	5.06	1.11	3.99
	3	756.96	2.69	6.91	1.62	6.28
	9	1494.25	3.79	10.98	2.96	11.41

de ruído investigadas é explicado pelo fato destas terem reduzido muito drasticamente o conjunto de dados original, dificultando a posterior interpretação dos dados pelas SVMs. Para o algoritmo C4.5, é possível observar que os tempos computacionais obtidos, quando aplicado sobre os dados pré-processados, são inferiores para todos os conjuntos e técnicas de detecção de ruído analisados.

4.5 Considerações Finais

Os conjuntos de dados inicialmente selecionados para a realização desse trabalho, a metodologia de desenvolvimento empregada, assim como os diferentes experimentos realizados foram apresentados e seus resultados discutidos neste capítulo.

Foram selecionados conjuntos de dados pertencentes ao repositório UCI, amplamente utilizados na literatura, que apresentam, em geral, pouco ou nenhum ruído e pequena complexidade, assim como conjuntos com dados de expressão gênica, mais complexos e com elevada quantidade de ruído.

Os resultados observados não foram satisfatórios para todos os conjuntos de dados

Tabela 4.10: Tempos de treinamento (em segundos) das SVMs aplicadas sobre os dados originais e pré-processados.

	k	Splice	Ionosphere	Golub	ExpGen	wBCancer
Originais		4.791	0.174	0.641	0.368	0.348
AllKNN	1	3.340	0.153	0.574	0.366	0.304
	3	3.216	0.148	0.552	0.367	0.290
	9	3.104	0.147	0.537	0.352	0.288
ENN	1	3.392	0.152	0.578	0.368	0.295
	3	3.357	0.148	0.562	0.345	0.298
	9	3.297	0.147	0.537	0.351	0.298
RENN	1	3.377	0.151	0.576	0.366	0.295
	3	3.301	0.147	0.559	0.341	0.300
	9	3.229	0.147	0.464	0.346	0.292
DROP1	1	0.269	0.267	0.467	0.782	0.259
	3	0.271	0.257	0.403	0.682	0.249
	9	0.296	0.230	0.385	0.580	0.216
DROP2	1	0.349	0.226	0.385	0.655	0.201
	3	0.365	0.218	0.381	0.645	0.191
	9	0.441	0.214	0.393	0.558	0.182
DROP3	1	0.298	2.210	3.894	0.662	0.229
	3	0.312	2.301	3.712	0.690	0.207
	9	0.339	1.990	3.406	0.558	0.196
DROP4	1	0.303	2.112	3.527	0.646	0.228
	3	0.330	1.814	3.309	0.664	0.204
	9	0.392	1.590	3.388	0.560	0.189
DROP5	1	0.288	1.970	3.498	0.724	0.226
	3	0.305	1.871	3.249	0.659	0.221
	9	0.347	1.628	3.073	0.582	0.198
DEL	1	0.295	1.612	4.175	0.654	0.268
	3	0.268	1.764	3.138	0.655	0.271
	9	0.281	0.888	2.007	0.567	0.256

sendo que, aqueles sobre problemas de expressão gênica apresentaram, de modo geral, melhor desempenho com a aplicação das técnicas de detecção de ruído avaliadas.

Uma vez que esses conjuntos apresentam elevada complexidade e grande quantidade de ruído, e são de difícil análise por técnicas de AM, o que dificulta o processo de generalização e a consequente obtenção de resultados satisfatórios pelos classificadores aplicados a esse tipo de dados, após a etapa inicial de testes, optou-se por dar continuidade aos experimentos com foco somente em dados de expressão gênica.

A escolha de conjuntos de dados que apresentam dados ruidosos implica em aumentar a efetividade dos experimentos realizados, assim como avaliar de maneira mais precisa as técnicas de detecção de ruído em estudo. Nesse sentido, os novos experimentos realizados com esses conjuntos, assim como as combinações desenvolvidas, são apresentados nos próximos capítulos.

Tabela 4.11: Tempos (em segundos) das ADs aplicadas sobre os dados originais e pré-processados.

	k	Splice	Ionosphere	Golub	ExpGen	wBCancer
Originais		0.088	0.044	0.435	0.075	0.070
AllKNN	1	0.067	0.025	0.358	0.044	0.056
	3	0.065	0.021	0.352	0.042	0.050
	9	0.063	0.018	0.293	0.036	0.049
ENN	1	0.069	0.023	0.353	0.049	0.052
	3	0.067	0.021	0.358	0.045	0.053
	9	0.065	0.019	0.289	0.042	0.053
RENN	1	0.068	0.022	0.354	0.047	0.050
	3	0.067	0.020	0.347	0.044	0.052
	9	0.062	0.019	0.161	0.040	0.047
DROP1	1	0.011	0.003	0.051	0.005	0.004
	3	0.010	0.003	0.065	0.006	0.005
	9	0.012	0.004	0.109	0.009	0.007
DROP2	1	0.015	0.004	0.069	0.006	0.006
	3	0.017	0.004	0.084	0.008	0.008
	9	0.019	0.005	0.115	0.011	0.009
DROP3	1	0.012	0.003	0.066	0.005	0.006
	3	0.013	0.003	0.070	0.006	0.007
	9	0.015	0.003	0.086	0.009	0.007
DROP4	1	0.014	0.003	0.067	0.006	0.005
	3	0.015	0.003	0.079	0.007	0.008
	9	0.016	0.004	0.107	0.010	0.009
DROP5	1	0.011	0.003	0.065	0.006	0.006
	3	0.014	0.004	0.079	0.007	0.005
	9	0.016	0.005	0.116	0.009	0.006
DEL	1	0.013	0.005	0.050	0.009	0.004
	3	0.010	0.003	0.061	0.008	0.005
	9	0.011	0.004	0.103	0.009	0.005

Experimentos com dados de Expressão Gênica

5.1 *Considerações Iniciais*

Neste capítulo são apresentados os experimentos, assim como os resultados obtidos, realizados com conjuntos de dados de análise de expressão gênica. A Seção 5.2 detalha os testes estatísticos empregados na avaliação e comparação dos experimentos realizados, na Seção 5.3 são descritos os novos conjuntos de dados utilizados nos experimentos, ao passo que, na Seção 5.4, são discutidos os resultados obtidos.

5.2 *Avaliação dos Experimentos*

Não existe na literatura, até o momento, um teste estatístico padrão para realizar a comparação de classificadores aplicados a múltiplos conjuntos de dados. Para avaliação dos resultados obtidos pelos experimentos realizados com os diferentes conjuntos de dados analisados neste projeto, foram utilizados os testes estatísticos de Friedman e de comparações múltiplas de Dunn, descritos em DEMSAR [2006].

O teste de Friedman (FRIEDMAN [1937; 1940]) foi adotado por se tratar de um teste adequado para comparação de classificadores aplicados a múltiplos conjuntos de dados. Trata-se de um teste não-paramétrico equivalente ao teste paramétrico ANOVA

de medidas repetidas (FISHER [1959]) que, ao contrário desse teste, não assume que as medidas são distribuídas segundo a distribuição Normal.

De maneira geral, o teste de Friedman compara três ou mais grupos pareados de dados. Inicialmente, os algoritmos são ranqueados separadamente para cada conjunto de dados, sendo organizados do menor para o maior valor do ranqueamento obtido. No caso da existência de *ranks* com valores idênticos, *ranks* médios são associados a todos os algoritmos que apresentarem esses valores.

Seja r_i^j o valor do *rank* associado ao j -ésimo dos k algoritmos no i -ésimo dos N conjuntos de dados avaliados. O teste de Friedman compara os *ranks* médios de todos os algoritmos analisados, de acordo com a Equação 5.1

$$R_j = 1/N \sum_i r_i^j \quad (5.1)$$

A hipótese nula desse teste afirma que, se todos os algoritmos analisados são equivalentes, então seus *ranks* médios R_j devem ser iguais.

Se a hipótese nula é rejeitada, ou seja, se os algoritmos analisados são estatisticamente diferentes, então um teste estatístico *post-hoc* pode ser aplicado de modo a avaliar quais algoritmos diferem entre si e a significância estatística existente entre eles.

Para a aplicação de um teste *post-hoc*, quando todos os classificadores são comparados a um único classificador de controle, o teste estatístico de comparações múltiplas de Dunn (DUNN [1961]) se mostra adequado.

O teste de Dunn é um método utilizado para comparações múltiplas entre vários algoritmos. Trata-se de uma generalização do teste estatístico *post-hoc* de Bonferroni (HOCHBERG [1988]), porém esse último é indicado somente quando não existe um algoritmo ou classificador de controle e as comparações são feitas entre todos os algoritmos analisados, sem exceção.

O teste estatístico de Dunn avalia a probabilidade de duas amostras serem diferentes e o nível de significância estatística existente entre elas. Para isso, trabalha com um nível de significância global α maior que o usual, uma vez que utiliza, para a delimitação do nível de significância dos testes de comparações múltiplas, a métrica $1 - (1 - \alpha)^{\frac{1}{p}}$, ao passo que o teste de Bonferroni se vale da métrica $\frac{\alpha}{p}$.

Uma vez que $1 - (1 - \alpha)^{\frac{1}{p}} > \frac{\alpha}{p}$, para quaisquer valores de p e α , em que p é o número de características de qualidade avaliadas e α é o nível de significância global dos

testes, o teste de Dunn mostra-se mais poderoso e confiável que o teste de Bonferroni. Na prática, porém, os testes estatísticos de Bonferroni e Dunn levam, muitas vezes, a resultados semelhantes (HAYTER e TSUI [1994]).

5.3 Conjuntos de Dados

Após a análise dos resultados inicialmente obtidos, optou-se pela utilização, neste trabalho, apenas de conjuntos de dados de expressão gênica, visto que dados de expressão gênica comumente apresentam elevada quantidade de ruídos e são de difícil análise pelos classificadores, apresentando taxas de erro elevadas. Nesse sentido, foram mantidos os conjuntos iniciais *Expressão Gênica* e *Golub*, apresentados no Capítulo 4, e adicionados novos conjuntos de dados, a saber, *Leukemia*, *Lung*, *Colon* e *Ovarian*, e versões reduzidas dos conjuntos *Golub* e *Colon*, que são descritos a seguir.

Todos os conjuntos de dados selecionados referem-se a problemas de classificação de dados de expressão gênica, com exceção do conjunto *Ovarian*, selecionado por ser complexo e apresentar elevada dimensionalidade. Eles contêm, de maneira geral, dados com elevada dimensionalidade e grande quantidade de ruídos, o que os tornam de difícil compreensão para os classificadores de Aprendizado de Máquina (AM) utilizados.

Sendo assim, o conjunto de dados *Leukemia*, referido na literatura como *St. Jude leukemia* (YEOH et al. [2002]), possui dados de expressão gênica obtidos com *microarrays*. Os dados apresentam subtipos de leucemia linfóide aguda pediátrica (*pediatric acute lymphoid leukemia*). Os dados permitem agrupar esse tipo de leucemia em 6 grupos diagnosticáveis e um outro grupo que contém amostras que não se enquadram em nenhum dos grupos anteriores. O conjunto de dados original contém 12558 atributos (genes), porém foi utilizada nesse trabalho uma versão pré-processada obtida em <http://sdmc.lit.org.sg/GEDatasets> e descrita no trabalho de YEOH et al. [2002], que utilizou como atributos apenas os genes que melhor definem cada grupo (40 genes por grupo) com a medida qui-quadrado. Os valores foram então normalizados para o intervalo [0, 1].

O conjunto de dados *Lung* apresenta amostras relacionadas a câncer de pulmão (considerando três tipos distintos) e tecido normal. Os três diferentes tipos de câncer pulmonar analisados são os adenocarcinomas (ADs), os carcinomas de células escamosas (*squamous cell carcinomas*) (SQs) e os carcinóides (COID). A versão utilizada para esse conjunto de dados foi a mesma utilizada por MONTI et. al. [2003], gentilmente cedida por esses autores.

Em ALON et. al. [1999] é descrito um conjunto de dados que apresenta informações referentes a pacientes com câncer de colo. Esse conjunto, referido neste trabalho como *Colon*, contém dados da expressão gênica obtidos de 62 imagens de *microarrays*, que medem os níveis de expressão de 6500 genes humanos. Os pacientes investigados podem apresentar ou não câncer de colo. Esse conjunto de dados apresenta um conjunto com 2000 genes, após a eliminação dos genes que possuem baixa intensidade mínima.

Por fim, o último conjunto analisado consiste em dados de câncer de ovário (EMANUEL et. al. [2002]), único conjunto que não apresenta dados de expressão gênica, porém selecionado por ser complexo, apresentar elevada dimensionalidade dos dados e ruídos. Esse conjunto, denominado *Ovarian*, objetiva determinar se uma paciente apresenta ou não esse tipo de câncer. O espectro proteômico para obtenção dos dados foi produzido por espectroscopia de massa, sendo que são descritos 91 pacientes que não apresentaram o câncer e 162 que apresentam a doença. No total, 15154 amplitudes relativas da intensidade de cada massa molecular foram analisadas por paciente. Os dados foram normalizados para o intervalo $[0, 1]$.

Os conjuntos *Golub* e *Colon* foram também avaliados após a redução da dimensionalidade de seus dados. Sendo assim, esses dois conjuntos, previamente descritos, foram reduzidos de acordo com procedimento apresentado em FERES [2005], em que um Algoritmo Genético (AG) (GOLDBERG [1989]) foi utilizado no processo de seleção dos genes existentes nos conjuntos de dados, com o objetivo de selecionar os genes que melhor representam o domínio do problema. Dessa forma, a complexidade de classificação pode ser minimizada, uma vez que genes considerados menos representativos serão eliminados do conjunto de dados e, ao mesmo tempo, genes importantes para a patologia em estudo podem ser revelados.

Após a redução dos dados, o conjunto *Golub* teve seus atributos iniciais reduzidos de 3571 genes para apenas 64 atributos finais, ao passo que o conjunto *Colon* teve seus atributos reduzidos de iniciais 2000 genes para somente 16 genes. Essas reduções foram efetuadas para avaliar se a redução no número de atributos iniciais pode auxiliar no processo de identificação de ruídos e, conseqüentemente, produzir melhores resultados pelos classificadores avaliados.

A Tabela 5.1 descreve as principais características de cada um dos conjuntos de dados descritos, em que é apresentado o conjunto de dados, o seu número total de exemplos (*Exemplos*), o número de atributos ou a dimensionalidade dos dados (*Atributos*) e a descrição das classes presentes (*Classes*).

Tabela 5.1: Descrição dos conjuntos de dados utilizados nos experimentos.

Conjunto	Exemplos	Atributos	Classes
ExpGen	207	79	B, H, T, R, P
Golub	72	3571	ALL, AML
Leukemia	327	271	BCR, E2A, HYP, MLL, T-ALL, TEL, OTHERS
Lung	197	1000	AD, SQ, COID, NL
Colon	62	2000	normal, tumor
Ovarian	253	15154	normal, cancer
Golub64	72	64	ALL, AML
Colon16	62	16	normal, tumor

5.4 Experimentos e Resultados

Os conjuntos de expressão gênica selecionados para dar continuidade aos experimentos deste trabalho são *ExpGen*, *Golub*, *Leukemia*, *Lung*, *Colon* e *Ovarian*, além das versões com redução de atributos dos conjuntos *Golub* e *Colon*, denominadas *Golub64* e *Colon16*, cujas descrições se encontram na Seção 5.3.

Um novo algoritmo de AM foi também avaliado com os novos conjuntos selecionados, sendo esse o algoritmo RIPPER (COHEN [1995]), cujo treinamento foi realizado pelo simulador Weka, desenvolvido pela Universidade de Waikato, da Nova Zelândia (FRANK e WITTEN [2005]). Assim como definido para os experimentos realizados com as SVMs e C4.5, os valores dos parâmetros desse algoritmo foram mantidos iguais em todos experimentos, sendo esses os valores *default* do simulador Weka.

O algoritmo RIPPER foi utilizado por se tratar de um algoritmo de AM baseado em regras, semelhante ao algoritmo C4.5, baseado em ADs, que apresenta robustez diante de dados com elevada dimensionalidade e na presença de ruídos. Outra característica está no fato de ser mais flexível que algoritmos baseados em ADs, uma vez que, com a inclusão de novos dados, novas regras podem ser adicionadas ou modificadas sem afetar as demais existentes no conjunto.

Os experimentos seguiram a mesma metodologia já anteriormente aplicada: cada um dos novos conjuntos de dados foi submetido às técnicas de detecção de ruído *AllKNN*, *ENN*, *RENN*, *DROPs 1 a 5* e *DEL* avaliadas. Os conjuntos de dados pré-processados pelas técnicas de detecção de ruído foram então apresentados aos classificadores SVMs, RIPPER e C4.5. Cada um desses conjuntos de dados foi também apresentado diretamente aos classificadores, de modo a comparar o desempenho obtido por esses com os dados originais e com os mesmos pré-processados (livres de ruídos).

