Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Identificação de Ruídos em Dados de Bioinformática

Ana Carolina Lorena
André C. P. L. F. de Carvalho

N° 295

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Fevereiro/2007

Identificação de Ruídos em Dados de Bioinformática*

Ana Carolina Lorena
André C. P. L. F. de Carvalho

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {aclorena, andre}@icmc.usp.br

Resumo. Os dados em Bioinformática geralmente se caracterizam pela presença de ruídos, os quais podem ser provenientes de diversas fontes. Como exemplo, pode-se citar erros provenientes de contaminações nas amostras laboratoriais utilizadas na geração dos dados. Neste relatório técnico é apresentada a utilização de técnicas para detecção e eliminação de ruídos de bases de dados biológicas.

Palavras-Chave: Bioinformática, Aprendizado de Máquina, identificação de ruídos.

Fevereiro 2007

*Trabalho realizado com auxílio da FAPESP e CNPq.

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

Sumário

1 Introdução **1**

2 Conceitos de Biologia Molecular **2**

3 Aprendizado de Máquina **4**

3.1 k-Vizinhos Mais Próximos 6

3.2 Árvores de Decisão 8

3.3 Máquinas de Vetores de Suporte 9

4 Identificação de Ruídos **12**

5 Técnicas de Pré-processamento **13**

6 Experimentos e Resultados **18**

6.1 Reconhecimento de Sítios de *Splicing* 18

6.1.1 Avaliação de Várias Técnicas Baseadas em Distância 19

6.1.2 Avaliação da Heurística *Tomek links* 24

6.2 Classificação Funcional de Genes 28

7 Conclusão **32**

Referências Bibliográficas **35**

Lista de Figuras

1 Célula, cromossomo, DNA, gene e suas relações 2

2 O processo de expressão gênica 3

3 *Splicing* em molécula de mRNA 4

4 Ilustração do processamento realizado pelo algoritmo k-NN (Mitchell, 1997) . . 7

5 AD simplificada para o diagnóstico de pacientes (Baranauskas & Monard, 2000) 8

6 (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características (Müller et al., 2001) 11

7 (a) Exemplo de conjunto de treinamento; (b) dados com rótulo incorreto; (c) dados redundantes; (d) *outliers*; (e) dados de borda; (f) dados seguros 13

8 Aplicando *Tomek links* a um conjunto de dados; (a) Conjunto de dados original; (b) *Tomek links* identificados; (c) *Tomek links* removidos (Batista et al., 2000) . 17

9 Exemplos de erros de seqüenciamento relacionados aos nucleotídeos; (a) substituição; (b) inserção; (c) remoção 18

Lista de Tabelas

1 Descrição dos conjuntos de dados utilizados nos experimentos 20
2 Resultados das SVMs 21
3 Resultados das ADs 22
4 Taxas de erros das ADs antes e após etapa de pré-processamento dos dados . . . 26
5 Taxas de erros das SVMs antes e após etapa de pré-processamento dos dados . . 27
6 Descrição do conjunto de dados 29
7 Resultados dos pré-processamentos para as SVMs 30
8 Resultados dos pré-processamentos para as ADs 31

1 Introdução

A Bioinformática é um campo interdisciplinar que consiste na aplicação de técnicas matemáticas, estatísticas e computacionais para a resolução de problemas provenientes da Biologia (Setubal & Meidanis, 1997). Atualmente, as pesquisas nessa área estão se concentrando na análise dos grandes volumes de dados gerados em laboratórios de Biologia Molecular. Técnicas inteligentes pertencentes ao campo de Aprendizado de Máquina (AM) estão entre as mais utilizadas nessas análises (Lorena & Carvalho, 2003d; Souto et al., 2003; Larrañaga et al., 2006). Esse campo da Inteligência Computacional estuda o desenvolvimento de algoritmos capazes de extrair descrições de conceitos a partir de exemplos (Mitchell, 1997), permitindo assim a extração de descrições automáticas de conceitos biológicos a partir de amostras de dados existentes.

Os dados em Bioinformática geralmente se caracterizam pela presença de ruídos, os quais podem ser provenientes de diversas fontes. A própria natureza dos experimentos laboratoriais em 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 nesses experimentos.

Embora grande parte das técnicas de AM apresentem uma robustez diante de dados ruidosos, a presença desses exemplos pode afetar o processo de indução da hipótese meta (Lavrac & Gamberger, 2001). Uma hipótese induzida a partir de um conjunto de dados sem ruídos tem maior probabilidade de ser menos complexa, além de ser mais precisa na classificação de novos dados. Isso se deve ao fato de 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 dados incorretos.

A utilização de técnicas para a detecção e eliminação de ruídos de dados biológicas constitui o tema investigado neste relatório. A seguir, na Seção 2, são apresentados os principais conceitos de Biologia Molecular necessários ao entendimento das aplicações abordadas neste relatório. Na Seção 3 são apresentados alguns conceitos de Aprendizado de Máquina. Uma descrição da etapa de pré-processamento para a identificação de ruídos é realizada na Seção 4. Em seguida, na Seção 5 são apresentadas técnicas para a eliminação de ruídos. Experimentos realizados na identificação de dados ruidosos em conjuntos de dados biológicos são descritos na Seção 6. Finalizando este relatório, na Seção 7 é apresentada uma conclusão do estudo realizado, com uma discussão de contribuições e possibilidades de trabalhos futuros.