Para prover estimativas dos erros dos classificadores induzidos nos estudos realizados, os conjuntos de dados foram divididos segundo o método *10-fold cross validation* (MITCHELL [1997]). Para efeito de comparação, são apresentados todos os conjuntos utilizados, inclusive os conjuntos *Golub* e *ExpGen*, cujos resultados foram discutidos anteriormente.

A redução do conjunto de dados original após a aplicação das técnicas de detecção de ruído pode ser observada na Tabela 5.2, em que é possível notar que as técnicas AllKNN, ENN e RENN, considerando os conjuntos *Leukemia* e *ExpGen*, reduziram entre 25% e 30% os conjuntos de dados originais.

Para os conjuntos *Lung* e *Golub64*, a redução foi mais uniforme, variando entre 22.50% e 24.50% para o primeiro conjunto e entre 21% e 23% para o segundo. Para os demais conjuntos, as reduções foram menos uniformes, sendo entre 23% e 33% para o conjunto *Ovarian*, entre 25% e 42% para o conjunto *Golub*, entre 38% e 51% para o conjunto *Colon* e entre 31% e 40% para o conjunto *Colon16*. Pode-se notar que as maiores reduções ocorreram para os conjuntos sobre câncer de colo, seja normal ou com atributos reduzidos. Aqueles com atributos reduzidos apresentaram redução do conjunto de dados original inferior ao obtido pelos mesmos conjuntos sem redução de atributos.

Analisando agora as reduções obtidas pelas técnicas DROPs 1 a 5 e DEL, estas eliminaram entre 84% e 94% dos dados existentes no conjunto *Leukemia* e entre 84% e 98% dos existentes no conjunto *Golub64*. Entre 87% e 96% aqueles encontrados no conjunto *Lung*, entre 85% e 94% aqueles contidos no conjunto *ExpGen*, entre 73% e 91% os constantes no conjunto *Golub*, entre 70% e 91% os existentes no conjunto *Ovarian*, entre 65% e 96% aqueles constituintes do conjunto *Colon* e entre 81% e 95% para aqueles encontrados no conjunto *Colon16*. Como mencionado, uma maior eliminação observada para essas técnicas ocorre pelo fato delas também eliminarem prováveis dados redundantes e não apenas ruídos.

As taxas de erro e desvio padrão médios obtidos pelos classificadores induzidos pelos algoritmos SVMs, C4.5 e RIPPER, quando aplicados sobre os conjuntos de dados originais e pré-processados podem ser observados nas tabelas 5.3, 5.4 e 5.5. É possível notar que, para o algoritmo RIPPER, a redução realizada pela técnica DEL com $k=1$ para o conjunto *Golub64* e pela técnica DROP3 com $k=9$ para o conjunto *Colon* não possibilitou a criação de um modelo que representasse os dados, uma vez que esses conjuntos de dados foram drasticamente diminuídos e, como consequência, apresentaram dados insuficientes para que o algoritmo RIPPER produzisse um modelo adequado.

Tabela 5.2: Redução média dos conjuntos de dados originais (em porcentagem) com a aplicação das técnicas de detecção de ruído.

	<i>k</i>	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
AllKNN	1	25.75	23.08	25.55	26.23	39.03	23.75	22.08	36.77
	3	27.61	24.00	29.30	27.53	46.29	25.53	22.91	39.67
	9	30.09	24.57	33.34	29.08	53.38	30.43	23.19	40.32
ENN	1	25.17	22.99	25.28	25.56	38.06	23.48	20.97	34.19
	3	25.29	22.59	27.78	25.65	44.51	24.15	21.39	31.93
	9	27.77	23.09	33.20	26.76	46.61	26.95	20.97	32.25
RENN	1	25.69	23.20	25.28	25.84	39.84	23.71	21.12	35.16
	3	26.60	24.06	28.19	25.94	51.61	24.66	21.80	33.22
	9	28.90	23.80	42.23	27.92	50.00	33.08	20.97	33.70
DROP1	1	94.10	96.45	91.67	94.97	88.22	89.33	95.84	92.74
	3	92.60	94.36	87.78	92.99	78.39	85.89	92.91	91.13
	9	87.46	88.93	74.86	88.02	74.03	77.40	84.72	81.77
DROP2	1	91.31	95.78	86.53	93.28	83.87	85.10	94.72	85.00
	3	89.97	93.50	82.23	90.91	77.74	80.39	92.23	85.80
	9	84.06	88.37	73.20	85.84	66.77	70.43	84.72	78.06
DROP3	1	93.40	95.78	87.36	94.10	84.84	85.41	94.82	95.00
	3	91.53	93.75	86.53	92.41	85.16	80.99	92.91	91.13
	9	87.25	87.76	82.08	87.00	96.45	77.98	84.58	81.93
DROP4	1	92.63	95.58	87.36	93.91	82.42	84.98	94.82	94.35
	3	90.00	93.35	83.61	91.50	78.87	80.83	92.08	90.48
	9	84.55	88.27	75.40	86.23	74.67	72.41	84.58	81.13
DROP5	1	91.80	94.82	87.90	94.15	83.87	86.52	95.84	95.32
	3	89.81	93.09	83.89	92.12	77.42	83.48	92.50	91.45
	9	84.71	87.71	72.78	88.30	72.09	74.70	84.58	81.61
DEL	1	90.34	96.45	91.80	91.12	76.93	91.54	98.61	86.45
	3	92.41	95.58	88.89	90.96	71.45	89.88	95.84	94.84
	9	86.55	90.35	76.67	87.73	65.97	81.82	87.50	84.68

Para todos os resultados apresentados, em *itálico* estão destacados aqueles estatisticamente superiores ou semelhantes aos obtidos para os dados originais. Em **negrito** é apresentado o melhor resultado obtido, dentre todos os demais encontrados nos experimentos realizados, para cada conjunto de dados avaliado.

De acordo com os resultados apresentados, é possível notar que, para os conjuntos de dados *Leukemia* e *Lung*, os resultados obtidos pelo classificador SVM para os dados pré-processados foram semelhantes aos observados por esse mesmo classificador quando aplicado aos dados originais. Já o classificador C4.5 apresentou alguns resultados superiores para os dados pré-processados para ambos os conjuntos. O classificador RIPPER obteve resultados superiores apenas para o conjunto *Lung*, para o conjunto *Leukemia*, alguns resultados foram semelhantes porém, em sua maioria, mostraram-se inferiores aos obtidos para os dados originais.

Para o conjunto de dados *Golub*, é possível observar que todos os classificadores avaliados apresentaram alguns resultados superiores quando aplicados aos dados pré-processados. O classificador SVM, quando aplicado aos dados pré-processados, obteve bons resultados para o conjunto *ExpGen*, ao passo que os algoritmos C4.5 e RIPPER

Tabela 5.3: Taxas de erro das SVMs aplicadas sobre os dados originais e pré-processados.

	<i>k</i>	Leukemia	Lung	Golub	ExpGen
Originais		<i>7.05+-4.39</i>	29.42+-3.74	29.28+-10.75	7.66+-6.82
AllKNN	1	7.95+-3.91	<i>29.42+-3.74</i>	32.14+-13.88	<i>7.69+-5.13</i>
	3	7.95+-3.91	<i>29.42+-3.74</i>	34.82+-13.91	8.19+-3.90
	9	<i>7.94+-2.96</i>	<i>29.42+-3.74</i>	34.82+-13.91	8.69+-3.76
ENN	1	7.34+-3.65	<i>29.42+-3.74</i>	32.14+-13.88	8.67+-4.34
	3	8.27+-3.61	<i>29.42+-3.74</i>	34.82+-13.91	<i>7.71+-3.32</i>
	9	11.00+-4.83	<i>29.42+-3.74</i>	34.82+-13.91	9.64+-5.48
RENN	1	<i>7.65+-3.93</i>	<i>29.42+-3.74</i>	32.14+-13.88	8.67+-4.34
	3	11.01+-5.08	<i>29.42+-3.74</i>	34.82+-13.91	<i>7.71+-3.32</i>
	9	13.43+-6.26	<i>29.42+-3.74</i>	56.25+-20.15	9.17+-4.16
DROP1	1	13.47+-3.71	77.37+-28.34	46.61+-20.90	8.64+-7.02
	3	12.52+-9.31	80.37+-26.76	39.46+-25.00	8.21+-5.58
	9	11.65+-7.34	76.55+-26.55	31.78+-19.52	9.64+-5.00
DROP2	1	11.29+-3.79	77.87+-25.77	31.96+-19.68	<i>7.19+-5.12</i>
	3	14.09+-6.50	78.52+-26.95	<i>24.82+-17.70</i>	8.62+-6.67
	9	8.87+-5.06	69.26+-29.61	<i>23.57+-19.06</i>	9.21+-6.25
DROP3	1	13.44+-5.76	79.08+-24.65	31.96+-16.42	8.19+-5.13
	3	11.31+-3.23	78.05+-26.77	<i>20.71+-15.13</i>	8.17+-5.96
	9	14.35+-4.71	61.73+-35.09	48.03+-22.21	8.69+-4.92
DROP4	1	14.05+-5.63	72.58+-28.28	31.96+-16.42	<i>7.69+-5.69</i>
	3	13.45+-7.31	71.21+-30.51	21.07+-13.72	9.59+-6.79
	9	8.89+-5.32	81.05+-28.04	33.93+-25.49	8.19+-5.50
DROP5	1	11.04+-5.54	42.26+-26.67	<i>22.32+-19.29</i>	<i>7.67+-6.40</i>
	3	12.84+-7.39	36.26+-20.85	29.46+-21.97	<i>6.71+-5.10</i>
	9	8.26+-2.50	48.71+-30.89	<i>21.25+-15.53</i>	8.19+-5.03
DEL	1	13.42+-5.87	83.87+-21.27	45.36+-26.19	<i>6.24+-6.78</i>
	3	14.34+-6.02	73.26+-30.71	36.43+-26.60	5.74+-4.37
	9	9.16+-4.96	52.52+-31.29	36.43+-27.44	8.17+-5.04

	<i>k</i>	Colon	Ovarian	Golub64	Colon16
Originais		35.71+-13.56	9.11+-7.78	34.82+-12.17	35.47+-18.63
AllKNN	1	<i>35.71+-13.56</i>	13.86+-10.82	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
	3	<i>35.71+-13.56</i>	15.84+-11.39	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
	9	<i>35.71+-13.56</i>	22.21+-13.86	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
ENN	1	<i>35.71+-13.56</i>	13.86+-10.82	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
	3	<i>35.71+-13.56</i>	15.44+-10.31	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
	9	42.38+-18.60	21.00+-13.17	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
RENN	1	<i>35.71+-13.56</i>	13.86+-10.82	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
	3	44.28+-19.35	17.03+-11.51	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
	9	64.28+-13.56	26.92+-14.58	<i>34.82+-12.17</i>	<i>35.47+-18.63</i>
DROP1	1	42.38+-18.60	25.63+-10.09	65.18+-12.17	43.57+-23.14
	3	50.95+-20.24	35.84+-11.64	60.89+-16.50	41.19+-22.25
	9	40.00+-17.31	24.09+-8.85	65.18+-12.17	46.90+-23.89
DROP2	1	<i>35.71+-13.56</i>	21.10+-14.80	48.75+-20.06	64.52+-18.63
	3	46.67+-19.96	15.41+-7.28	56.96+-18.71	63.09+-19.77
	9	<i>35.71+-13.56</i>	21.35+-10.68	65.18+-12.17	61.19+-21.03
DROP3	1	39.05+-16.65	21.44+-14.34	50.18+-20.10	54.52+-23.63
	3	<i>35.71+-13.56</i>	21.80+-10.75	60.89+-16.50	64.52+-18.63
	9	53.33+-19.96	27.21+-11.31	65.18+-12.17	56.90+-22.99
DROP4	1	39.05+-16.65	19.04+-13.88	50.18+-20.10	46.43+-23.81
	3	46.67+-19.95	22.17+-10.71	59.46+-17.45	64.52+-18.63
	9	<i>35.71+-13.56</i>	24.09+-9.79	65.18+-12.17	60.24+-21.56
DROP5	1	46.67+-19.96	22.44+-11.82	65.18+-12.17	54.52+-23.63
	3	<i>35.71+-13.56</i>	16.63+-10.10	47.32+-19.90	50.24+-24.11
	9	<i>35.71+-13.56</i>	11.84+-7.35	63.75+-13.93	39.76+-21.56
DEL	1	40.00+-17.31	17.69+-8.07	65.18+-12.17	64.52+-18.63
	3	42.38+-18.60	23.71+-9.16	65.18+-12.17	57.86+-22.64
	9	45.71+-19.75	31.23+-10.70	65.18+-12.17	53.57+-23.81

Tabela 5.4: Taxas de erro das ADs aplicadas sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen
Originais		17.18+-6.98	9.14+-5.22	16.44+-8.12	8.19+-4.60
AllKNN	1	18.95+-5.07	6.06+-5.15	17.87+-8.91	8.16+-7.46
	3	17.47+-5.51	7.56+-4.21	13.58+-12.69	8.65+-4.91
	9	18.70+-5.58	7.56+-4.21	11.97+-14.24	12.52+-5.07
ENN	1	17.43+-4.75	5.56+-4.97	16.44+-10.54	9.18+-6.10
	3	16.83+-7.05	8.08+-4.83	13.58+-12.69	12.03+-7.47
	9	17.43+-6.19	8.08+-6.32	11.08+-12.66	12.49+-7.07
RENN	1	17.74+-4.87	6.06+-4.58	16.44+-10.54	9.66+-6.32
	3	16.83+-6.72	7.58+-4.87	13.58+-12.69	12.50+-7.10
	9	16.81+-4.98	9.13+-4.59	29.48+-18.04	12.49+-7.07
DROP1	1	52.70+-12.04	62.85+-15.54	41.96+-24.57	37.46+-24.40
	3	33.62+-10.81	56.00+-17.17	31.63+-13.63	31.45+-6.35
	9	33.37+-7.62	50.76+-8.17	16.98+-13.19	19.64+-10.31
DROP2	1	37.36+-5.37	50.13+-19.90	37.70+-20.99	26.88+-21.09
	3	37.64+-5.10	47.65+-20.36	20.55+-15.32	42.20+-22.49
	9	29.93+-7.43	44.58+-18.86	10.54+-13.05	27.40+-14.36
DROP3	1	42.02+-10.87	62.39+-17.94	33.77+-22.37	36.14+-20.67
	3	32.45+-7.17	41.42+-20.52	24.66+-11.69	35.74+-14.42
	9	27.83+-6.07	34.97+-5.68	19.48+-14.55	21.07+-16.43
DROP4	1	36.75+-7.12	60.39+-16.84	33.77+-22.37	34.64+-22.06
	3	32.16+-9.79	47.26+-15.61	19.12+-11.78	45.06+-14.69
	9	31.51+-7.92	44.65+-16.66	26.44+-14.19	22.97+-17.41
DROP5	1	42.51+-5.58	43.35+-21.18	26.63+-15.22	38.52+-21.15
	3	36.17+-11.32	38.52+-12.82	31.62+-20.30	44.83+-23.98
	9	31.52+-9.16	31.84+-12.23	11.44+-11.28	25.53+-15.22
DEL	1	44.95+-10.74	49.14+-16.06	30.73+-16.10	69.19+-23.59
	3	35.17+-11.23	51.72+-13.81	34.84+-19.97	40.20+-21.03
	9	30.94+-5.21	40.65+-12.13	21.45+-18.15	29.87+-19.44

	k	Colon	Ovarian	Golub64	Colon16
Originais		20.97+-13.45	3.13+-4.08	14.12+-9.55	20.96+-15.57
AllKNN	1	26.43+-21.38	3.55+-5.06	15.19+-10.15	21.43+-15.90
	3	24.52+-11.48	4.73+-4.46	15.19+-10.15	19.77+-15.38
	9	30.70+-5.29	4.36+-3.98	15.19+-10.15	19.77+-15.38
ENN	1	21.43+-19.40	3.55+-5.06	15.19+-10.15	18.10+-16.61
	3	30.00+-17.52	4.73+-4.46	13.94+-9.56	18.10+-16.61
	9	32.86+-19.30	5.54+-4.66	15.19+-10.15	18.11+-14.63
RENN	1	28.58+-23.47	3.95+-4.90	12.33+-9.98	19.77+-15.38
	3	40.72+-14.91	4.33+-4.70	9.65+-9.50	21.43+-15.90
	9	35.72+-13.55	11.41+-13.24	15.19+-10.15	19.77+-18.98
DROP1	1	32.39+-27.28	11.82+-5.85	14.12+-11.69	24.77+-21.27
	3	42.85+-17.55	4.73+-4.52	16.62+-12.89	20.72+-18.34
	9	30.24+-14.43	8.64+-8.34	6.97+-7.36	25.96+-15.07
DROP2	1	46.90+-16.21	12.57+-8.99	18.41+-13.64	42.62+-16.94
	3	39.76+-25.50	6.68+-6.97	11.08+-8.95	35.46+-20.90
	9	24.29+-14.08	8.29+-7.11	8.22+-11.43	34.77+-23.06
DROP3	1	48.57+-17.72	5.49+-5.90	12.69+-10.54	47.16+-29.16
	3	45.96+-27.88	4.73+-6.18	13.40+-15.96	39.52+-28.69
	9	43.83+-17.96	6.33+-8.27	12.51+-13.82	33.11+-25.04
DROP4	1	43.11+-22.98	7.03+-8.13	12.69+-10.54	35.25+-23.92
	3	43.07+-10.62	4.33+-4.77	16.26+-15.85	42.85+-27.67
	9	32.38+-17.21	4.35+-5.78	12.51+-13.82	34.53+-21.54
DROP5	1	46.43+-18.34	9.51+-6.25	9.65+-11.18	36.91+-21.27
	3	27.38+-17.34	6.27+-5.18	19.30+-9.41	35.72+-25.71
	9	44.74+-23.16	4.32+-2.87	16.62+-12.89	22.39+-16.95
DEL	1	46.19+-24.99	11.41+-7.13	34.84+-12.17	26.45+-25.21
	3	31.90+-17.94	12.91+-10.31	12.51+-10.00	36.19+-29.56
	9	32.86+-21.23	6.69+-6.98	6.79+-9.29	30.95+-12.63

Tabela 5.5: Taxas de erro do algoritmo RIPPER aplicado sobre os dados originais e pré-processados.