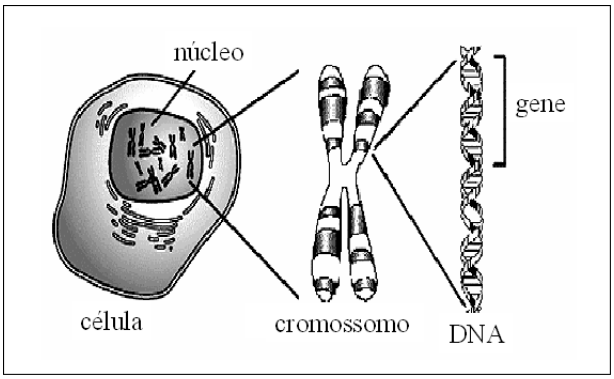

Figura 1: Célula, cromossomo, DNA, gene e suas relações

2 Conceitos de Biologia Molecular

Nesta seção é apresentada uma breve introdução aos principais conceitos de Biologia Molecular necessários ao entendimento das aplicações abordadas neste relatório.

A Biologia Molecular retrata o estudo das células e moléculas, blocos básicos utilizados na construção de todas as formas de vida (Casley, 1992). Em particular, estuda-se o genoma dos organismos, definido como o conjunto de informações genéticas dos mesmos.

Gregor Mendel, em seus experimentos realizados no século XVII, foi o primeiro a identificar fatores responsáveis pela hereditariedade nos organismos vivos (Silva, 2001). A esses fatores, que codificam a informação genética, foi dada, posteriormente, a denominação de genes. Na busca pela localização dos genes foram identificados os cromossomos. Essas estruturas possuem capacidade de replicação (reprodução) e estão presentes em todas as células. Os estudos acerca dos cromossomos, por sua vez, levaram à descoberta de que estes são compostos por moléculas de Ácido Desoxirribonucléico (DNA) e que os genes são seqüências contíguas de DNA. Na Figura 1 as estruturas citadas e suas relações são ilustradas.

Uma molécula de DNA consiste de duas fitas entrelaçadas em forma de dupla hélice. Cada fita é composta por uma seqüência de nucleotídeos, que podem ser de quatro tipos: Adenina (A), Guanina (G), Citosina (C) e Timina (T). Cada nucleotídeo de uma fita se liga a outro complementar da segunda, conforme a regra: A se liga com T, e C com G.

Um fragmento de DNA pode conter diversos genes. A propriedade mais importante dos genes está no fato de que eles codificam proteínas, componentes essenciais de todo ser vivo. Ao processo de geração de proteínas a partir da informação contida nos genes dá-se o nome de expressão gênica.

As proteínas possuem diversas funções biológicas (Lewis, 2001). Elas podem ter papel estrutural, como no caso do colágeno presente nos tendões, ou estar ligadas a atividades regulatórias, como no caso das enzimas, que catalizam diversas reações químicas nas

células. Essas moléculas também são seqüências lineares, sendo compostas de conjuntos de aminoácidos. Existe um total de 20 tipos de aminoácidos, que se combinam em uma seqüência característica para cada proteína.

A expressão gênica constitui o processo pelo qual as seqüências de nucleotídeos dos genes são interpretados na produção de proteínas. Ela é composta por duas etapas, conforme pode ser visualizado na Figura 2.

Na primeira, denominada transcrição, a enzima RNA (Ácido Ribonucléico) polimerase se liga a uma região do DNA denominada promotora e inicia a síntese de um RNA mensageiro (mRNA). O mRNA é bastante similar ao DNA, com exceção de duas características: é composto por apenas uma fita e possui o nucleotídeo Uracila (U) no lugar da Timina (T).

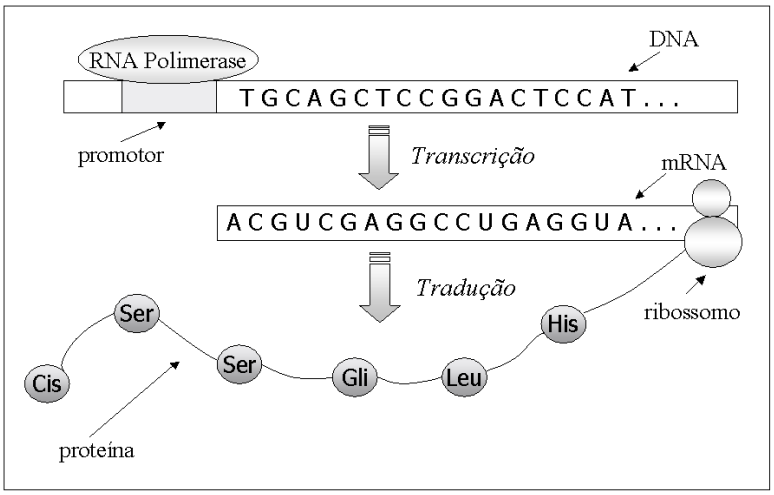

Figura 2: O processo de expressão gênica

Na segunda etapa da expressão é realizada, a partir do mRNA, a síntese da molécula de proteína. Esse processo é denominado tradução. Cada grupo de três nucleotídeos do mRNA representa um aminoácido, constituinte de uma proteína. O código genético consiste no mapeamento desses grupos, também referenciados por códons, nos aminoácidos correspondentes.

O nível de expressão de um gene constitui uma medida da quantidade de proteínas produzidas por ele em um determinado intervalo de tempo. Experimentos para medida do nível de expressão gênica são usualmente realizados visando monitorar um conjunto de genes em diferentes condições (de pacientes com e sem tratamento por uma nova droga, por exemplo) ou comparar a expressão de um gene em diferentes tipos de tecidos (normal e tumor, por exemplo) (Furey et al., 2000).

Existem algumas diferenças na forma como o processo de expressão gênica é realizado em organismos eucariotos (seres vivos complexos, tais como os seres humanos), que

possuem o material genético em um núcleo delimitado por uma membrana, e procariotos (seres unicelulares, como por exemplo as bactérias), que possuem o material genético difuso em suas células. Uma das mais importantes e que merece destaque é a de que, em organismos eucariotos, algumas partes da molécula de mRNA não são traduzidas em proteínas. O material genético dos organismos eucariotos possui, portanto, seqüências de nucleotídeos que são codificadas em proteínas, os exons, e seqüências que não participam desse processo, os introns. As fronteiras entre essas seqüências são denominadas sítios de *splicing*, nome decorrente do processo de *splicing*, em que os introns são removidos da molécula de mRNA (Figura 3).

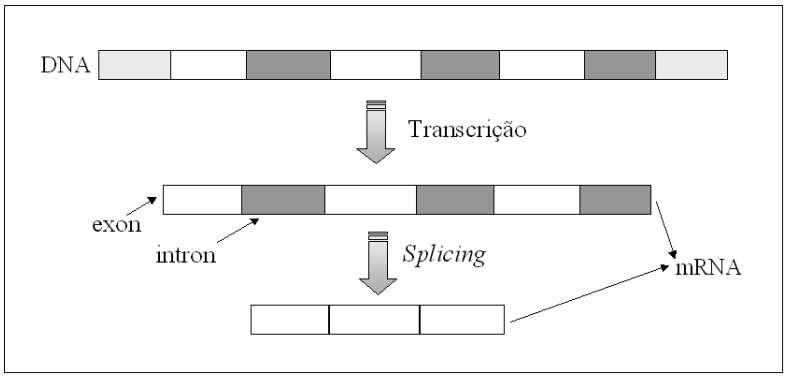

Figura 3: *Splicing* em molécula de mRNA

3 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um campo de pesquisa da Inteligência Computacional que estuda o desenvolvimento de técnicas capazes de extrair conceitos (conhecimento) a partir de amostras de dados (Mitchell, 1997).

As técnicas de AM empregam um princípio de inferência denominado indução, no qual obtém-se conclusões genéricas a partir de um conjunto particular de exemplos. O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não-supervisionado.

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada (Haykin, 1999). O algoritmo de AM extrai a representação do conhecimento a partir desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente.

No aprendizado não-supervisionado não há a presença de um professor, ou seja, não existem exemplos rotulados. O algoritmo de AM aprende a representar (ou agrupar) as

entradas submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados (Souto et al., 2003).

O tipo de aprendizado empregado neste trabalho é o supervisionado. Nesse aprendizado, dado um conjunto de exemplos rotulados, deve-se produzir um classificador (modelo, preditor ou hipótese) capaz de predizer precisamente a classe de novos dados para os quais não se conhece a classificação. Este classificador representa uma aproximação da função objetivo que descreve os dados. Essa é a função que mapeia todos os dados do domínio em que está ocorrendo o aprendizado, incluindo os disponíveis para geração do classificador, em seus respectivos rótulos. Os rótulos ou classes representam o fenômeno de interesse sobre o qual se deseja fazer previsões.

Cada exemplo (também referenciado por dado) é tipicamente representado por um vetor de características. Cada característica, também denominada atributo, expressa um determinado aspecto do exemplo. Normalmente, há dois tipos básicos de atributos: nominal e contínuo (Monard & Baranauskas, 2003). Um atributo é definido como nominal (ou categórico) quando não existe uma ordem entre os valores que ele pode assumir (por exemplo, entre cores). No caso de atributos contínuos, é possível definir uma ordem linear nos valores assumidos (por exemplo, entre pesos $\in \Re$).

Para estimar a taxa de predições corretas ou incorretas (também denominadas taxa de acerto e taxa de erro, respectivamente) obtidas por um classificador sobre novos dados, o conjunto de exemplos é, em geral, dividido em dois subconjuntos disjuntos: de treinamento e de teste. O subconjunto de treinamento é utilizado no aprendizado do conceito e o subconjunto de teste é utilizado para medir o grau de efetividade do conceito aprendido na predição da classe de novos dados. Nesse processo, os dados de teste são apresentados ao classificador induzido e as predições produzidas pelo modelo são comparadas aos rótulos conhecidos, produzindo assim a estimativa de acerto/erro. Como a taxa de acerto constitui o complemento da taxa de erro, as considerações subseqüentes irão se referir à taxa de erro.

Em diversos casos tem-se apenas um conjunto com n dados, o qual deve ser empregado na indução do classificador e também em sua avaliação. Calcular a taxa de erro do classificador nos mesmos dados utilizados em seu treinamento produz estimativas otimistas. Deve-se então utilizar métodos de amostragem alternativos para obter estimativas de acerto mais confiáveis. Entre os principais métodos existentes encontra-se o *r-fold cross-validation*.