	<i>k</i>	Leukemia	Lung	Golub	ExpGen
Originais		19.28+-5.82	<i>11.05+-8.39</i>	<i>15.54+-12.54</i>	<i>12.98+-6.71</i>
AllKNN	1	21.46+-8.97	<i>7.55+-5.37</i>	12.50+-12.05	<i>13.48+-5.93</i>
	3	19.96+-9.72	<i>9.55+-6.81</i>	<i>15.18+-14.76</i>	<i>12.98+-7.39</i>
	9	22.36+-10.66	<i>9.55+-6.81</i>	16.61+-12.88	15.36+-8.82
ENN	1	<i>20.52+-7.87</i>	<i>6.55+-4.09</i>	12.50+-12.05	11.52+-6.74
	3	<i>20.52+-4.48</i>	<i>8.55+-6.23</i>	<i>15.18+-12.18</i>	13.45+-8.26
	9	21.42+-7.81	<i>9.05+-6.96</i>	15.36+-18.14	15.36+-9.90
RENN	1	22.39+-8.10	6.05+-3.93	12.50+-12.05	<i>13.43+-7.95</i>
	3	<i>19.64+-7.44</i>	<i>10.58+-6.78</i>	<i>13.75+-11.23</i>	<i>12.98+-8.04</i>
	9	22.65+-9.92	<i>12.11+-7.08</i>	30.54+-17.34	13.93+-8.15
DROP1	1	58.14+-7.44	29.42+-3.74	46.61+-20.90	61.86+-21.19
	3	53.89+-15.75	72.71+-19.04	46.25+-24.81	61.50+-20.92
	9	43.40+-8.03	50.16+-11.21	27.32+-14.73	38.60+-16.99
DROP2	1	42.81+-8.16	65.53+-27.72	46.96+-18.67	55.29+-16.63
	3	45.24+-11.58	50.85+-23.40	50.18+-17.71	48.48+-13.10
	9	32.08+-7.05	56.76+-15.00	17.50+-13.28	43.40+-9.49
DROP3	1	48.61+-10.40	74.58+-25.51	31.96+-13.38	64.31+-19.59
	3	43.72+-10.07	50.11+-25.73	32.14+-15.08	45.02+-14.99
	9	30.53+-5.31	38.45+-15.93	31.25+-17.52	43.02+-14.91
DROP4	1	49.83+-14.27	68.58+-27.18	31.96+-13.38	62.81+-18.81
	3	43.47+-10.25	46.16+-25.96	33.39+-15.29	51.29+-16.91
	9	36.98+-8.89	44.68+-19.85	31.43+-22.65	39.19+-12.77
DROP5	1	51.40+-10.55	34.21+-21.88	41.79+-17.96	73.43+-24.46
	3	50.12+-6.93	35.58+-7.54	31.96+-16.42	71.57+-17.58
	9	33.97+-8.05	34.13+-13.54	20.54+-10.95	47.31+-19.99
DEL	1	53.76+-7.40	60.08+-33.84	52.96+-19.88	83.12+-8.20
	3	51.35+-6.91	53.26+-31.96	44.64+-22.46	67.55+-11.96
	9	33.30+-8.68	40.87+-25.62	38.04+-25.62	32.79+-14.61

	<i>k</i>	Colon	Ovarian	Golub64	Colon16
Originais		<i>24.29+-17.95</i>	<i>3.55+-3.46</i>	11.25+-13.10	<i>22.62+-19.45</i>
AllKNN	1	<i>24.05+-13.06</i>	<i>2.34+-3.73</i>	<i>9.64+-9.49</i>	<i>20.95+-17.92</i>
	3	<i>26.19+-13.88</i>	1.97+-2.80	6.79+-9.84	<i>21.43+-19.41</i>
	9	30.71+-5.32	<i>3.15+-3.63</i>	<i>8.21+-9.79</i>	24.52+-18.27
ENN	1	27.38+-15.07	<i>2.38+-2.79</i>	<i>11.07+-11.19</i>	<i>23.10+-21.18</i>
	3	<i>25.71+-15.79</i>	<i>3.17+-4.12</i>	12.68+-10.53	<i>23.10+-18.03</i>
	9	32.62+-14.02	<i>3.15+-4.50</i>	<i>11.07+-11.19</i>	26.19+-19.86
RENN	1	<i>27.62+-13.85</i>	<i>2.38+-2.79</i>	<i>11.07+-11.19</i>	14.76+-19.95
	3	39.05+-20.02	<i>2.38+-2.79</i>	<i>11.25+-11.23</i>	<i>19.76+-20.54</i>
	9	35.71+-13.56	3.97+-4.59	<i>9.82+-11.70</i>	24.52+-19.89
DROP1	1	51.67+-20.94	13.38+-8.08	34.82+-12.18	41.43+-30.91
	3	49.52+-19.85	9.12+-9.27	21.96+-19.81	39.52+-35.09
	9	35.71+-11.05	8.23+-8.26	<i>11.07+-10.71</i>	28.57+-20.57
DROP2	1	41.67+-20.64	11.02+-6.52	33.39+-18.02	47.62+-17.60
	3	41.43+-15.44	3.94+-5.58	19.11+-15.84	36.90+-22.95
	9	27.62+-15.92	4.72+-4.88	20.71+-11.76	39.29+-20.12
DROP3	1	39.05+-20.02	6.69+-6.17	34.82+-15.46	64.52+-18.63
	3	39.76+-21.85	5.52+-3.83	24.64+-31.52	56.19+-25.13
	9	————	5.52+-6.00	13.93+-13.09	32.38+-15.32
DROP4	1	37.38+-18.06	8.29+-7.11	34.82+-15.46	58.33+-23.44
	3	46.19+-17.79	5.54+-3.37	27.50+-31.77	47.86+-26.45
	9	38.10+-16.42	5.14+-5.66	13.93+-13.09	37.14+-10.95
DROP5	1	45.00+-12.92	10.34+-6.61	34.82+-12.18	57.86+-22.64
	3	40.24+-15.12	5.89+-4.56	12.86+-15.72	41.90+-21.86
	9	43.57+-12.85	5.11+-4.14	20.71+-13.55	29.29+-16.42
DEL	1	38.81+-16.03	19.00+-9.43	————	45.48+-25.48
	3	19.52+-21.67	10.96+-7.02	34.82+-12.18	59.52+-23.94
	9	36.67+-14.60	7.92+-5.97	<i>8.57+-12.05</i>	27.86+-18.03

limitaram-se a poucos resultados semelhantes aos obtidos quando aplicados aos dados originais, sendo que o algoritmo RIPPER apresentou um único resultado superior, para a técnica ENN com $k=1$.

Os resultados obtidos para o conjunto *Colon* pelo classificador SVM indicam que as técnicas de detecção de ruído não melhoraram o desempenho desse classificador, uma vez que seus resultados foram iguais ou inferiores aos obtidos pelas SVMs aplicadas sobre os dados originais. O mesmo é válido para os conjuntos de dados *Golub64* e *Colon16*. Para o conjunto *Ovarian*, a etapa de eliminação de ruídos apenas deteriorou os resultados obtidos por esse classificador.

Já para o algoritmo C4.5, os resultados obtidos para o conjunto *Colon* foram inferiores aos obtidos por esse mesmo algoritmo quando aplicado sobre os dados originais. O mesmo é válido para o conjunto *Ovarian*. Para os conjuntos de dados *Golub64* e *Colon16*, foi obtido desempenho superior pelo classificador C4.5 para diversas das técnicas de detecção de ruído investigadas, com destaque para os resultados obtidos com o conjunto *Golub64*.

O algoritmo RIPPER, para o conjunto *Colon*, apresentou um único resultado superior, para a técnica DEL com $k=3$. Os demais resultados foram semelhantes ou inferiores aos obtidos quando aplicado sobre os dados originais. Para os conjuntos *Ovarian*, *Golub64* e *Colon16* é possível afirmar que a etapa de eliminação de ruídos foi satisfatória, uma vez que diversos resultados observados foram superiores ou semelhantes aos obtidos sobre os dados originais.

As tabelas 5.6, 5.7 e 5.8 apresentam um resumo, somente com as técnicas de detecção em que os classificadores investigados obtiveram resultados superiores ou semelhantes aos obtidos para os dados originais. Nessas tabelas, Δ indica resultados semelhantes, sejam levemente superiores ou inferiores; $\blacktriangle$ indica resultados superiores e $\square$ indica resultados iguais.

De modo a avaliar estatisticamente os resultados apresentados, foi aplicado o teste estatístico de Friedman, com 95% de confiança estatística. Para todos os conjuntos de dados avaliados, a hipótese nula, que afirma que todos os algoritmos analisados são equivalentes, foi rejeitada, ou seja, de acordo com este teste, existem diferenças estatísticas entre os algoritmos analisados. Assim, foi aplicado o teste de comparações múltiplas de Dunn para analisar as diferenças entre os algoritmos em estudo. O nível de significância estatística foi também fixado em 95%.

Os resultados obtidos estão descritos nas tabelas 5.9, 5.10 e 5.11, respectivamente, para os classificadores induzidos pelos algoritmos SVMs, C4.5 e RIPPER, em que é pos-

Tabela 5.6: Resumo dos resultados obtidos pelo classificador SVM.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Golub64	Colon16
AllKNN	1		□		△	□	□	□
	3		□			□	□	□
	9	△	□			□	□	□
ENN	1	△	□			□	□	□
	3		□		△	□	□	□
	9		□			□	□	□
RENN	1	△	□			□	□	□
	3		□		△		□	□
	9		□				□	□
DROP2	1				▲	□		
	9					□		
DROP3	3			▲		□		
DROP4	3			▲				
	9					□		
DROP5	1			▲	△			
	3				▲	□		
	9			▲		□		
DEL	1				▲			
	3				▲			

Tabela 5.7: Resumo dos resultados obtidos pelo classificador C4.5.

	k	Leukemia	Lung	Golub	ExpGen	Golub64	Colon16
AllKNN	1	△	▲		△		
	3	△	▲	▲	△		▲
	9	△	▲	▲			▲
ENN	1	△	▲	△	△		▲
	3	▲	▲	▲		△	▲
	9	△	△	▲			▲
RENN	1	△	▲	△		▲	▲
	3	▲	▲	▲		▲	
	9	▲	△				
DROP1	1					△	
	9			△		▲	
DROP2	3					▲	
	9			▲		▲	
DROP3	1					△	
DROP4	1					△	
DROP5	1					▲	
	9			▲			
DEL	3					▲	
	9					▲	

Tabela 5.8: Resumo dos resultados obtidos pelo classificador RIPPER.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
AllKNN	1		▲	▲	△	△	▲	▲	▲
	3		▲	△	△	△	▲	▲	△
	9		▲	△			△	▲	
ENN	1	△	▲	▲	▲		▲	△	△
	3	△	▲	△		△	△		△
	9		▲	△			△	△	
RENN	1		▲	▲	△	△	▲	△	▲
	3	△	△	▲	△		▲	△	▲
	9		△					▲	
DROP1	9							△	
DEL	3					▲			
	9							▲	

sível observar que nenhuma das técnicas de detecção de ruído avaliadas obteve resultados superiores aos obtidos para os dados originais com significância estatística. Nessas tabelas, resultados inferiores aos obtidos para os dados originais são representados por ▼; ▽ indica resultados inferiores que obtiveram significância estatística; △ indica resultados semelhantes, sejam levemente superiores ou inferiores; resultados superiores são representados por ▲, ao passo que □ indica resultados iguais.

Tabela 5.9: Resultado da análise estatística do classificador SVM pelo teste de comparações múltiplas de Dunn.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
AllKNN	1	▼	□	▼	△	□	▼	□	□
	3	▼	□	▼	▼	□	▼	□	□
	9	▼	□	▼	▼	□	▼	□	□
ENN	1	△	□	▼	▼	□	▼	□	□
	3	▼	□	▼	△	□	▼	□	□
	9	▼	□	▼	▼	▼	▼	□	□
RENN	1	△	□	▼	▼	□	▼	□	□
	3	▼	□	▼	△	▼	▼	□	□
	9	▽	□	▼	▼	▼	▽	□	□
DROP1	1	▼	▼	▼	▼	▼	▼	▼	▼
	3	▼	▽	▼	▼	▼	▽	▼	▼
	9	▼	▼	△	▼	▼	▼	▼	▼
DROP2	1	▼	▼	△	▲	□	▼	▼	▼
	3	▽	▽	▲	▼	▼	▼	▼	▼
	9	▼	▼	▲	▼	□	▼	▼	▼
DROP3	1	▽	▼	▼	▼	▼	▼	▼	▼
	3	▼	▼	▲	▼	□	▼	▼	▼
	9	▽	▼	▼	▼	▼	▽	▼	▼
DROP4	1	▽	▼	▼	▼	▼	▼	▼	▼
	3	▼	▼	▲	▼	▼	▼	▼	▼
	9	▼	▽	△	▼	□	▼	▼	▼
DROP5	1	▼	▼	▲	△	▼	▼	▼	▼
	3	▼	▼	△	▲	□	▼	▼	▼
	9	▼	▼	▲	▼	□	▼	▼	▼
DEL	1	▼	▽	▼	▲	▼	▼	▼	▼
	3	▽	▼	▼	▲	▼	▼	▼	▼
	9	▼	▼	▼	▼	▼	▽	▼	▼

Visando analisar a complexidade das técnicas de detecção de ruído avaliadas, seus tempos de execução foram calculados. Eles são apresentados na Tabela 5.12, em que é possível verificar um maior tempo computacional na execução das técnicas que eliminaram maior quantidade de ruídos, como as técnicas DROP1 a DROP5 e a técnica DEL, sendo que essa última apresenta tempo computacional superior ao apresentado pelas técnicas AllKNN, ENN e RENN, porém um pouco inferior ao apresentado pelas técnicas DROP1 a DROP5.

Nas tabelas 5.13, 5.14 e 5.15 são informados os tempos computacionais necessários para o treinamento dos classificadores induzidos pelos algoritmos SVMs, C4.5 e RIPPER, para os dados originais e pré-processados.

Tabela 5.10: Resultado da análise estatística do classificador C4.5 pelo teste de comparações múltiplas de Dunn.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
AllKNN	1	Δ	$\blacktriangle$	$\blacktriangledown$	Δ	$\blacktriangledown$	$\blacktriangledown$	Δ	Δ
	3	Δ	Δ	$\blacktriangle$	Δ	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangle$
	9	Δ	Δ	$\blacktriangle$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangle$
ENN	1	Δ	$\blacktriangle$	Δ	Δ	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangle$
	3	Δ	Δ	$\blacktriangle$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangle$
	9	Δ	Δ	$\blacktriangle$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangle$
RENN	1	$\blacktriangle$	$\blacktriangle$	Δ	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangle$
	3	Δ	Δ	$\blacktriangle$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	Δ
	9	Δ	Δ	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	Δ
DROP1	1	∇	∇	$\blacktriangledown$	∇	$\blacktriangledown$	∇	Δ	$\blacktriangledown$
	3	$\blacktriangledown$	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$
	9	$\blacktriangledown$	∇	Δ	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$
DROP2	1	∇	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$
	3	∇	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$
	9	$\blacktriangledown$	∇	$\blacktriangle$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$
DROP3	1	∇	∇	$\blacktriangledown$	∇	∇	$\blacktriangledown$	Δ	$\blacktriangledown$
	3	$\blacktriangledown$	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$
	9	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangledown$
DROP4	1	∇	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangledown$
	3	$\blacktriangledown$	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangledown$
	9	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangledown$
DROP5	1	∇	$\blacktriangledown$	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$
	3	∇	$\blacktriangledown$	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$
	9	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$
DEL	1	∇	∇	$\blacktriangledown$	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$
	3	$\blacktriangledown$	∇	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	Δ	$\blacktriangledown$
	9	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangledown$	∇	$\blacktriangledown$	$\blacktriangledown$	$\blacktriangle$	$\blacktriangledown$

De acordo com estas tabelas é possível observar que os tempos computacionais obtidos pelo classificador SVM, quando aplicado aos dados pré-processados, são inferiores para todos os conjuntos estudados e para a maioria das técnicas de detecção de ruído investigadas. Para o algoritmo C4.5, é possível observar que os tempos computacionais obtidos, quando aplicado aos dados pré-processados, são inferiores para todos os conjuntos e técnicas de detecção de ruído estudadas. O mesmo ocorre para o algoritmo RIPPER. Vale ressaltar que os tempos computacionais apresentados pelos três classificadores avaliados foram pequenos.