No método de validação cruzada *r-fold cross-validation*, o conjunto de exemplos é dividido em r subconjuntos de tamanho aproximadamente igual. Os dados de $r - 1$ partições são utilizados no treinamento de um classificador, o qual é então testado sobre a

partição restante. Esse processo é repetido r vezes, utilizando em cada ciclo uma partição diferente para teste. A taxa de erro final do classificador é dada pela média dos erros observados sobre cada subconjunto de teste.

Estimado o erro de um classificador, a generalização deste é definida como a sua capacidade de prever corretamente a classe de novos dados, sendo avaliada por seu desempenho nos conjuntos de teste. No caso em que o modelo se especializa nos dados utilizados em seu treinamento, apresentando alto erro quando confrontado com novos dados, tem-se a ocorrência de *overfitting* (super-ajustamento). É também possível induzir hipóteses que apresentam alto erro mesmo no conjunto de treinamento. Tem-se então uma condição de *underfitting* (sub-ajustamento). Essa situação pode ocorrer, por exemplo, quando os exemplos de treinamento disponíveis são pouco representativos ou o modelo obtido é muito simples (Monard & Baranauskas, 2003).

A seguir, três técnicas de AM são apresentadas: o algoritmo k-vizinhos mais próximos, as Árvores de Decisão (Quinlan, 1986) e as Máquinas de Vetores de Suporte (Cristianini & Shawe-Taylor, 2000). Essas foram as técnicas de AM empregadas nos experimentos descritos na Seção 6.

3.1 k-Vizinhos Mais Próximos

O algoritmo k-vizinhos mais próximos (*k-Nearest Neighbors* - k-NN) é o representante mais simples de uma classe de técnicas de AM supervisionado denominadas baseadas em instâncias (Mitchell, 1997). Seu funcionamento reside em armazenar todas os dados de treinamento e produzir saídas baseadas em medidas de similaridade com parte desses dados. A fração de exemplos consultados em cada nova previsão são os k vizinhos mais próximos ao novo dado submetido. A classificação retornada corresponde à classe mais comum entre os rótulos desses k exemplos.

Na Figura 4 é ilustrada a operação do algoritmo k-NN em um conjunto de dados cujos exemplos são pontos no espaço bidimensional. Esse exemplo também ilustra o efeito da variação do valor do parâmetro k, cuja escolha é determinante no desempenho do algoritmo k-NN. Deseja-se classificar o dado representado por $\mathbf{x}$. Se k é escolhido como 1 (em que tem-se o algoritmo 1-NN), $\mathbf{x}$ é classificado como pertencente à classe "círculo". Empregando um valor maior de k, por exemplo, 5 (5-NN), $\mathbf{x}$ é classificado como "triângulo".

Para determinação dos vizinhos mais próximos, o k-NN utiliza uma medida para computação da distância entre pares de dados. Em geral, a medida de distância empregada é

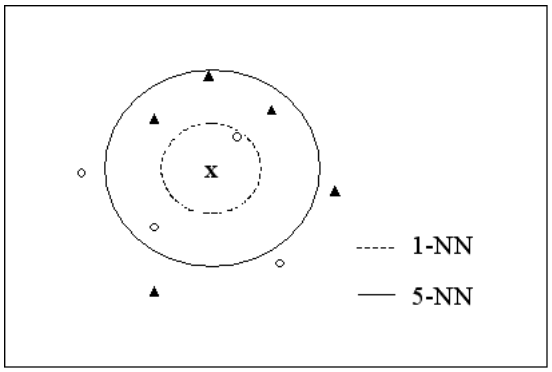

Figura 4: Ilustração do processamento realizado pelo algoritmo k-NN (Mitchell, 1997)

a Euclidiana, representada na Equação 1.

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

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

Um problema da medida Euclidiana está em pressupor que os dados correspondem a pontos no espaço m-dimensional ($\Re^m$), ou seja, que seus atributos são numéricos (contínuos). Na computação da distância entre dados com atributos categóricos, normalmente conta-se o número de posições em que os valores dos atributos diferem (Batista et al., 2000). Porém, uma medida de distância mais adequada é a VDM (*Value Difference Metric*) (Stanfill & Waltz, 1986), apresentada nas Expressões 2 e 3.

$$d\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sqrt{\sum_{q=1}^{m} \left(d_{\text{VDM}}\left(x_{iq}, x_{jq}\right)\right)^2} \tag{2}$$

$$d_{\text{VDM}}\left(x_{iq}, x_{jq}\right) = \sum_{a=1}^{C} \left| \frac{N_a\left(x_{iq}\right)}{N\left(x_{iq}\right)} - \frac{N_a\left(x_{jq}\right)}{N\left(x_{jq}\right)} \right|^b \tag{3}$$

Na Equação 3, C é o número de classes presentes no conjunto de dados, $N_a(x_{iq})$ é o número de dados da classe a com valor de atributo x_{iq}, $N(x_{iq})$ é o número total de exemplos com atributo x_{iq} e b é uma constante, frequentemente igual a 1 ou 2. Pela interpretação dessa expressão, tem-se que o VDM considera dois valores de atributos similares se eles ocorrem com freqüência relativa aproximada em todas as classes.

Para o caso em que os dados possuem atributos categóricos e contínuos, pode-se utilizar uma composição das medidas de distância Euclidiana e VDM.

Por meio de seu processamento, o k-NN possui a habilidade de construir aproximações locais da função objetivo, diferentes para cada novo dado a ser classificado. Essa característica pode ser vantajosa quando a função objetivo é muito complexa, mas ainda pode ser descrita por uma coleção de aproximações locais de menor complexidade (Mitchell, 1997). Outra vantagem particular do k-NN é sua simplicidade.

Como desvantagem principal do k-NN, assim como de outras técnicas baseadas em instâncias, tem-se o custo de classificar novos dados e de armazenar todo o conjunto de dados, os quais podem ser bastante elevados.

3.2 Árvores de Decisão

As Árvores de Decisão (ADs) organizam o conhecimento extraído dos dados em uma estrutura composta de nós e ramificações (Quinlan, 1986). Cada nó representa um atributo e está associado a um teste relevante para a classificação dos dados, ou a uma classe, no caso do nó ser uma folha (pertencente à base da AD). Os ramos representam cada um dos possíveis resultados do teste aplicado.

A classificação de um novo dado se dá seguindo os nós e as ramificações formados até que uma folha seja atingida. Um exemplo simplificado de uma AD para o diagnóstico de pacientes é ilustrado na Figura 5 (Baranauskas & Monard, 2000).

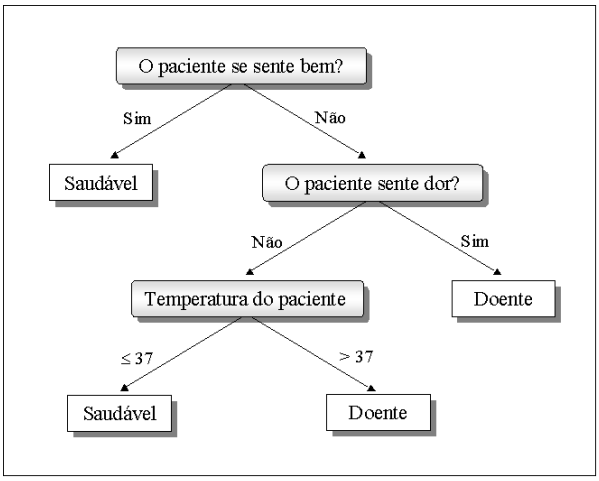

Figura 5: AD simplificada para o diagnóstico de pacientes (Baranauskas & Monard, 2000)

Há diversos algoritmos para a indução de ADs. Entre os mais utilizados estão o ID3 (Quinlan, 1986), cujas versões mais recentes denominam-se C4.5 (Quilan, 1988) e C5.0 (Rulequest-Research, 2000), e o CART (Breiman et al., 1984).

De maneira simplificada, o ID3 seleciona para teste em cada nó da AD um atributo capaz de separar melhor os dados de diferentes classes de acordo com um critério. Um tipo de critério é o ganho de informação, o qual mede a efetividade do atributo na separação dos

dados. O ID3 é aplicado recursivamente, sendo que para cada nó, exceto a raiz, o número de exemplos sobre os quais um novo teste de atributo deve ser definido é reduzido de acordo com os resultados dos testes realizados por nós anteriores. Caso todos os exemplos que chegam a um nó pertençam a uma única classe, ele é considerado uma folha, sendo então rotulado com essa classe. Caso contrário, gera-se novas ramificações e nós para cada valor que o atributo pode assumir e aplica-se o algoritmo ID3 recursivamente sobre eles.

A princípio, o ID3 é aplicado até que todos os dados do subconjunto de treinamento sejam corretamente classificados. Com isso, a AD gerada pode ser sujeita a um superajustamento aos dados de treinamento (Mitchell, 1997). Para evitar esse efeito, uma etapa de poda é geralmente aplicada à árvore treinada. A poda elimina ramos da árvore que são considerados pouco representativos. Os algoritmos C4.5 e C5.0, sucessores do ID3, implementam esse procedimento.

Uma das vantagens das ADs está na compreensibilidade das estruturas de classificação geradas. Para cada novo dado, é possível verificar quais atributos determinaram a sua classificação. Em aplicações médicas, por exemplo, muitas vezes é necessário não somente predizer uma doença, mas também especificar quais sintomas determinaram o seu diagnóstico. As ADs são, contudo, pouco robustas a exemplos de grandes dimensões (com muitos atributos). O desempenho obtido neste caso mostra-se muitas vezes inferior ao de outras técnicas de AM, como as SVMs.

3.3 Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte (SVMs - do inglês *Support Vector Machines*) constituem uma técnica de aprendizado que vem recebendo grande atenção nos últimos anos (Hearst et al., 1998). Entre as principais características que tornam seu uso atrativo em Bioinformática estão sua boa capacidade de generalização e robustez diante de dados de grande dimensão, como os presentes em grande parte das aplicações envolvendo o reconhecimento de genes e a análise de dados de expressão.

Esses resultados são alcançados pelo emprego dos conceitos da Teoria de Aprendizado Estatístico (Vapnik, 1995), que apresenta diversos limites na capacidade de generalização de um classificador linear. Para tal, dado um conjunto de treinamento S com n pares $(\mathbf{x}_i, y_i)$, em que $\mathbf{x}_i \in \Re^m$ e $y_i \in \{-1, +1\}$, as SVMs buscam o classificador linear $g(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$ capaz de separar os dados pertencentes a S com erro mínimo e que também maximize a margem ρ de separação entre as classes presentes em S.