A complexidade dos modelos obtidos pelos classificadores avaliados é apresentada nas tabelas 5.16, 5.17 e 5.18.

Para a análise da complexidade do algoritmo RIPPER, foi observado o número de regras produzidas durante o treinamento dos dados. Um número menor de regras indica menor complexidade do modelo obtido.

Nessas tabelas é possível notar que, para os classificadores SVMs e C4.5, os modelos foram simplificados após a aplicação das técnicas de detecção de ruído, sendo que, tanto o tamanho médio das árvores produzidas pelo algoritmo C4.5, como o número médio de

Tabela 5.11: Resultado da análise estatística do classificador RIPPER pelo teste de comparações múltiplas de Dunn.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
AllKNN	1	▼	▲	▲	△	△	▲	▲	△
	3	▼	▲	△	△	▼	▲	▲	△
	9	▼	▲	△	▼	▼	△	▲	▼
ENN	1	△	▲	▲	▲	△	▲	△	△
	3	▼	▲	△	△	△	△	△	△
	9	▼	▲	△	▼	▼	△	△	▼
RENN	1	▼	▲	▲	△	▼	▲	△	▲
	3	△	△	▲	△	▼	▲	△	△
	9	▼	△	▼	△	▼	▼	△	▼
DROP1	1	▽	▼	▽	▽	▼	▼	▼	▼
	3	▽	▽	▽	▽	▼	▼	▼	▼
	9	▽	▽	▼	▼	▼	▼	△	▼
DROP2	1	▽	▽	▽	▽	▼	▼	▼	▼
	3	▽	▽	▽	▽	▼	△	▼	▼
	9	▼	▽	△	▼	▼	▼	▼	▼
DROP3	1	▽	▽	▼	▽	▼	▼	▼	▽
	3	▽	▽	▼	▼	▼	▼	▼	▽
	9	▼	▼	▼	▼	—	▼	△	▼
DROP4	1	▽	▽	▼	▽	▼	▼	▼	▽
	3	▽	▼	▼	▽	▼	▼	▼	▼
	9	▼	▼	▼	▼	▼	▼	△	▼
DROP5	1	▽	▼	▽	▽	▼	▼	▼	▽
	3	▽	▼	▼	▽	▼	▼	△	▼
	9	▼	▼	▼	▽	▼	▼	▼	▼
DEL	1	▽	▽	▽	▽	▼	▽	—	▼
	3	▽	▽	▽	▽	▲	▼	▼	▽
	9	▼	▼	▼	▼	▼	▼	▲	▼

Tabela 5.12: Tempos (em segundos) da aplicação das técnicas de detecção de ruído.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
AllKNN	1	3.128	4.683	2.523	0.393	1.231	108.049	0.052	0.016
	3	3.115	4.665	2.504	0.382	1.193	107.761	0.052	0.014
	9	3.113	4.664	2.486	0.392	1.168	107.366	0.052	0.015
ENN	1	4.661	6.365	2.807	0.486	1.688	122.457	0.059	0.020
	3	5.253	6.536	3.051	0.519	2.287	130.978	0.061	0.022
	9	6.795	7.349	5.452	0.625	3.272	189.457	0.066	0.029
RENN	1	4.737	6.381	2.794	0.485	1.711	122.951	0.061	0.020
	3	5.575	7.110	3.073	0.513	2.493	132.109	0.063	0.023
	9	7.013	7.730	6.596	0.753	3.511	264.557	0.065	0.030
DROP1	1	8.181	13.003	5.109	1.121	2.156	269.389	0.110	0.024
	3	13.163	20.754	7.415	1.720	3.069	411.882	0.165	0.033
	9	26.856	44.183	13.274	3.630	6.084	747.595	0.312	0.057
DROP2	1	6.713	9.727	4.193	0.925	1.926	236.814	0.111	0.025
	3	10.368	14.456	5.577	1.417	2.593	344.469	0.150	0.034
	9	19.818	29.685	9.821	2.849	4.326	587.938	0.228	0.048
DROP3	1	7.919	11.794	4.393	0.973	2.139	245.804	0.119	0.029
	3	11.520	17.343	6.159	1.434	3.382	347.836	0.161	0.039
	9	19.651	34.040	13.150	2.675	6.217	650.106	0.239	0.054
DROP4	1	7.560	11.502	4.390	0.970	1.927	242.672	0.119	0.029
	3	10.581	14.956	5.863	1.435	2.987	344.209	0.157	0.039
	9	19.923	29.658	10.226	2.752	5.007	606.625	0.239	0.054
DROP5	1	12.435	22.262	6.543	1.906	2.292	339.850	0.151	0.028
	3	19.904	33.384	9.537	2.921	3.689	497.269	0.238	0.046
	9	42.502	68.721	16.154	5.558	7.262	879.981	0.413	0.070
DEL	1	8.519	11.943	5.060	1.115	2.134	278.872	0.125	0.028
	3	13.125	18.772	6.913	1.628	2.726	410.059	0.173	0.041
	9	22.780	33.107	10.981	2.964	4.608	733.975	0.255	0.058

Tabela 5.13: Tempos de treinamento (em segundos) das SVMs aplicadas sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
Originais		0.691	1.018	0.641	0.368	0.434	11.762	0.180	0.185
AllKNN	1	0.623	0.874	0.574	0.366	0.392	9.699	0.191	0.208
	3	0.614	0.868	0.552	0.367	0.367	9.415	0.193	0.214
	9	0.607	0.862	0.537	0.352	0.359	8.641	0.192	0.213
ENN	1	0.622	0.881	0.578	0.368	0.376	9.765	0.190	0.204
	3	0.613	0.884	0.562	0.345	0.367	9.625	0.190	0.203
	9	0.607	0.888	0.537	0.351	0.358	9.186	0.190	0.204
RENN	1	0.621	0.876	0.576	0.366	0.373	9.726	0.190	0.205
	3	0.605	0.868	0.559	0.341	0.351	9.581	0.191	0.204
	9	0.605	0.859	0.464	0.346	0.339	8.240	0.190	0.205
DROP1	1	0.880	0.695	0.467	0.782	0.423	1.286	0.615	0.519
	3	0.814	0.611	0.403	0.682	0.343	1.615	0.457	0.444
	9	0.694	0.548	0.385	0.580	0.391	2.492	0.314	0.291
DROP2	1	0.744	0.697	0.385	0.655	0.350	1.697	0.544	0.335
	3	0.734	0.593	0.381	0.645	0.336	2.170	0.442	0.355
	9	0.661	0.542	0.393	0.558	0.345	3.281	0.315	0.238
DROP3	1	0.833	0.717	3.894	0.662	0.381	1.655	0.554	0.650
	3	0.742	0.582	3.712	0.690	0.382	2.112	0.460	0.441
	9	0.684	0.541	3.406	0.558	1.053	2.422	0.308	0.299
DROP4	1	0.791	0.699	3.527	0.646	0.355	1.694	0.552	0.586
	3	0.761	0.584	3.309	0.664	0.328	2.122	0.438	0.433
	9	0.672	0.543	3.388	0.560	0.321	3.044	0.310	0.299
DROP5	1	0.760	0.649	3.498	0.724	0.377	1.550	0.612	0.677
	3	0.693	0.580	3.249	0.659	0.306	1.849	0.445	0.450
	9	0.669	0.541	3.073	0.582	0.336	2.777	0.308	0.295
DEL	1	0.773	0.735	4.175	0.654	0.328	1.076	1.133	0.378
	3	0.776	0.691	3.138	0.655	0.335	1.230	0.611	0.594
	9	0.695	0.568	2.007	0.567	0.330	2.028	0.347	0.330

Tabela 5.14: Tempos (em segundos) das ADs aplicadas sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
Originais		0.634	1.099	0.435	0.075	0.402	14.047	0.010	0.005
AllKNN	1	0.450	0.692	0.358	0.044	0.209	11.362	0.009	0.003
	3	0.392	0.689	0.352	0.042	0.178	11.106	0.008	0.002
	9	0.369	0.671	0.293	0.036	0.111	8.195	0.008	0.002
ENN	1	0.434	0.699	0.353	0.049	0.201	11.419	0.008	0.003
	3	0.411	0.703	0.358	0.045	0.173	12.061	0.009	0.004
	9	0.373	0.684	0.289	0.042	0.117	8.511	0.008	0.004
RENN	1	0.428	0.682	0.354	0.047	0.177	11.617	0.008	0.004
	3	0.378	0.688	0.347	0.044	0.090	11.018	0.009	0.002
	9	0.339	0.659	0.161	0.040	0.052	7.983	0.010	0.002
DROP1	1	0.035	0.035	0.051	0.005	0.038	1.092	0.002	0.002
	3	0.042	0.049	0.065	0.006	0.061	1.391	0.002	0.002
	9	0.064	0.084	0.109	0.009	0.073	2.369	0.003	0.002
DROP2	1	0.049	0.039	0.069	0.006	0.047	1.473	0.002	0.003
	3	0.057	0.054	0.084	0.008	0.066	1.990	0.002	0.002
	9	0.087	0.088	0.115	0.011	0.093	3.570	0.002	0.002
DROP3	1	0.039	0.039	0.066	0.005	0.043	1.381	0.002	0.002
	3	0.047	0.053	0.070	0.006	0.042	1.882	0.002	0.002
	9	0.058	0.091	0.086	0.009	0.020	1.994	0.002	0.003
DROP4	1	0.043	0.040	0.067	0.006	0.049	1.459	0.003	0.001
	3	0.057	0.055	0.079	0.070	0.058	1.983	0.002	0.001
	9	0.083	0.088	0.107	0.010	0.070	2.791	0.002	0.002
DROP5	1	0.047	0.044	0.065	0.006	0.045	1.313	0.002	0.001
	3	0.057	0.057	0.079	0.007	0.061	1.561	0.002	0.002
	9	0.077	0.093	0.116	0.009	0.078	2.567	0.002	0.002
DEL	1	0.055	0.034	0.050	0.009	0.075	0.897	0.001	0.002
	3	0.043	0.041	0.061	0.008	0.093	1.026	0.003	0.003
	9	0.071	0.075	0.103	0.009	0.106	1.818	0.002	0.002

Tabela 5.15: Tempos (em segundos) do algoritmo RIPPER aplicado sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
Originais		1.389	1.670	0.777	0.170	3.000	17.318	0.023	0.018
AllKNN	1	0.942	1.012	0.582	0.111	0.303	13.530	0.022	0.013
	3	0.861	0.953	0.507	0.101	0.213	12.083	0.021	0.006
	9	0.799	0.982	0.443	0.085	0.133	10.262	0.023	0.008
ENN	1	1.011	1.019	0.560	0.120	0.264	13.759	0.021	0.008
	3	0.973	1.044	0.521	0.108	0.202	12.638	0.026	0.012
	9	0.842	1.089	0.476	0.089	0.128	10.574	0.021	0.008
RENN	1	0.933	0.933	0.541	0.111	0.247	13.585	0.022	0.004
	3	0.848	1.056	0.513	0.102	0.121	12.575	0.022	0.002
	9	0.690	1.004	0.202	0.087	0.031	8.338	0.023	0.006
DROP1	1	0.048	0.047	0.059	0.010	0.043	0.939	0.015	0.002
	3	0.068	0.096	0.085	0.014	0.067	1.378	0.004	0.001
	9	0.079	0.109	0.155	0.022	0.081	2.727	0.006	0.001
DROP2	1	0.070	0.051	0.092	0.014	0.053	1.423	0.001	0.004
	3	0.082	0.086	0.096	0.021	0.072	1.882	0.004	0.002
	9	0.121	0.123	0.149	0.024	0.109	3.435	0.012	0.004
DROP3	1	0.048	0.054	0.076	0.012	0.053	1.341	0.002	0.001
	3	0.077	0.086	0.084	0.016	0.049	1.951	0.001	0.004
	9	0.057	0.121	0.146	0.019	—	2.071	0.010	0.002
DROP4	1	0.057	0.052	0.088	0.016	0.066	1.397	0.006	0.001
	3	0.076	0.084	0.098	0.021	0.070	1.834	0.008	0.002
	9	0.123	0.118	0.138	0.044	0.082	3.071	0.010	0.004
DROP5	1	0.061	0.062	0.096	0.020	0.056	1.356	0.002	0.008
	3	0.079	0.080	0.091	0.020	0.070	1.446	0.008	0.002
	9	0.109	0.121	0.188	0.041	0.094	2.659	0.008	0.004
DEL	1	0.087	0.046	0.059	0.022	0.077	0.815	—	0.006
	3	0.072	0.055	0.078	0.021	0.100	0.985	0.001	0.001
	9	0.093	0.090	0.141	0.022	0.127	1.891	0.004	0.001

Tabela 5.16: Número médio de SVs obtidos pelas SVMs aplicadas sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
Originais		49.6+-0.94	177.3+-0.48	64.8+-0.42	35.7+-1.09	55.8+-0.42	222.2+-1.68	64.8+-0.42	55.8+-0.42
AllKNN	1	40.0+-0.90	151.1+-2.08	54.6+-0.70	26.9+-1.78	38.8+-2.97	191.1+-2.92	57.1+-0.87	40.2+-1.87
	3	38.1+-1.56	150.7+-1.94	51.9+-1.91	24.7+-2.25	34.3+-2.67	186.6+-2.95	56.5+-1.43	38.4+-2.50
	9	37.4+-1.37	149.6+-2.17	49.0+-2.16	22.5+-1.85	29.9+-2.51	173.5+-2.32	56.3+-1.57	38.0+-2.83
ENN	1	40.3+-0.68	152.7+-1.94	54.8+-0.92	27.3+-1.90	39.4+-3.10	191.8+-2.61	57.9+-0.57	41.8+-1.47
	3	38.1+-1.21	153.5+-1.08	53.0+-1.56	25.3+-2.88	35.4+-1.43	189.8+-2.74	57.6+-0.96	43.2+-1.31
	9	36.5+-1.35	152.5+-1.35	49.1+-2.51	22.9+-1.07	27.5+-14.56	182.2+-2.35	57.9+-0.57	43.0+-1.34
RENN	1	40.0+-0.84	152.3+-2.62	54.8+-0.92	26.3+-2.18	38.3+-4.24	191.4+-2.95	57.8+-0.63	41.2+-1.55
	3	37.5+-2.61	150.6+-2.87	52.7+-1.89	24.1+-2.30	19.0+-16.43	188.5+-3.53	57.3+-0.95	42.4+-1.43
	9	33.1+-1.12	147.6+-14.81	15.0+-24.21	22.3+-2.11	30.2+-0.00	167.1+-10.23	57.9+-0.57	42.1+-1.85
DROP1	1	12.8+-1.21	8.0+-0.00	7.0+-1.15	9.7+-1.26	8.3+-1.34	28.0+-3.20	4.0+-0.00	5.5+-2.01
	3	14.7+-0.79	12.1+-1.10	9.8+-2.15	11.4+-1.74	14.4+-3.72	36.7+-4.90	6.1+-0.31	6.5+-0.53
	9	19.5+-0.63	22.8+-0.42	19.1+-2.13	14.4+-2.26	16.8+-6.21	58.2+-5.33	12.0+-0.00	12.3+-0.48
DROP2	1	15.7+-1.26	9.3+-1.49	10.7+-1.49	11.0+-1.38	11.0+-1.49	38.7+-2.54	4.8+-0.63	10.3+-1.42
	3	17.5+-0.63	13.8+-1.47	13.8+-1.68	13.6+-0.64	14.8+-2.70	50.6+-4.11	6.6+-1.07	9.8+-1.55
	9	22.2+-1.71	23.9+-1.28	20.3+-1.49	17.0+-1.26	21.6+-2.83	75.8+-4.92	12.0+-0.00	14.6+-0.70
DROP3	1	13.8+-1.30	9.3+-1.06	10.1+-1.59	9.8+-1.77	10.4+-1.58	37.9+-3.66	4.7+-0.67	4.1+-1.20
	3	16.7+-0.70	13.3+-0.82	10.7+-2.06	10.6+-1.61	10.2+-3.15	49.1+-3.81	6.1+-0.57	6.5+-0.71
	9	17.5+-1.77	25.1+-0.87	13.9+-1.73	12.6+-1.43	2.2+-4.26	56.7+-5.29	12.1+-0.31	12.2+-0.42
DROP4	1	14.5+-1.24	9.7+-0.82	10.1+-1.59	10.2+-1.43	11.9+-1.85	39.0+-2.21	4.7+-0.67	4.5+-0.97
	3	17.4+-0.90	14.1+-1.52	12.8+-1.62	12.2+-2.10	14.1+-2.08	49.5+-4.14	6.7+-1.06	6.9+-1.10
	9	21.8+-1.30	24.1+-0.99	18.7+-1.94	14.7+-1.56	16.7+-1.77	70.8+-5.33	12.1+-0.31	12.7+-0.95
DROP5	1	16.2+-0.74	11.2+-1.81	9.7+-1.49	11.6+-1.46	11.0+-2.00	35.1+-3.75	4.0+-0.00	3.9+-1.20
	3	17.8+-0.59	14.6+-1.50	12.6+-1.43	13.4+-1.57	15.0+-1.49	42.8+-3.26	6.4+-0.51	6.3+-0.48
	9	22.1+-1.51	25.2+-2.15	20.6+-2.36	16.3+-2.33	18.3+-2.45	65.0+-5.96	12.1+-0.31	12.4+-0.96
DEL	1	18.2+-1.70	8.0+-1.70	6.9+-2.33	15.2+-0.74	15.3+-2.21	22.4+-2.01	2.0+-0.00	9.4+-4.30
	3	15.2+-0.75	9.7+-0.95	9.0+-2.00	14.8+-1.12	18.7+-3.59	26.6+-5.62	4.0+-0.00	4.2+-0.42
	9	20.3+-0.76	20.0+-1.49	17.8+-3.39	14.4+-3.15	22.1+-5.17	47.0+-8.90	10.0+-0.00	10.5+-1.08

Tabela 5.17: Tamanho médio das árvores produzidas pelo algoritmo C4.5 aplicado sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
Originais		34.20+-2.15	9.40+-0.84	4.40+-0.96	14.00+-1.41	6.80+-0.63	10.20+-1.68	4.20+-1.03	8.80+-2.74
AllKNN	1	24.60+-2.27	7.00+-0.00	4.00+-1.05	9.00+-0.94	4.40+-0.96	8.40+-2.99	4.00+-1.05	3.20+-0.63
	3	21.40+-1.84	7.00+-0.00	3.40+-0.84	7.80+-1.03	3.40+-0.84	6.40+-2.12	3.40+-0.84	3.00+-0.00
	9	18.80+-1.47	7.00+-0.00	3.40+-0.84	7.00+-0.00	2.80+-0.63	4.40+-2.12	3.40+-0.84	3.00+-0.00
ENN	1	23.80+-1.68	7.20+-0.63	4.00+-1.05	8.60+-0.84	4.40+-0.96	8.40+-2.99	4.00+-1.05	3.40+-0.84
	3	21.60+-1.35	7.00+-0.00	3.40+-0.84	8.60+-0.84	3.60+-0.96	6.60+-2.06	3.40+-0.84	3.00+-0.00
	9	19.00+-1.34	7.00+-0.00	3.40+-0.84	8.20+-1.40	2.20+-1.03	5.40+-1.84	4.00+-1.05	3.20+-0.63
RENN	1	22.40+-2.12	7.00+-0.00	4.00+-1.05	8.60+-0.84	4.40+-0.96	7.80+-3.01	4.00+-1.05	3.20+-0.63
	3	19.80+-1.40	7.00+-0.00	3.40+-0.84	8.40+-0.96	2.20+-1.03	6.80+-1.99	3.40+-0.84	3.00+-0.00
	9	17.00+-0.94	6.80+-0.63	2.00+-1.70	8.00+-1.41	1.00+-0.00	4.20+-1.93	4.00+-1.05	3.00+-0.00
DROP1	1	13.00+-0.00	7.00+-0.00	3.00+-0.00	8.60+-0.84	3.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00
	3	13.00+-0.00	6.80+-0.63	3.00+-0.00	8.20+-1.03	3.00+-0.00	3.20+-0.63	3.00+-0.00	3.00+-0.00
	9	13.00+-0.00	7.00+-0.00	3.00+-0.00	7.80+-1.03	2.80+-0.63	3.80+-1.03	3.00+-0.00	3.00+-0.00
DROP2	1	13.00+-0.00	6.00+-1.41	3.00+-0.00	8.80+-0.63	3.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00
	3	13.00+-0.00	7.00+-0.00	3.00+-0.00	9.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00
	9	13.60+-0.96	7.00+-0.00	3.00+-0.00	8.40+-0.96	3.00+-0.00	3.60+-0.96	3.00+-0.00	3.20+-0.63
DROP3	1	12.60+-0.84	6.60+-0.84	3.00+-0.00	8.20+-1.03	3.00+-0.00	3.00+-0.00	3.00+-0.00	2.20+-1.03
	3	13.00+-0.00	7.00+-0.00	3.00+-0.00	7.40+-0.84	3.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00
	9	11.20+-0.63	7.00+-0.00	3.00+-0.00	6.80+-0.63	1.20+-0.63	3.00+-0.00	3.00+-0.00	3.00+-0.00
DROP4	1	13.00+-0.00	6.80+-0.63	3.00+-0.00	8.20+-1.03	3.00+-0.00	3.00+-0.00	3.00+-0.00	2.80+-0.63
	3	13.00+-0.00	7.00+-0.00	3.00+-0.00	8.20+-1.03	3.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00
	9	13.20+-0.63	7.00+-0.00	3.00+-0.00	7.60+-0.96	3.00+-0.00	3.80+-1.03	3.00+-0.00	3.00+-0.00
DROP5	1	13.00+-0.00	6.60+-0.84	3.00+-0.00	8.60+-0.84	3.00+-0.00	3.20+-0.63	3.00+-0.00	2.00+-1.05
	3	13.00+-0.00	7.00+-0.00	3.00+-0.00	8.40+-0.96	3.00+-0.00	3.00+-0.00	3.00+-0.00	3.00+-0.00
	9	13.20+-0.63	7.00+-0.00	3.00+-0.00	8.20+-1.03	3.00+-0.00	3.20+-0.63	3.00+-0.00	3.00+-0.00
DEL	1	12.80+-1.13	4.20+-1.03	2.80+-0.63	7.40+-0.84	3.00+-0.00	3.00+-0.00	1.00+-0.00	3.20+-1.13
	3	13.00+-0.00	7.00+-0.00	3.00+-0.00	8.80+-0.63	3.80+-1.03	3.00+-0.00	3.00+-0.00	3.00+-0.00
	9	13.00+-0.00	7.00+-0.00	3.00+-0.00	7.80+-1.03	3.80+-1.03	3.40+-0.84	3.00+-0.00	3.00+-0.00

Tabela 5.18: Número médio de regras produzidas pelo algoritmo RIPPER aplicado sobre os dados originais e pré-processados.

	k	Leukemia	Lung	Golub	ExpGen	Colon	Ovarian	Golub64	Colon16
Originais		10.80+-1.40	5.00+-0.67	2.10+-0.31	5.70+-0.67	3.00+-0.67	3.10+-0.31	2.10+-0.31	2.60+-0.51
AllKNN	1	9.30+-1.06	4.10+-0.31	2.20+-0.42	4.80+-0.42	2.40+-0.51	2.90+-0.31	2.10+-0.31	2.10+-0.31
	3	8.70+-1.06	4.00+-0.00	2.30+-0.48	4.70+-0.67	2.40+-0.51	2.40+-0.51	2.10+-0.31	2.00+-0.00
	9	8.00+-0.94	4.00+-0.00	2.10+-0.31	4.10+-0.57	1.60+-0.51	2.20+-0.42	2.20+-0.42	2.00+-0.00
ENN	1	9.60+-1.07	4.10+-0.31	2.20+-0.42	5.10+-0.74	2.30+-0.48	2.80+-0.42	2.10+-0.31	2.20+-0.42
	3	9.10+-1.45	4.10+-0.31	2.20+-0.42	4.70+-0.67	2.10+-0.31	2.60+-0.70	2.10+-0.31	2.10+-0.31
	9	7.80+-0.79	4.20+-0.42	2.20+-0.00	4.40+-0.84	1.20+-0.42	2.10+-0.31	2.10+-0.31	2.20+-0.42
RENN	1	9.60+-0.96	4.00+-0.00	2.20+-0.00	5.20+-0.79	2.40+-0.51	2.80+-0.42	2.10+-0.31	2.20+-0.42
	3	8.80+-0.92	4.10+-0.31	2.10+-0.31	4.70+-0.67	1.30+-0.48	2.60+-0.51	2.10+-0.31	2.00+-0.00
	9	7.90+-0.56	4.10+-0.57	1.30+-0.48	4.20+-0.63	1.00+-0.00	2.30+-0.48	2.10+-0.31	2.10+-0.31
DROP1	1	5.60+-1.26	1.00+-0.00	1.10+-0.31	2.90+-1.10	1.10+-0.31	2.00+-0.00	1.00+-0.00	1.40+-0.51
	3	5.50+-1.08	2.40+-0.70	1.20+-0.42	3.40+-0.84	1.60+-0.51	2.30+-0.48	1.80+-0.42	1.60+-0.51
	9	7.30+-0.48	4.00+-0.47	1.90+-0.31	4.00+-1.15	1.80+-0.42	2.20+-0.42	2.00+-0.00	1.90+-0.31
DROP2	1	6.40+-0.70	1.30+-0.48	1.50+-0.53	3.90+-0.74	1.40+-0.51	2.20+-0.42	1.10+-0.31	1.80+-0.42
	3	6.90+-0.31	2.50+-0.71	1.30+-0.48	4.20+-0.42	1.90+-0.31	2.20+-0.42	1.60+-0.51	1.70+-0.48
	9	7.10+-0.31	3.80+-0.42	2.00+-0.00	4.60+-0.70	2.00+-0.00	2.30+-0.48	2.00+-0.00	2.00+-0.00
DROP3	1	5.70+-1.16	1.10+-0.31	1.40+-0.51	3.10+-1.20	1.50+-0.53	2.00+-0.00	1.10+-0.31	1.00+-0.00
	3	6.40+-0.70	2.90+-0.57	1.40+-0.51	3.80+-0.63	1.60+-0.51	2.00+-0.00	1.80+-0.42	1.40+-0.51
	9	5.70+-1.06	4.10+-0.31	2.00+-0.00	3.90+-0.31	-----	2.20+-0.42	2.00+-0.00	2.00+-0.00
DROP4	1	5.70+-1.06	1.20+-0.42	1.40+-0.51	3.20+-1.13	1.70+-0.48	2.00+-0.00	1.10+-0.31	1.12+-0.34
	3	6.60+-0.84	3.10+-0.74	1.40+-0.51	4.30+-0.67	1.80+-0.42	2.10+-0.31	1.70+-0.48	1.50+-0.71
	9	7.00+-0.00	4.10+-0.31	2.00+-0.00	4.30+-0.48	1.80+-0.42	2.20+-0.42	2.00+-0.00	2.00+-0.00
DROP5	1	6.10+-1.20	1.20+-0.42	1.40+-0.51	3.30+-1.06	1.30+-0.48	2.00+-0.00	1.00+-0.00	1.00+-0.00
	3	6.60+-0.51	2.70+-0.95	1.80+-0.42	3.40+-0.84	1.90+-0.31	2.10+-0.31	2.00+-0.00	1.80+-0.42
	9	7.40+-0.51	4.00+-0.00	2.00+-0.00	4.40+-0.51	1.90+-0.31	2.20+-0.42	2.00+-0.00	2.00+-0.00
DEL	1	4.40+-0.84	1.00+-0.00	1.00+-0.00	2.60+-0.84	1.40+-0.51	2.00+-0.00	-----	1.60+-0.51
	3	5.90+-0.57	1.00+-0.00	1.10+-0.31	3.80+-0.79	1.90+-0.31	2.00+-0.00	1.00+-0.00	1.10+-0.31
	9	7.00+-0.00	3.70+-0.48	1.90+-0.31	4.40+-0.51	2.00+-0.47	2.50+-0.53	2.00+-0.00	2.00+-0.00

SVs obtidos pelas SVMs, foram reduzidos.

Para o algoritmo RIPPER, os modelos finais obtidos também foram simplificados. Vale observar que, para os conjuntos *Golub* e *Golub64*, a redução da complexidade foi inferior à obtida pelos demais conjuntos, não ocorrendo para algumas das técnicas analisadas.

5.5 Considerações Finais

Neste capítulo foram apresentados os novos conjuntos de dados selecionados, assim como os diferentes experimentos realizados.

As técnicas de detecção de ruído avaliadas compõem basicamente dois grupos distintos. O primeiro, constituído pelas técnicas AllKNN, ENN e RENN, detecta e elimina uma menor quantidade de dados; o segundo, formado pelas técnicas DROPs 1 a 5 e DEL, elimina uma quantidade significativamente superior de dados, uma vez que são eliminados não apenas possíveis ruídos, mas também dados redundantes.

Apesar dessa distinção, não se pode precisamente afirmar que um deles seja superior ao outro, uma vez que o desempenho obtido pelos diferentes classificadores foi variado, resultando em melhoria, manutenção ou deterioração dos resultados produzidos de acordo com o conjunto de dados ou técnica avaliada. Porém, de um modo geral, é possível afirmar que o primeiro grupo é mais conservador e produziu resultados superiores ao segundo (taxas de erro e desvio padrão inferiores). Mesmo que, em algumas situações, técnicas pertencentes ao segundo grupo tenham se destacado e apresentado desempenho superior, essa não foi geralmente a regra observada.

Com base na avaliação dos experimentos e resultados obtidos, foram investigadas diferentes abordagens para a combinação de técnicas de detecção de ruído, que são apresentadas no próximo capítulo.

Experimentos realizados com a Combinação

6.1 *Considerações Iniciais*

Neste capítulo são discutidos os experimentos realizados com as combinações de técnicas de detecção de ruído, investigadas a partir dos resultados obtidos nos experimentos previamente realizados. Na Seção 6.2, são detalhados os resultados obtidos com os diferentes tipos de combinação desenvolvidas. As Considerações Finais são apresentadas na Seção 6.3.

6.2 *Combinação de Técnicas de Detecção de Ruído*

Todos os módulos de combinação desenvolvidos consistem em combinações das técnicas de detecção de ruído analisadas individualmente ao longo do trabalho realizado. O objetivo foi a obtenção de combinações que apresentem bom desempenho (pequenas taxas de erro) para os conjuntos de dados analisados neste trabalho, de modo a demonstrar que as técnicas de detecção de ruído baseadas em distância utilizadas apresentam comportamento adequado diante de diferentes tipos de dados que tratam problemas de expressão gênica.

Foram desenvolvidas combinações para cada um dos conjuntos de dados em que as técnicas de detecção de ruído apresentaram, para os dados pré-processados, resultados

superiores ou semelhantes aos obtidos sobre os dados originais.

As técnicas de detecção de ruído foram combinadas inicialmente pela metodologia de votação por maioria, em que cada dado é considerado ruído se, para n técnicas avaliadas, ao menos m delas, em que m é maior que $n/2$, classificaram o dado como ruidoso. É possível notar que a metodologia adotada não é conservadora, de modo a garantir que uma maior quantidade de dados seja considerada ruidosa e possa, assim, ser eliminada do conjunto de dados final resultante da combinação desenvolvida.

Os experimentos seguiram a mesma metodologia anteriormente aplicada para a realização deste trabalho. Para prover estimativas dos erros dos classificadores induzidos pelas combinações, os conjuntos de dados foram divididos segundo o método *10-fold cross validation* (MITCHELL [1997]).

Dentre as técnicas de detecção de ruído avaliadas, basicamente dois grupos distintos podem ser observados: o primeiro, constituído pelas técnicas AllKNN, ENN e RENN, que detecta e elimina uma menor quantidade de dados; e o segundo, constituído pelas técnicas DEL e DROPs 1 a 5, que detecta e elimina uma quantidade significativamente superior de dados, já que são eliminados também possíveis dados redundantes e não apenas ruído. Os exemplos eliminados pelas técnicas pertencentes ao segundo grupo englobam a maioria dos exemplos detectados pelas técnicas existentes no primeiro grupo.

O objetivo em desenvolver combinações que não sejam conservadoras foi de garantir que a maioria dos dados ruidosos fosse eliminada do conjunto de dados final, sejam eles provenientes do primeiro ou segundo grupo de técnicas de detecção de ruído avaliadas.

Todas as técnicas de detecção de ruído que apresentaram desempenho superior ou semelhante (levemente superior) aos obtidos para os dados originais foram utilizadas nas combinações desenvolvidas.

Nesse âmbito, para o classificador SVM, apenas três combinações foram elaboradas, como segue:

- *Leukemia*, composta pelas técnicas ENN e RENN com $k=1$ e AllKNN com $k=9$;
- *Golub*, constituída pelas técnicas DROPs 2, 3 e 4 com $k=3$, e DROP5 com $k=1$ e $k=9$;
- *ExpGen*, composta pelas técnicas DROP2 com $k=1$, DROP5 com $k=3$ e DEL com $k=1$ e $k=3$.

Para os demais conjuntos, os resultados obtidos foram inferiores e, portanto, não foram desenvolvidas combinações.