Maximizar ρ equivale a minimizar a norma de $\|\mathbf{w}\|$ (Hearst et al., 1998). Fixando ρ em 1, tem-se o seguinte problema de otimização:

$$\textbf{Minimizar: } \|\mathbf{w}\|^2$$

$$\textbf{Sob as restrições: } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0, \text{ para } i = 1, \ldots, n$$

O hiperplano obtido na resolução deste problema é capaz de realizar a classificação de conjuntos linearmente separáveis. Para generalizá-lo a conjuntos de dados mais gerais, pode-se utilizar o artifício de permitir que alguns exemplos tenham margem menor que ρ. Isso é obtido com o relaxamento das restrições do problema de otimização apresentado, que é reformulado da seguinte maneira:

$$\textbf{Minimizar: } \|\mathbf{w}\|^2 + B \sum_{i=1}^{n} \varepsilon_i$$

$$\textbf{Sob as restrições: } \begin{cases} \varepsilon_i \geq 0 \\ y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 - \varepsilon_i \end{cases}$$

em que B é uma constante de regularização que impõe um peso diferente para o treinamento em relação à generalização e ε representa a variável responsável pela imposição do relaxamento.

Este é um problema clássico em otimização denominado programação quadrática, para o qual há uma ampla e estabelecida teoria (Hearst et al., 1998). Por meio da resolução do mesmo, obtém-se o classificador representado na Equação 4, em que os α_i correspondem a variáveis determinadas no processo de otimização. Pode-se verificar que a determinação do classificador final se dá unicamente em função de dados para os quais $\alpha_i > 0$, denominados vetores de suporte (SVs - do inglês *Support Vectors*). Esses dados correspondem aos exemplos de treinamento mais próximos ao hiperplano separador e são considerados os dados mais informativos do conjunto de treinamento.

$$g(\mathbf{x}) = \operatorname{sgn}(f(\mathbf{x})) = \operatorname{sgn}(\sum_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b) = \begin{cases} +1 \ se \sum\limits_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b > 0 \\ -1 \ se \sum\limits_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b < 0 \end{cases} \tag{4}$$

em que

$$b = \frac{1}{n_{\mathrm{SV}}} \sum_{\mathbf{x}_j \in \mathrm{SV}: \alpha_j < B} \left(\frac{1}{y_j} - \sum_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \left(\mathbf{x}_i\right) \cdot \left(\mathbf{x}_j\right) \right) \tag{5}$$

O classificador apresentado na Equação 4 ainda tem, porém, utilização limitada. Há muitos casos em que não é possível dividir satisfatoriamente os dados de treinamento por um hiperplano e uma fronteira não linear é mais adequada.

Para generalizar as SVMs para lidar com essas situações, mapea-se cada dado do conjunto de treinamento S para um novo espaço, denominado espaço de características. A escolha de uma função de mapeamento $\mathbf{\Phi}$ apropriada torna o conjunto de treinamento mapeado linearmente separável. Esse processo é ilustrado na Figura 6. SVMs lineares

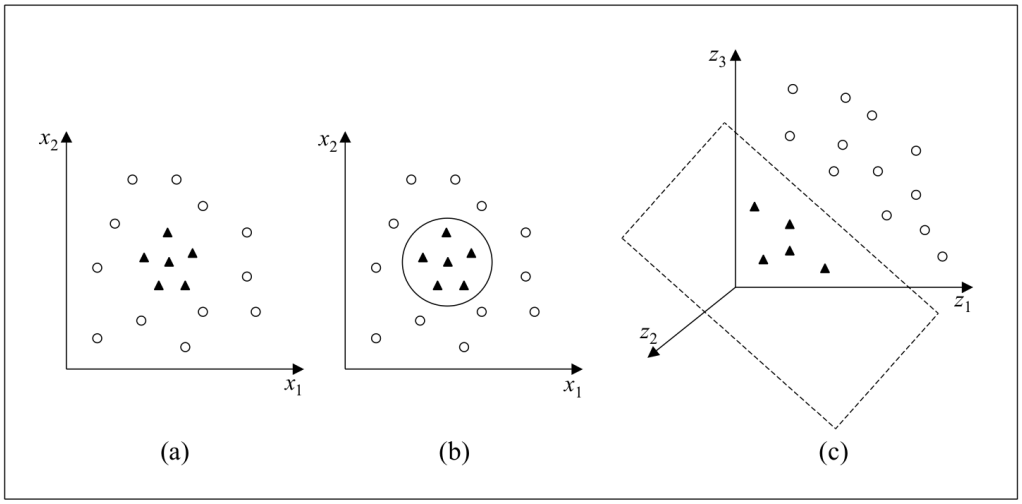

Figura 6: (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características (Müller et al., 2001)

podem então ser utilizadas sobre o conjunto de treinamento mapeado no espaço de características (Cristianini & Shawe-Taylor, 2000). Para isso, basta aplicar a função de mapeamento $\mathbf{\Phi}$ a cada exemplo nas equações listadas para o caso linear.

Por meio desse procedimento, percebe-se que a única informação necessária sobre o mapeamento é uma definição de como o produto interno $\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$ pode ser realizado. Isso é obtido com a introdução do conceito de Kernels, funções que recebem dois pontos $\mathbf{x}_i$ e $\mathbf{x}_j$ do espaço de entradas e computam o produto escalar $\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$ no espaço de características (Haykin, 1999). De maneira geral, a função Kernel é mais simples que a do mapeamento $\mathbf{\Phi}$. Por esse motivo, é comum definí-la sem conhecer-se explicitamente o mapeamento $\mathbf{\Phi}$. Alguns dos Kernels mais utilizados são os Polinomiais, os Gaussianos ou RBF (*Radial-Basis Function*) e os Sigmoidais.