Para o classificador C4.5, seis combinações foram elaboradas. Como já mencionado, apenas para os conjuntos em que os resultados obtidos foram semelhantes ou superiores, a saber:

- *Leukemia*, composta pelas técnicas ENN e RENN com $k=1$, $k=3$ e $k=9$, e AllKNN com $k=3$ e $k=9$;
- *Lung*, composta pelas técnicas AllKNN, ENN e RENN com $k=1$, $k=3$ e $k=9$;
- *Golub*, constituída pelas técnicas AllKNN e ENN com $k=3$ e $k=9$, RENN com $k=3$ e DROPs 2 e 5 com $k=9$;
- *ExpGen*, formada pelas técnicas AllKNN com $k=1$ e $k=3$ e ENN com $k=1$;
- *Golub64*, composta pelas técnicas RENN com $k=3$, DROPs 1, 2 e DEL com $k=9$ e DROP5 com $k=1$;
- *Colon16*, constituída pelas técnicas AllKNN com $k=3$ e $k=9$ e ENN com $k=1$, $k=3$ e $k=9$.

Já para o classificador RIPPER, sete diferentes combinações foram desenvolvidas, descritas abaixo:

- *Lung*, constituída pelas técnicas AllKNN e ENN com $k=1$, $k=3$ e $k=9$ e RENN com $k=1$;
- *Golub*, formada pelas técnicas AllKNN, ENN e RENN com $k=1$ e $k=3$;
- *ExpGen*, composta pelas técnicas AllKNN com $k=1$ e $k=3$ e ENN com $k=1$;
- *Colon*, formada pelas técnicas AllKNN com $k=1$ e $k=3$, ENN e DEL com $k=3$;
- *Ovarian*, constituída pelas técnicas AllKNN, ENN e RENN com $k=1$ e $k=3$;
- *Golub64*, formada pelas técnicas AllKNN com $k=1$, $k=3$ e $k=9$, RENN, DROP1 e DEL com $k=9$;
- *Colon16*, composta pelas técnicas AllKNN e RENN com $k=1$ e $k=3$.

Para esse algoritmo, apenas o conjunto de dados *Leukemia* apresentou resultados inferiores para os dados pré-processados. Sendo assim, não foram desenvolvidas combinações para esse conjunto.

As tabelas 6.1, 6.2 e 6.3 apresentam as combinações desenvolvidas, respectivamente, para os classificadores induzidos pelos algoritmos SVM, C4.5 e RIPPER. Nessas tabelas, a coluna "Melhor Desempenho" indica a técnica de detecção de ruído que obteve os melhores resultados quando aplicada individualmente, ao passo que a coluna "Redução (%)" indica a porcentagem de dados existentes no conjunto de dados original que foi considerada ruído e, conseqüentemente, eliminada, pela combinação desenvolvida.

A coluna intitulada "Desempenho Estatístico" apresenta os resultados obtidos pela aplicação do teste estatístico de Friedman, com 95% de confiança estatística, seguido pelo teste de comparações múltiplas de Dunn, para analisar o desempenho obtido pelas combinações. O nível de significância estatística para esse teste foi também fixado em 95%.

Para o teste estatístico aplicado, resultados inferiores ao melhor resultado obtido, apresentado na coluna "Melhor Desempenho", são representados por ▼; ∇ indica resultados inferiores que obtiveram significância estatística; Δ indica resultados semelhantes, levemente superiores ou inferiores; resultados superiores são representados por ▲; ao passo que □ indica resultados iguais.

Tabela 6.1: Resultados obtidos pelas combinações desenvolvidas para o classificador SVM.

Conjunto de Dados	Número de técnicas	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Leukemia	3	7.34+-3.65	40.11+-0.88	7.34+-3.65 ENN $k=1$	16.92	□
Golub	5	42.68+-22.40	9.90+-2.23	21.07+-13.72 DROP4 $k=3$	84.73	∇
ExpGen	4	6.21+-5.51	13.46+-1.50	5.74+-4.37 DEL $k=3$	90.39	▼

A comparação dos resultados apresentados na Tabela 6.1 com aqueles encontrados na Tabela 5.3, que possui as taxas de erro do classificador SVM para cada uma das técnicas de detecção de ruído avaliadas, indica que as combinações desenvolvidas conferiram, de modo geral, resultados diversos se comparados aos obtidos para os dados originais, e nem sempre produziram os melhores resultados para cada conjunto avaliado.

Apenas a desenvolvida para o conjunto *Leukemia* apresentou resultado idêntico ao melhor resultado existente para esse conjunto, que foi obtido para a técnica ENN com

$k=1$ e para os dados originais. A complexidade encontrada foi também semelhante à da técnica ENN com $k=1$, porém inferior à encontrada para os dados originais.

A combinação para o conjunto *Golub* deteriorou significativamente o resultado final obtido, sendo essa a única que obteve resultado inferior ao obtido para os dados originais. A desenvolvida para o conjunto *ExpGen* equiparou-se aos melhores resultados obtidos nesse conjunto, sendo superior ao desempenho obtido com os dados originais, porém inferior ao melhor resultado encontrado, obtido para a técnica DEL com $k=3$. Para esses dois conjuntos as complexidades obtidas foram sempre inferiores àquelas apresentadas para os dados originais.

Para a determinação da complexidade das SVMs foi considerado o número de vetores de suporte (*support vectors* - *SVs*) do modelo induzido. Um número menor de SVs indica menor complexidade do modelo obtido.

Tabela 6.2: Resultados obtidos pelas combinações desenvolvidas para o classificador C4.5.

Conjunto de Dados	Número de técnicas	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Leukemia	8	15.60+-6.74	20.80+-1.47	16.81+-4.98 RENN $k=9$	17.09	▲
Lung	9	6.58+-4.78	7.00+-0.00	5.56+-4.97 ENN $k=1$	14.49	▼
Golub	7	13.58+-12.69	3.40+-0.84	11.44+-11.28 DROP5 $k=9$	18.05	Δ
ExpGen	3	9.66+-6.32	8.60+-0.84	8.16+-7.46 AllKNN $k=1$	17.60	▼
Golub64	5	11.08+-10.71	3.00+-0.00	6.97+-7.36 DROP1 $k=9$	84.73	▼
Colon16	5	19.77+-15.38	3.00+-0.00	18.11+-14.63 ENN $k=9$	26.52	□

Na avaliação do classificador C4.5, os resultados encontrados na Tabela 6.2, se comparados aos existentes na Tabela 5.4, permitem diferentes conclusões. De modo geral, todas as combinações, com exceção da desenvolvida para o conjunto *ExpGen*, obtiveram resultados superiores àqueles encontrados para os dados originais.

Nesse sentido, para o conjunto *Leukemia*, a combinação resultou no melhor desempenho obtido. Para o conjunto *Colon16*, a combinação apresentou resultado semelhante, estatisticamente igual, aos melhores até então encontrados para esse conjunto. Para o conjunto *Golub*, a combinação desenvolvida apresentou desempenho semelhante ao melhor resultado, apresentado na coluna "Melhor Desempenho". Para os demais conjuntos investigados, as combinações produziram resultados superiores aos encontrados para os dados originais, porém inferiores aos melhores resultados obtidos por algumas das técnicas de detecção de ruído aplicadas individualmente. Assim como ocorreu com as combina-

ções desenvolvidas para o classificador SVM, as complexidades apresentadas por todos os conjuntos avaliados foram inferiores às aquelas apresentadas para os dados originais.

A análise da complexidade para o classificador C4.5 se dá pela observação do tamanho médio das árvores produzidas. Reduções nas árvores refletem em ganhos de compreensibilidade dos modelos obtidos, uma vez que árvores menores podem ser consideradas mais fáceis de serem examinadas.

Tabela 6.3: Resultados obtidos pelas combinações desenvolvidas para o classificador RIPPER.

Conjunto de Dados	Número de técnicas	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Lung	7	8.05+-5.34	4.20+-0.42	6.05+-3.93 RENN $k=1$	14.94	▼
Golub	6	12.32+-9.97	2.20+-0.42	12.50+-12.05 AllKNN, ENN, RENN $k=1$	15.74	▲
ExpGen	3	13.45+-7.01	5.20+-0.63	11.52+-6.74 ENN $k=1$	17.60	Δ
Colon	4	26.67+-18.23	2.30+-0.48	19.52+-21.67 DEL $k=3$	29.03	▼
Ovarian	6	1.97+-2.07	3.00+-0.00	1.97+-2.80 AllKNN $k=3$	14.53	Δ
Golub64	6	15.35+-10.11	2.00+-0.00	6.79+-9.84 AllKNN $k=3$	11.73	▼
Colon16	4	19.76+-18.98	2.10+-0.31	14.76+-19.95 RENN $k=1$	26.16	▼

A análise dos resultados para o classificador RIPPER, presentes na Tabela 6.3, comparados aos encontrados na Tabela 5.5, indica que apenas as combinações desenvolvidas para os conjuntos *Colon* e *Golub64* produziram resultados inferiores aos obtidos para os dados originais, ao passo que a combinação desenvolvida para o conjunto *ExpGen* apresentou resultados estatisticamente semelhantes ao melhor resultado apresentado e aos resultados obtidos para os dados originais.

A combinação desenvolvida para o conjunto *Lung* obteve resultado superior ao encontrado para os dados originais, porém inferior a algumas das técnicas de detecção de ruído aplicadas individualmente. O conjunto *Colon16* apenas apresentou resultado inferior ao obtido pela técnica RENN com $k=1$ aplicada individualmente, sendo que seus resultados foram semelhantes aos obtidos para os dados originais.

As combinações desenvolvidas para os conjuntos *Golub* e *Ovarian* foram aquelas com melhor desempenho para o algoritmo RIPPER, uma vez que obtiveram resultados superiores a todos os testes anteriormente realizados. Com relação às complexidades obtidas, para todos os conjuntos de dados, essas foram inferiores às apresentadas para os dados originais.

O número de regras produzidas durante o treinamento dos dados representa a complexidade do modelo induzido pelo classificador RIPPER. Um número menor de regras indica menor complexidade do modelo obtido.

Novas combinações foram implementadas com o objetivo de desenvolver combinações que fossem ainda mais eficientes. Nessas, foi utilizado apenas um conjunto das técnicas empregadas no desenvolvimento das combinações anteriormente apresentadas. Esse conjunto consiste apenas das técnicas de detecção de ruído que apresentaram, de acordo com o teste estatístico realizado, desempenho relativamente superior ao obtido para os dados originais, sendo assim, não foram consideradas as técnicas que apresentaram resultados semelhantes (levemente superiores).

Nesse sentido, a Tabela 6.4 apresenta as novas combinações desenvolvidas para o classificador SVM. É possível notar que apenas duas combinações puderam ser desenvolvidas de acordo com o critério adotado, listadas abaixo:

- *Golub*, constituída pelas técnicas DROPs 3 e 4 com $k=3$, e DROP5 com $k=9$;
- *ExpGen*, composta pelas técnicas DROP5 com $k=3$ e DEL com $k=1$ e $k=3$.

Tabela 6.4: Resultados obtidos pelas novas combinações desenvolvidas para o classificador SVM.

Conjunto de Dados	Número de técnicas	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Golub	3	30.71+-26.09	10.60+-2.17	21.07+-13.72 DROP4 $k=3$	83.64	▼
ExpGen	3	6.71+-6.45	11.04+-1.48	5.74+-4.37 DEL $k=3$	92.64	▼

Na Tabela 6.5 são encontradas as novas combinações desenvolvidas para o classificador C4.5. Para esse classificador, apenas cinco novas combinações foram criadas, a saber:

- *Leukemia*, composta pelas técnicas ENN com $k=1$ e $k=3$, e RENN com $k=1$, $k=3$ e $k=9$;
- *Lung*, com as técnicas AllKNN, ENN e RENN com $k=1$;
- *Golub*, constituída pelas técnicas ENN, DROPs 2 e 5 com $k=9$;
- *Golub64*, composta pelas técnicas DROPs 1, 2 e DEL com $k=9$;
- *Colon16*, formada pelas técnicas ENN com $k=1$, $k=3$ e $k=9$.

Tabela 6.5: Resultados obtidos pelas novas combinações desenvolvidas para o classificador C4.5.

Conjunto de Dados	Número de técnicas	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Leukemia	5	16.20+-6.66	21.00+-1.63	16.81+-4.98 RENN $k=9$	17.29	▲
Lung	3	5.06+-4.08	7.00+-0.00	5.56+-4.97 ENN $k=1$	14.94	▲
Golub	3	13.58+-12.26	3.00+-0.00	11.44+-11.28 DROP5 $k=9$	53.24	□
Golub64	3	8.22+-11.43	3.00+-0.00	6.97+-7.36 DROP1 $k=9$	84.41	▼
Colon16	3	18.10+-16.61	3.00+-0.00	18.11+-14.63 ENN $k=9$	23.47	□

Na Tabela 6.6 são apresentadas as novas combinações criadas para o classificador RIPPER, sendo apenas seis as novas combinações desenvolvidas, descritas abaixo:

- *Lung*, constituída pelas técnicas AllKNN, ENN e RENN com $k=1$;
- *Golub*, formada também pelas técnicas AllKNN, ENN e RENN com $k=1$;
- *Colon*, composta pelas técnicas AllKNN com $k=1$, ENN e DEL com $k=3$;
- *Ovarian*, constituída pelas técnicas AllKNN e RENN com $k=1$ e $k=3$ e ENN com $k=1$;
- *Golub64*, formada pelas técnicas AllKNN com $k=1$, $k=3$ e $k=9$;
- *Colon16*, composta pelas técnicas AllKNN com $k=1$ e $k=3$ e RENN com $k=1$.

Tabela 6.6: Resultados obtidos pelas novas combinações desenvolvidas para o classificador RIPPER.

Conjunto de Dados	Número de técnicas	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Lung	3	8.05+-5.34	4.20+-0.42	6.05+-3.93 RENN $k=1$	14.94	▼
Golub	3	12.32+-9.97	2.20+-0.42	12.50+-12.05 AllKNN, ENN, RENN $k=1$	15.43	▲
Colon	3	26.90+-17.71	2.40+-0.51	19.52+-21.67 DEL $k=3$	32.61	▼
Ovarian	5	1.97+-2.07	3.00+-0.00	1.97+-2.80 AllKNN $k=3$	14.53	Δ
Golub64	3	11.07+-8.94	2.10+-0.31	6.79+-9.84 AllKNN $k=3$	12.81	▼
Colon16	3	14.76+-16.56	2.10+-0.31	14.76+-19.95 RENN $k=1$	28.31	Δ

Para as novas combinações desenvolvidas é possível observar que a maioria delas, com exceção da combinação desenvolvida para o conjunto de dados *ExpGen* e classificador

SVM, e da desenvolvida para o conjunto de dados *Leukemia* e classificador C4.5, apresentaram desempenho superior ao obtido pelas primeiras combinações implementadas. Mesmo para as duas combinações citadas, o desempenho foi apenas levemente inferior.

Com relação ao desempenho final obtido pelas novas combinações, mesmo com uma melhora no desempenho, a combinação desenvolvida para o classificador SVM e conjunto de dados *Golub* obteve desempenho superior ao obtido para os dados originais, porém inferior ao apresentado na coluna "Melhor Desempenho".

Para o classificador C4.5, todas as novas combinações apresentaram desempenho superior ou semelhante aos obtidos para os dados originais, com a vantagem de apresentarem, com exceção da desenvolvida para o conjunto de dados *Leukemia*, desempenho superior às primeiras combinações desenvolvidas.

Nesse sentido, a nova combinação desenvolvida para o conjunto de dados *Lung* obteve desempenho superior a quaisquer dos resultados anteriores obtidos, ao passo que as desenvolvidas para os conjuntos de dados *Golub* e *Colon16* apresentaram desempenho estatisticamente igual aos encontrados na coluna "Melhor Desempenho". Para o conjunto de dados *Golub64*, o desempenho da nova combinação foi superior ao obtido pela primeira desenvolvida, porém ainda inferior ao desempenho apresentado na coluna "Melhor Desempenho".

Na análise do classificador RIPPER, tem-se que todas as novas combinações implementadas apresentaram resultados iguais às primeiras desenvolvidas, exceto para os conjuntos *Colon*, *Golub64* e *Colon16*, em que as novas combinações foram superiores às inicialmente analisadas. Nesse sentido, para as novas combinações mencionadas, apenas a desenvolvida para o conjunto de dados *Colon* apresentou desempenho inferior ao encontrado para os dados originais, porém superior ao encontrado pela primeira combinação desenvolvida, ao passo que a desenvolvida para o conjunto de dados *Golub* obteve resultados superiores a todos os testes anteriormente realizados, sendo iguais aos obtidos pela primeira combinação desenvolvida para esse mesmo conjunto de dados.

Para os conjuntos de dados *Ovarian* e *Colon16*, o desempenho obtido foi semelhante àquele encontrado na coluna "Melhor Desempenho". Os conjuntos de dados *Lung* e *Golub64* apresentaram resultados inferiores aos encontrados na coluna "Melhor Desempenho", sendo que, para os dados originais, apresentaram, respectivamente, desempenho superior e semelhante.