Como já apontado, além de sua boa capacidade de generalização, as SVMs também se destacam por sua robustez em relação a dados de grandes dimensões. Essa característica as torna adequadas a aplicações em Bioinformática, em que os dados frequentemente se caracterizam por possuir um grande número de atributos.

Suas principais deficiências encontram-se em sua sensibilidade a escolhas de parâmetros pelo usuário e na dificuldade de interpretação do modelo gerado por essa técnica.

Deve-se destacar também que as SVMs realizam originalmente classificações binárias. Diversas aplicações em Bioinformática, porém, envolvem mais que duas classes. Existem diversas abordagens para generalizar as SVMs a problemas desse tipo. A mais comum é a decomposição "um-contra-todos" (1ct). Sendo C o número de classes, produz-se C classificadores, cada um separando uma classe i das $C - 1$ restantes. A classe de um novo dado é dada pelo índice da SVM que produz a maior saída. Uma revisão de outras

estratégias para a generalização das SVMs a problemas multiclasses pode ser consultada em (Lorena & Carvalho, 2003c; Lorena, 2006).

Um estudo mais detalhado a respeito das SVMs pode também ser consultado em (Lorena & Carvalho, 2003b; Lorena, 2006).

4 Identificação de Ruídos

Para facilitar a compreensão da etapa de identificação de ruídos, será considerado o conjunto de dados ilustrado na Figura 7a. Os exemplos desse conjunto podem ser agrupados basicamente em cinco categorias:

- **Dados com rótulo incorreto**: compreendem dados incorretamente classificados. Estes dados são considerados ruídos. Exemplos dessa categoria são destacados em 7b;

- **Dados redundantes**: dados que podem ser representadas por outros presentes no conjunto de treinamento. Alguns exemplos desse caso são apresentados na Figura 7c. Pode-se observar que estes dados formam agrupamentos no conjunto de dados. A redução 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 presentes no conjunto de dados. Esses dados podem ser ruídos ou casos muito particulares e sua influência na indução da fronteira de decisão deve ser minimizada. Exemplos desses casos são apresentados na Figura 7d;

- **Dados próximos da borda de decisão**: esses dados podem ser considerados relativamente inseguros, uma vez que mesmo uma pouca quantidade de ruídos pode movê-los para o lado incorreto da fronteira de decisão. A sua remoção pode também suavizar a borda de decisão, possibilitando uma redução no efeito de superajustamento (*overfitting*). Exemplos são destacados na Figura 7e;

- **Dados seguros**: exemplos remanescentes. Esses são os dados mais típicos de cada classe, devendo ser mantidos no treinamento dos algoritmos de AM. Os pontos destacadas na Figura 7f representam este caso. Deve-se salientar que dados redundantes também podem ser considerados seguros. Porém, a remoção de alguns deles não necessariamente leva a prejuízos à generalização do classificador, uma vez que os dados pertencentes ao mesmo agrupamento redundante mantidos representam informações similares.

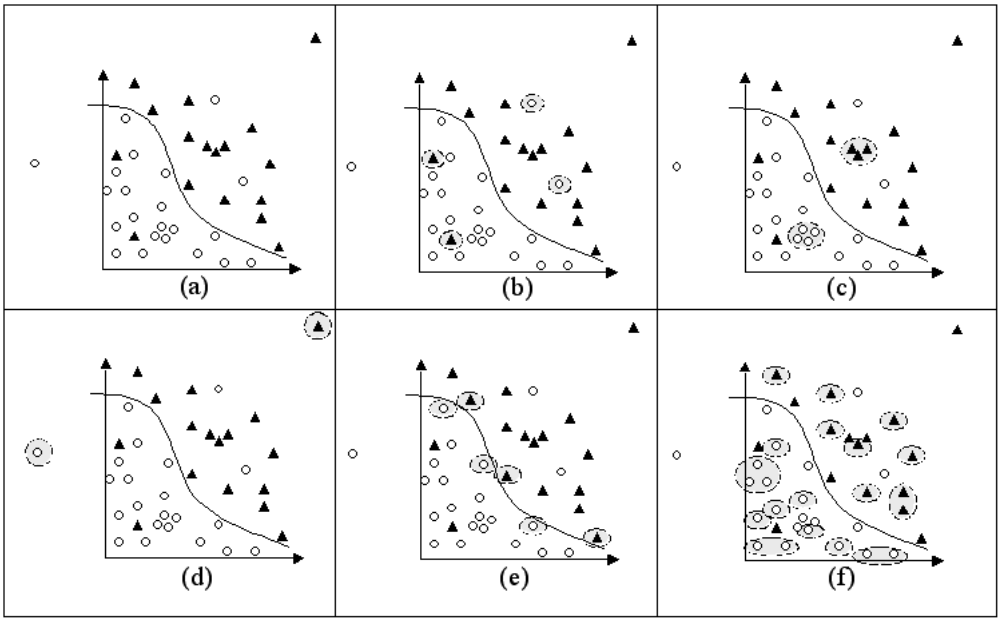

Figura 7: (a) Exemplo de conjunto de treinamento; (b) dados com rótulo incorreto; (c) dados redundantes; (d) *outliers*; (e) dados de borda; (f) dados seguros

Nas considerações subsequentes, *outliers* e dados com rótulo incorreto serão referenciados como ruídos, a menos que uma distinção explícita entre esses dois conceitos seja necessária.