A análise da complexidade das novas combinações desenvolvidas para os três classificadores avaliados permite notar que esta se mostrou semelhante às obtidas pelas primeiras

combinações apresentadas.

A partir da observação dos resultados apresentados é possível notar que as combinações que se valem de técnicas de detecção de ruído pertencentes aos dois grupos anteriormente citados, adquirem as características do grupo que possui maior quantidade de técnicas na combinação, seja na observação dos resultados, complexidade ou quantidade de exemplos eliminados do conjunto original.

Uma vez que as combinações desenvolvidas não alcançaram, em sua maioria, resultados superiores ao melhor desempenho obtido pela aplicação individual de uma determinada técnica de detecção de ruído, foram desenvolvidas combinações pela metodologia de consenso. Nessa metodologia, cada dado é considerado ruído se, para n técnicas avaliadas, todas elas classificarem o dado como ruidoso. O objetivo foi avaliar se uma metodologia mais conservadora, em que uma menor quantidade de dados é considerada ruído, é capaz de melhorar o desempenho da combinação.

Sendo assim, foram selecionadas algumas das combinações pertencentes ao segundo grupo desenvolvido (uma vez que essas apresentaram desempenho superior àquelas inicialmente implementadas), para comporem as novas combinações, agora segundo a metodologia por consenso. Nesse sentido, as combinações selecionadas foram:

- *Golub*, para o classificador SVM, pois apresenta desempenho inferior ao melhor resultado até então encontrado;
- *Golub*, para os classificadores C4.5 e RIPPER, pois os resultados para os dois conjuntos de combinações desenvolvidas foram praticamente iguais;
- *Colon*, para o classificador RIPPER, por apresentar desempenho inferior ao obtido para os dados originais;
- *Colon16*, para os classificadores C4.5 e RIPPER, já que tiveram seu desempenho semelhante ao melhor resultado até agora obtido;
- *Golub64*, para os classificadores C4.5 e RIPPER, pois tiveram seus desempenhos aperfeiçoados, porém ainda inferiores ao melhor resultado obtido para esse conjunto.

Os resultados obtidos para as novas combinações, agora segundo a metodologia por consenso, são apresentados na Tabela 6.7. A complexidade das novas combinações são superiores ou semelhantes se comparadas às complexidades das mesmas combinações segundo a metodologia de votação por maioria. Esse fato ocorre porque uma menor quantidade de dados é eliminada nas combinações por consenso.

Tabela 6.7: Resultados obtidos pelas combinações por consenso desenvolvidas.

Conjunto de Dados	Classificador Avaliado	Taxas de Erro	Complexidade	Melhor Desempenho	Redução (%)	Desempenho Estatístico
Golub	SVM	25.36+-14.72	30.00+-3.09	21.07+-13.72 DROP4 $k=3$	53.70	▼
	C4.5	15.37+-10.11	4.00+-1.05	11.44+-11.28 DROP5 $k=9$	13.27	▼
	RIPPER	12.32+-9.97	2.20+-0.42	12.50+-12.05 AllKNN, ENN, RENN $k=1$	15.41	▲
Colon	RIPPER	24.05+-15.24	2.10+-0.31	19.52+-21.67 DEL $k=3$	15.23	▼
Golub64	C4.5	17.15+-19.97	3.00+-0.00	6.97+-7.36 DROP1 $k=9$	68.05	▼
	RIPPER	15.41+-10.02	2.10+-0.31	6.79+-9.84 AllKNN $k=3$	11.88	▼
Colon16	C4.5	18.10+-16.61	3.80+-1.03	18.11+-14.63 ENN $k=9$	19.89	△
	RIPPER	21.43+-17.74	2.10+-0.31	14.76+-19.95 RENN $k=1$	25.27	▼

Na análise dos resultados, tem-se que os conjuntos de dados *Colon16*, para o classificador C4.5, e *Golub*, para o classificador RIPPER, apresentaram resultados idênticos aos obtidos pelas combinações de votação por maioria, sendo assim, para esses conjuntos, a melhoria no desempenho, imposta pela combinação de votação por maioria, não pôde ser aumentada, mesmo diminuindo a quantidade de dados eliminados. Isso pode ser explicado pela pequena quantidade de dados que deixou de ser eliminada, fato que não implicou em alterações no desempenho final.

Já os conjuntos de dados *Golub*, para o classificador SVM, e *Colon*, para o classificador RIPPER, apresentaram, respectivamente, resultados semelhantes e superiores às combinações anteriormente desenvolvidas. Nesse sentido, ambos foram inferiores ao resultado apresentado na coluna "Melhor Desempenho", porém apresentaram, respectivamente, desempenho superior e semelhante ao obtido para os dados originais. Para os demais conjuntos avaliados, o desempenho foi deteriorado com a metodologia de consenso, ou seja, com a manutenção de uma maior quantidade de dados no conjunto de treinamento, as taxas de erro do classificador aumentaram.

De modo a facilitar a análise dos resultados apresentados, a Tabela 6.8 apresenta um resumo dos resultados obtidos pelas combinações desenvolvidas. Nessa tabela, apenas são apresentadas as melhores combinações obtidas dentre todas as desenvolvidas, a saber combinações de votação por maioria primeiro e segundo grupos, e combinações por consenso. Não são apresentadas também as combinações que não obtiveram resultados superiores aos obtidos pelas técnicas de detecção de ruído aplicadas individualmente ou aos encontrados para os dados originais, uma vez que essas combinações apenas deterioraram o

desempenho final encontrado.

Resultados inferiores são representados por ▼; △ indica resultados semelhantes, levemente superiores ou inferiores; resultados superiores são representados por ▲; ao passo que □ indica resultados iguais.

Tabela 6.8: Resumo dos resultados obtidos pelas combinações desenvolvidas.

Conjunto de Dados	Classificador Avaliado	Melhor Combinação	Dados Originais	Melhor Desempenho
Golub	SVM	Maioria(2)	▲	▼
		Consenso	▲	▼
	RIPPER	Maioria(2)	▲	□
		Maioria(1,2)	▲	▲
Colon	RIPPER	Consenso	△	▼
ExpGen	SVM	Maioria(1)	△	▼
	RIPPER	Maioria(1)	△	△
Leukemia	SVM	Maioria(1)	△	△
	C4.5	Maioria(1)	▲	▲
Lung	C4.5	Maioria(2)	▲	▲
	RIPPER	Maioria(1,2)	▲	▼
Ovarian	RIPPER	Maioria(1,2)	▲	△
Golub64	C4.5	Maioria(2)	▲	▼
	RIPPER	Maioria(2)	△	▼
Colon16	C4.5	Maioria(1,2)	□	□
		Consenso	□	□
	RIPPER	Maioria(2)	▲	△

Para ilustrar as diferenças existentes entre os dados removidos pelas combinações desenvolvidas pela metodologia de consenso e de votação por maioria, as figuras 6.1 e 6.2 apresentam gráficos indicando os exemplos removidos pelas duas metodologias citadas, respectivamente, para os conjuntos *Colon* e classificador RIPPER e *Golub64* e classificador C4.5.

Figura 6.1: Exemplos removidos pelas combinações desenvolvidas pela metodologia de consenso e de votação por maioria, para o conjunto *Colon*.

Figura 6.2: Exemplos removidos pelas combinações desenvolvidas pela metodologia de consenso e de votação por maioria, para o conjunto *Golub64*.

De acordo com os resultados obtidos, é possível observar que as combinações por consenso deterioraram os resultados para a maioria dos conjuntos de dados analisados.

A metodologia de votação por maioria pode ter propiciado uma melhor combinação das características existentes entre as técnicas de detecção de ruído pertencentes ao primeiro ou segundo grupo, fato prejudicado quando se utiliza a metodologia de consenso. Nesse sentido, para as combinações por consenso em que a quantidade de dados eliminados foi semelhante àquela das combinações de votação por maioria, os resultados obtidos foram iguais ou semelhantes aos anteriormente obtidos nas combinações de votação por maioria desenvolvidas. Para os conjuntos em que a votação por consenso manteve uma quantidade significativamente maior de dados nos conjuntos de treinamento, o desempenho final foi deteriorado.

Como exceção, tem-se a combinação desenvolvida para o conjunto de dados *Colon* e o classificador RIPPER, que apresentou desempenho superior. Para esse conjunto, a eliminação de uma menor quantidade de dados resultou em melhores resultados, ou seja, para ser ruído, um exemplo precisa ser considerado como tal por todas as técnicas avaliadas.

Esses resultados sugerem que, na eliminação por consenso, é reduzido o número de exemplos eliminado pela melhor técnica de detecção de ruído no conjunto final de exemplos eliminados. Quando é utilizada a combinação de votação por maioria, mais exemplos eliminados pela melhor técnica tendem a ser eliminados no conjunto final. Assim, exemplos indicados como ruídos pela melhor técnica de detecção, cuja eliminação teve grande influência na melhoria da taxa de classificação correta, podem não ter sido eliminados na

combinação por consenso em maior número que na combinação de votação por maioria. Uma alternativa para reduzir esse problema seria ponderar os exemplos de acordo com o desempenho obtido pelo classificador quando ele foi eliminado. Para o conjunto de dados *Colon* e o classificador RIPPER, é possível que a combinação por consenso tenha deixado de descartar exemplos que provavelmente não eram ruídos e que foram descartados pela combinação de votação por maioria.

Nesse âmbito, as combinações desenvolvidas obtiveram, de uma maneira geral, resultados superiores àqueles obtidos para os dados originais, mantendo ou diminuindo a complexidade do classificador de AM envolvido. Entretanto, em apenas alguns casos essas foram capazes de apresentar desempenho superior ou semelhante ao obtido por algumas das técnicas de detecção de ruído aplicadas individualmente. Sendo assim, as combinações apresentaram desempenho satisfatório, porém não foi possível a obtenção de melhoria significativa no desempenho para todos os conjuntos de dados e classificadores analisados.

6.3 Considerações Finais

Neste capítulo foram apresentados os experimentos realizados com as diferentes combinações de técnicas de detecção de ruído desenvolvidas.

A etapa de combinação apresentou, em sua maioria, desempenho superior à aplicação de cada técnica de detecção de ruído individualmente sobre os dados originais, mesmo que, em alguns casos, tenha deteriorado o desempenho ou obtido resultados inferiores aos encontrados anteriormente. Para algumas das combinações desenvolvidas, o desempenho equiparou-se ou foi superior aos melhores resultados até então encontrados.

A metodologia de combinação de votação por maioria mostrou-se mais adequada, uma vez que apresentou resultados melhores que aqueles apresentados pela metodologia de consenso. Isso pode ser explicado com a hipótese de a combinação por consenso ter deixado de eliminar exemplos que poderiam, uma vez eliminados, melhorar o desempenho obtido pelo classificador. E que esses exemplos foram eliminados pela combinação de votação por maioria.

Uma grande vantagem do uso da combinação é que, ao custo de possivelmente não obter o melhor desempenho possível com a detecção de ruídos, melhorar ou manter as taxas de classificação corretas obtidas com os dados originais. Outra vantagem é reduzir a necessidade de descobrir qual a técnica de detecção mais indicada para um novo conjunto de dados.

Nesse sentido, as combinações desenvolvidas demonstraram ser capazes de tratar, com obtenção de desempenho satisfatório, problemas de expressão gênica representados por conjuntos de dados complexos e de elevada dimensionalidade sendo, portanto, adequadas para utilização em demais problemas relacionados, não abordados neste trabalho.

No entanto, diferentes modelos de combinações se adequaram aos diferentes conjuntos de dados analisados, sendo necessário algum esforço para determinação de quais as melhores técnicas a serem combinadas na análise de um novo conjunto sobre dados de expressão gênica.

Conclusões

Este trabalho investigou a detecção de ruídos em conjuntos de dados e sua influência no desempenho de classificadores de Aprendizado de Máquina (AM). Na literatura não pode ser encontrada uma abordagem única ou um determinado algoritmo que seja aplicável a todos os problemas de detecção de ruídos sem perda de desempenho. Pelo contrário, diferentes e diversas são as metodologias e abordagens para a detecção de ruídos que podem ser encontradas.

De acordo com HODGE e AUSTIN [2004], abordagens utilizadas para detecção de ruídos se enquadram em três categorias: identificação de ruídos sem o conhecimento prévio dos dados, detecção de anomalias em sistemas ou conjuntos de dados e detecção de novidades.

O que deve ser considerado é a escolha de qual metodologia e algoritmos utilizar, de acordo com as características do problema e dos dados a serem analisados. Deve-se optar por algoritmos ou conjuntos de algoritmos que sejam capazes de modelar o problema estudado e de identificar possíveis ruídos de modo preciso. Os algoritmos selecionados devem também ser escaláveis aos conjuntos de dados que serão avaliados.

Sendo assim, neste trabalho foram utilizados dados de problemas de bioinformática, mais especificamente dados provenientes de análises de expressão gênica, cujos conjuntos de dados apresentam elevada dimensionalidade e grande quantidade de ruído. Conjuntos

de dados de elevada dimensionalidade possuem maior probabilidade de apresentar problemas. O principal deles diz respeito à dificuldade de manipulação e entendimento dos dados, porém outros mais advêm do modo com que os dados são coletados e armazenados nesses conjuntos. Assim, é comum que eles contenham erros, como dados incompletos, corrompidos ou distorcidos, erros de medições, falhas humanas ou dos equipamentos utilizados, dentre diversos fatores, que dificultam ainda mais a indução de hipóteses por algoritmos de AM.

Para a detecção de ruídos, foram avaliados algoritmos baseados em distância, que analisam medidas de similaridade existentes entre os dados presentes em um conjunto de dados.

De uma maneira geral, para as diferentes técnicas de detecção de ruído baseadas em distância, os resultados encontrados apresentaram uma grande variação, dependente da técnica avaliada, do conjunto de dados e do algoritmo de AM investigado. Nesse sentido, em diversos experimentos foram encontrados resultados satisfatórios, superiores aos encontrados para os dados originais.

Os resultados permitiram identificar a existência de dois grupos distintos de técnicas. O primeiro, constituído pelas técnicas AllKNN, ENN e RENN, que elimina uma menor quantidade de dados; e o segundo, formado pelas técnicas DROPs 1 a 5 e DEL, que elimina uma quantidade significativamente superior de dados.

Combinações de técnicas de detecção de ruído foram desenvolvidas valendo-se, para isso, dos melhores resultados obtidos por essas técnicas individualmente e objetivando complementá-las de modo a melhorar o desempenho final.

As técnicas de detecção de ruído foram combinadas pela metodologia de votação por maioria, em que cada dado é considerado ruído se, para n técnicas avaliadas, ao menos m delas, em que m é maior que $n/2$, classificaram o dado como ruidoso. Assim como pela metodologia de consenso, em que cada dado é considerado ruído e, portanto, eliminado, se, para n técnicas avaliadas, todas elas classificarem o dado como ruidoso.

Para a metodologia de votação por maioria, dois grupos de combinações foram desenvolvidos. O primeiro, em que as combinações foram constituídas por todas as técnicas de detecção de ruído que apresentaram desempenho superior ou semelhante (levemente superior) aos obtidos para os dados originais. O segundo, com combinações compostas apenas pelas técnicas de detecção de ruído que apresentaram, de acordo com o teste estatístico, desempenho relativamente superior ao obtido para os dados originais. Para a votação por consenso, foram utilizadas apenas as técnicas que apresentaram desempenho

relativamente superior, de acordo com o teste estatístico.

O resultados indicaram que as combinações de votação por maioria pertencentes ao segundo grupo apresentaram desempenho superior ao obtido pelas primeiras implementadas e superior aos dados originais, com exceção daquela desenvolvida para o conjunto de dados *Colon* e o classificador RIPPER, cujo resultado foi inferior ao obtido para os dados originais. Entretanto, a combinação citada foi a única das combinações por consenso desenvolvidas que apresentou melhores resultados que as demais implementadas. Todas as demais combinações por consenso desenvolvidas deterioraram os resultados das combinações de votação por maioria criadas.

É importante observar que a utilização de uma etapa de combinação apresentou, em sua maioria, desempenho superior à aplicação de cada técnica de detecção de ruído individualmente sobre os dados originais. Para algumas das combinações desenvolvidas, o desempenho equiparou-se ou foi superior aos melhores resultados anteriormente obtidos pela utilização apenas das técnicas individuais de detecção de ruídos.

A metodologia de combinação de votação por maioria mostrou-se mais adequada, pois apresentou resultados melhores que aqueles encontrados para a metodologia de consenso. Isso permite concluir que essa metodologia propicia uma combinação, embora discreta, das características existentes entre as técnicas de detecção de ruído pertencentes aos dois grupos citados, permitindo que esses se complementem, fato prejudicado quando se utiliza a metodologia de consenso.

Para trabalhos futuros, a investigação de diferentes algoritmos de AM, assim como técnicas de detecção de ruído pertencentes a diferentes grupos, tais como técnicas de agrupamento de dados, por exemplo, mostra-se como uma possibilidade interessante para avaliar e comparar os novos resultados encontrados com os apresentados neste trabalho. Novas combinações também podem ser implementadas, combinando técnicas de detecção de ruído pertencentes a grupos distintos e avaliando como esses se completam na busca por resultados cada vez mais satisfatórios, com taxas de erro reduzidas. Uma outra possibilidade é ponderar a influência das técnicas individuais na combinação, de forma a levar em conta o quanto cada uma delas melhora o desempenho do classificador empregado.

Referências Bibliográficas

- Aggarwal, C. C.; Hinneburg, A. e Keim, D. A. (2001) On the surprising behavior of distance metrics in high dimensional space. Em Bussche, J. V. e Vianu, V. (eds.): Proceedings of 8th International Conference on Database Theory (ICDT 2001), Lecture Notes in Computer Science, v. 1973, pp. 420-434. Springer-Verlag.
- Aggarwal, C. e Yu, P. (2001) Outlier detection for high dimensional data. Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 37-47.
- Aha, D. W.; Dennis, K. e Marc, K. A. (1991) Instance-Based Learning Algorithms. Machine Learning, v. 6, pp. 37-66.
- Aha, D. W. (1992) Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. International Journal of Man-Machine Studies, v. 36, pp. 267-287.
- Alon, U.; Barkai, N.; Notterman, D. A.; Gish, K.; Ybarra, S.; Mack, D. e Levine, A. J. (1999) Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. Proceedings of National Academy of Sciences of the United States of American, v. 96, pp. 6745-6750.
- Baranauskas, J. A. e Monard, M. C. (2000) Reviewing some machine learning concepts and methods. Relatório Técnico 102, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, campus São Carlos. Disponível em ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_102.ps.zip.
- Baranauskas, J. A. (2001) Extração de conhecimento por múltiplos indutores. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, campus São Carlos.

- Barnett, V. e Lewis, T. (1994) *Outliers in Statistical Data*. 3rd Edition. John Wiley and Sons.
- Batista, G. E. A. P. A.; Carvalho, A. C. P. L. F. e Monard, M. C. (2000) Applying one-sided selection to unbalanced datasets. *Mexican International Conference on Artificial Intelligence (MICAI), Lecture Notes in Artificial Intelligence*, v. 1793, pp. 315-325. Springer-Verlag.
- Beyer, K.; Goldstein, J.; Ramakrishnan, R. e Shaft, U. (1999) When is 'nearest neighbor' meaningful? Em Beerli, C. e Buneman, P. (eds.): *7th International Conference on Database Theory (ICDT-99), Lecture Notes in Computer Science*, v. 1540, pp. 217-235. Springer-Verlag.
- Blake, C. L. e Merz, C. J. (1998) UCI repository of machine learning databases. Disponível em <http://www.ics.uci.edu/mllearn/MLRepository.html>
- Blockeel, H. e Raedt, L. (1998) Top-down induction of first order logical decision trees. *Artificial Intelligence*, v. 101, n. 1-2, pp. 285-297.
- Brodley, C. E. e Friedl, M. A. (1996) Identifying and Eliminating Mislabeled Training Instances. *Proceedings of the 13th National Conference on Artificial Intelligence AAAI/IAAI*, v. 1, pp. 799-805, AAAI Press.
- Brodley, C. E. e Friedl, M. A. (1999) Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, v. 11, pp. 131-167.
- Brown, M.; Grundy, W.; Lin, D.; Christianini, N.; Sugnet, C. Jr M. e Haussler, D. (1999) Support vector machine classification of microarray gene expression data. Technical Report UCSC-CRL 99-09, Departamento de Computação, Universidade California Santa Cruz.
- Cohen, W. W. (1995) Fast effective rule induction. Em Elrington, A.; e Russell, S., (eds.): *Proceedings of the 12th International Conference on Machine Learning*, pp. 115-123, Morgan Kaufmann.
- Collobert, R. e Bengio, S. (2001) SVM-Torch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, v. 1, pp. 143-160.
- Corney, D. P. A. (2002) *Intelligent Analysis of Small Data Sets for Food Design*. Tese PhD., Departamento de Computação, London University College.

- Cristianini, N. e Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.
- Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, v. 7, pp. 1-30.
- Dudoit, S.; Fridlyand, J.; e Speed T. P. (2000) Comparison of discrimination methods for the classification of tumors using gene expression data. Technical Report 576, Department of Statistics, UC Berkeley.
- Dunn, O. J. (1961) Multiple comparisons among means. *Journal of the American Statistical Association*, v. 56, pp. 52-64.
- Efron, B. e Tibshirani, R. (1993) *An Introduction to the Bootstrap*. Chapman and Hall.
- Emanuel, F P.III; Ali, M. A.; Ben, A. H.; Peter, J. L.; Vincent, A. F.; Seth, M. S., Gordon, B. M., Charles, S.; David, A. F.; Elise, C. K.; e Lance, A. L. (2002) Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, v. 359, pp. 572-577. <http://www.thelancet.com>
- Frank, E. e Witten, I. H. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann. <http://www.cs.waikato.ac.nz/ml/weka/> - acessado em agosto de 2007.
- Fürnkranz, J. e Widmer, G. (1994) Incremental reduced error pruning. Em Cohen, W. e Hirsh, H., (eds.): *Proceedings of the 11th International Conference on Machine Learning*, pp. 70-77, Morgan Kaufmann.
- Feres, B. S. (2005) *Seleção de Características em SVMs Aplicadas a Dados de Expressão Gênica*. Dissertação de Mestrado. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, campus São Carlos.
- Freund, Y. e Schapire, R. E. (1996) Experiments with a new boosting algorithm. Em Saitta, L. (ed.): *Proceedings of the 13th International Conference on Machine Learning*, pp. 148-156. Morgan Kaufmann.
- Fisher, R. A. (1959) *Statistical methods and scientific inference*. Hafner Publishing Co., 2nd edition.
- Friedman, M. (1940) A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, v. 11, n. 1, pp. 86-92.

- Friedman, M. (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, v. 32, n. 200, pp. 675-701.
- Gamberger, D. e Lavrac, N. (1997) Conditions for Occam's razor applicability and noise elimination. *Proceedings of the European Conference on Machine Learning*, pp. 108-123.
- Gamberger, D., Boskovic, R. e Lavrac, N. (1999) Experiments with noise filtering in a medical domain. *Proceedings of the 16th International Conference on Machine Learning*, pp. 143-151. Morgan Kaufmann.
- Gamberger, D., Dzeroski, S. e Lavrac, N. (2000) Noise detection and elimination in data preprocessing: experiments in medical domains. *Applied Artificial Intelligence*, v. 14, pp. 205-223.
- Gates, G. W. (1972) The Reduced Nearest Neighbor Rule. *IEEE Transactions on Information Theory*, v. 18, n. 5, pp. 431-433.
- Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing.
- Golub T. R.; Tamayo, P.; Slonim, D.; Mesirov, J.; Zhu, Q.; Kitareewan, S.; Dmitrovsky, E. e Lander, E. S. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of National Academy of Sciences*, v. 96, pp. 2907-2912.
- Haykin, S. (1999) *Neural Networks - A Comprehensive Foundation*. Prentice-Hall, New Jersey, 2nd edition.
- Hayter, A. J. e Tsui, K. L. (1994) Identification and quantification in multivariate quality control problems. *Journal of Quality Technology*, v. 26, n. 3, pp. 197-208.
- Hawkins, S., He, H., Williams, G. e Baxter, R. (2002) Outlier detection using replicator neural networks. *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pp. 170-180.
- He, Z. e Xu, S. (2002) Outlier detection integrating semantic knowledge. *Proceedings of the 3th International Conference on Web-Age Information Management*, pp. 126-131.

- He, Z., Xu, X. e Deng, S. (2003) Discovering cluster-based local outliers. *Pattern Recognition Letters*, v. 24, n. 9-10, pp. 1641-1650, Elsevier Science Inc.
- Hinneburg, A.; Aggarwal, C. C. e Keim, D. A. (2000) What is the nearest neighbor in high dimensional spaces? Em Abbadi, A. E.; Brodie, M. L.; Chakrovarthy, S.; Dayal, U.; Kamel, N.; Schlageter, G. e Whang, K. Y. (eds.): *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB 2000)*, pp. 506-515.
- Hochberg Y. (1988) A sharper Bonferonni procedure for multiple tests of significance. *Biometrika*, v. 75, pp. 800-803.
- Hodge, V. e Austin, J. (2004) A survey of outlier detection methodologies. *Artificial Intelligence Review*, v. 22, pp. 85-126. Kluwer Academic Publishers.
- Jankowski, N. e Grochowski, M. (2004) Comparison of instances selection algorithms: I. Algorithms survey. *Proceedings of ICAISC, Lecture Notes in Artificial Intelligence* 3070, pp. 598-603. Springer-Verlag.
- Jin, W.; Tung, A. K. H. e Han, J. (2001) Mining top-n local outliers in large databases. *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 293-298.
- Joachims, T. (2002) *Learning to Classify Text using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers.
- John, G. H. (1995) Robust Decision Trees: Removing outliers from databases. *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, pp. 174-179. AAAI Press.
- Khoshgoftaar, T. M. e Rebours, P. (2004) Generating Multiple Noise Elimination Filters with the Ensemble-Partitioning Filter. *Proceedings of the IEEE International Conference on Information Reuse and Integration. IRI 2004*, pp. 369-375.
- Knorr, E. M. e Ng, R. T. (1998) Algorithms for mining distance-based outliers in large datasets. *Proceedings of the 24th Conference on Very Large Data Bases (VLDB)*, pp. 392-403.
- Knorr, E. M.; Ng, R. T. e Tucakov, V. (2000) Distance-based outliers: algorithms and applications. *The VLDB Journal: Very Large Data Bases*, v. 8, n. 3-4, pp. 237-253.

- Lavrac, N. e Gamberger, D. (2001) Saturation filtering for noise and outlier detection. Em Scheffer, T. e Wrobel, S. (eds.): Proceedings of the Workshop on Active Learning, Database Sampling, Experimental Design: Views on Instance Selection: Working Notes, pp. 1-4, 12th European Conference on Machine Learning / 5th European Conference on Principles and Practice of Knowledge Discovery in Databases.
- Lorena, A. C. e Carvalho, A. C. P. L. F. (2003) Introdução às Máquinas de Vetores Suporte (Support Vector Machines). Relatório Técnico 192, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, campus São Carlos.
- Lorena, A. C. e Carvalho, A. C. P. L. F. (2004) Evaluation of noise reduction techniques in the splice junction recognition problem. *Genetics and Molecular Biology*, v. 27, n. 4, pp. 665-672.
- Lowe, D. G. (1995) Similarity Metric Learning for a Variable-Kernel Classifier. *Neural Computation.*, v. 7, n. 1, pp. 72-85.
- Mitchell, T. (1997) *Machine Learning*. McGraw Hill.
- Monard, M. C. e Baranauskas, J. A. (2003) Conceitos de aprendizado de máquina. Em Rezende, S. O. (ed.): *Sistemas Inteligentes - Fundamentos e Aplicações*, pp. 89-114, Editora Manole.
- Monti, S.; Tamayo, P.; Mesirov, J. e Golub, T. (2003) Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, v. 52, n. 1-2, pp. 91-118, Springer.
- Muller, K. R.; Mika, S.; Ratsch, G.; Tsuda, K. e Scholkopf, B. (2001) An introduction to kernel-based learning algorithm. *IEEE Transactions on Neural Networks*, v. 12, n. 2, pp. 181-202.
- Nilsson, N. J. (1965) *Learning Machines*. McGraw-Hill.
- Noble, W. S. (2004) *Kernel Methods in Computational Biology*. Chapter 3: Support vector machines applications in computational biology, pp. 71-92. MIT Press.
- Pedersen, A. G. e Nielsen, H. (1997) Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB-97)*, pp. 226-233.

- Prampero, P. S. e Carvalho, A. C. P. L. F. (1999) Reconhecimento de imagens de navios utilizando combinação de classificadores. Anais do IV Workshop de Teses e Dissertações, , pp. 170-177. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.
- Prampero, P. S. (1998). Combinação de classificadores para o reconhecimento de padrões. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, campus São Carlos.
- Quinlan, J. R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann.
- Ramaswamy, S.; Rastogi, R. e Shim, K. (2000) Efficient algorithms for mining outliers from large data sets. Proceedings of the ACM SIGMOD International Conference on Management of Data and Symposium on Principles of Database Systems, pp. 427-438.
- Rezende, S. O. (2003) Sistemas Inteligentes: Fundamentos e Aplicações. Editora Manole.
- Sigillito, V. G.; Wing, S. P.; Hutton, L. V. e Baker, K. B. (1989) Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, v. 10, pp. 262-266.
- Smola, A. J. e Schoolkopf, B. (2002) Learning with Kernels. MIT Press.
- Souto, M. C. P.; Lorena, A. C.; Delbem, A. C. B. e Carvalho, A. C. P. L. F. (2003) Técnicas de Aprendizado de Máquina para problemas de Biologia Molecular. Minicursos de Inteligência Artificial, Jornada de Atualização Científica em Inteligência Artificial, XXIII Congresso da Sociedade Brasileira de Computação, pp. 103-152.
- Stanfill, C. e Waltz, D. (1986) Toward memory-based reasoning. Communications of the ACM, v. 29, n. 12, pp. 1213-1228, ACM Press.
- Stolfo, S. J.; Prodromidis, A. L.; Tselepis, S.; Lee, W.; Fan, D. W. e Chan, P. K. (1997) JAM: Java agents for meta-learning over distributed databases. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 74-81.
- Takeuchi, K. e Collier, N. (2003) Bio-medical entity extraction using support vector machines. Proceedings of the ACL 2003 Workshop on Natural Language processing in Biomedicine, v. 13, pp. 57-64, Association for Computational Linguistics.

- Tang, J.; Chen, Z.; Fu, A. W. e Cheung, D. (2002) A robust outlier detection scheme in large data sets. 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining.
- Tomek, I. (1976a) An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, v. 6, n. 6, pp. 448-452.
- Tomek, I. (1976b) Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, v. 7, n. 11, pp. 769-772.
- Tumer, K. e Ghosh, J. (1996) Classifier combining: Analytical results and implications. *Proceedings of the National Conference on Artificial Intelligence - Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pp. 126-132, AAAI Press.
- Vapnik, V. N. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, 2nd edition.
- Verbaeten, S. (2002) Identifying mislabeled training examples in ILP classification problems. Em Wiering, M. (ed.): *Proceedings of 12th Belgian-Dutch Conference on Machine Learning*, pp. 1-8.
- Verbaeten, S. e Assche, A. V. (2003) Ensemble Methods for noise elimination in Classification problems. Em Windeatt, T. e Roli, (eds.): *Proceedings of 4th International Workshop on Multiple Classifier Systems, LNCS 2709*, pp. 317-325, Springer.
- Wettschereck, D. (1994) A study of distance-based machine learning algorithms. Tese PhD. Department of Computer Science. Oregon State University.
- Wilson, D. L. (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, v. 2, n. 3, pp. 408-421.
- Wilson, D. R. e Martinez, T. R. (1997a) Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research (JAIR)*, v. 6, n. 1, pp. 1-34.
- Wilson, D. R. e Martinez, T. R. (1997b) Instance Pruning Techniques. *Proceedings of the 14th International Conference on Machine Learning*, pp. 404-411, Morgan Kaufmann.
- Wilson, D. R. e Martinez, T. R. (2000) Reduction techniques for instance-based learning algorithms. *Machine Learning*, v. 38, n. 3, pp. 257-286.
- Wolberg, W. H.; Mangasarian, O. L. e Street W. N. (1995) Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, v. 43, n. 4, pp. 570-577.

- Yeoh, E. J.; Ross, M. E.; Shurtle, S. A.; Williams, W. K.; Patel, D.; Mahfouz, R.; Behm, F. G.; Raimondi, S. C.; Relling, M. V.; Patel, A.; Cheng, C.; Campana, D.; Wilkins, D.; Zhou, X.; Li, J.; Liu, H.; Pui, C. H.; Evans, W. E.; Naeve, C.; Wong, L. e Downing, J. R. (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, v. 1, n. 2, pp. 133-143, Elsevier.
- Zhang, J. (1992) Selecting Typical Instances in Instance-Based Learning. *Proceedings of the 9th International Conference on Machine Learning*, pp. 470-479, Morgan Kaufmann Publishers Inc.
- Zheng, S.; Liu, J. e Tian, J. W. (2004) A new efficient SVM-based edge detection method. *Pattern Recognition Letters*, v. 25, n. 10, pp. 1143-1154, Elsevier Science Inc.
- Zhu, X.; Wu, X. e Chen, Q. (2003) Eliminating class noise in large datasets. *Proceedings of the 20th International Conference on Machine Learning*, v. 20, pp. 920-927.
- Zhu, X. e Wu, X. (2004) Class noise vs. Attribute noise: A quantitative study of their impacts. *Artificial Intelligence Review*, v. 22, n. 3, pp. 177-210. Kluwer Academic Publishers.