A partir da categorização de dados apresentada, pode-se resumir o objetivo da etapa de identificação de ruídos como detectar e eliminar ruídos presentes nos conjuntos de dados, procurando manter os dados seguros. Esses são os dados mais representativos para a indução da hipótese meta pela técnica de AM.

5 Técnicas de Pré-processamento

Existem diversas técnicas de pré-processamento que podem ser aplicadas na detecção e remoção de ruídos. Em Estatística, esse problema é comumente solucionado por meio de técnicas baseadas em distribuição, em que os ruídos são identificados como observações que diferem de uma distribuição utilizada na modelagem dos dados (Barnett & Lewis, 1994). O maior problema dessa abordagem está em assumir que a distribuição dos dados é conhecida a priori, o que não reflete a verdade em grande parte das aplicações práticas.

Uma segunda categoria de técnicas aplicadas em Estatística utiliza o conceito de profundidade na busca por ruídos (Barnett & Lewis, 1994; Nuts & Rousseeuw, 1996). Baseado em alguma definição de profundidade, os dados são organizados em camadas. Os ruídos são identificados como objetos pertencentes a níveis superficiais. Esse tipo de estratégia

é, porém, computacionalmente cara, principalmente para dados de grandes dimensões.

Técnicas de agrupamento de dados também vêm sendo aplicadas na identificação de ruídos (Corney, 2002; He & Xu, 2002; He et al., 2003; Jin et al., 2001). Em He et al. (2003), essa estratégia é empregada na identificação de *outliers*. Pequenos agrupamentos de dados são considerados possíveis *outliers*. Uma medida baseada nessa observação foi derivada pelos autores para medir o grau com que cada exemplo pode ser considerado um *outlier*.

Em outra frente, diversos trabalhos exploraram o uso de técnicas de AM na obtenção de filtros para ruídos (Brodley & Friedl, 1999; Gamberger & Lavrac, 1997; Gamberger et al., 1999; Hawkins et al., 2002; Verbaeten, 2002; Jiang & Zhou, 2004; Escalante, 2006).

Em (Hawkins et al., 2002), por exemplo, uma Rede Neural (RN) (Braga et al., 2000) autoassociativa é utilizada na detecção desses dados. RNs autoassociativas são modelos neurais que possuem a camada de saída com mesmo tamanho que a de entrada, tentando dessa forma reproduzir cada dado apresentado em seu treinamento. Seguindo essa metodologia, os dados para os quais a rede comete maiores erros na reprodução podem ser considerados ruídos.

Em (Brodley & Friedl, 1999), um conjunto de classificadores foi empregado na filtragem de dados com rótulo incorreto. Segundo a técnica proposta, em um procedimento de validação cruzada, o conjunto de treinamento é inicialmente dividido em n partições. Fixando cada uma das partições, m algoritmos de aprendizado são treinados nos $n-1$ subconjuntos restantes. Os conjuntos de classificadores gerados são utilizados para rotular os dados da partição excluída. Cada dado recebe então m rótulos. A partir dessas classificações, o filtro compara os rótulos originais dos dados aos gerados pelos m classificadores e utiliza um critério de decisão para eliminá-los ou não. Os critérios utilizados foram de consenso e de votação por maioria. No consenso, um exemplo é removido se todos m rótulos que ele recebeu diferem de sua classe. No caso do voto por maioria, o exemplo é removido se a maioria de seus rótulos diferem de sua classe. Empiricamente, observou-se que a estratégia por consenso é mais conservativa, eliminando menos exemplos. O filtro de votação por maioria elimina mais exemplos, porém é mais provável que os dados removidos sejam casos seguros.

Gamberger & Lavrac (1997) também propuseram um filtro com o uso de técnicas de AM, baseando a eliminação de ruídos na condição de Occam. Segundo o princípio de Occam, entre as hipóteses corretas na classificação de um determinado conjunto de dados, a mais simples deve ser selecionada (Rissanen, 1978). Os autores utilizaram essa condição na formulação de uma medida de complexidade das hipóteses mais simples geradas sobre um determinado conjunto de dados. A partir dessa medida, dados ruidosos podem ser detectados como aqueles que reduzem o valor da medida quando adicionados a um conjunto

de dados. Empiricamente, essa técnica se mostrou superior ao filtro de consenso proposto em (Brodley & Friedl, 1999), o que é comprovado em (Gamberger et al., 1999). Nesse mesmo trabalho é apresentado que uma abordagem envolvendo a combinação dessas duas técnicas pode trazer resultados interessantes, com uma redução no número de exemplos eliminados, mantendo-se uma alta taxa de acerto do classificador final treinado sobre os dados filtrados.

Existem também trabalhos em que a identificação de ruídos é realizada considerando a distância desses dados a outros exemplos no conjunto de dados (Wilson, 1972; Tomek, 1976a,b; Knorr & Ng, 1998; Knorr et al., 2000; Ramaswamy et al., 2000).

Uma das primeiras aplicações dessa estratégia se deu em (Wilson, 1972), que propôs a regra *Edited Nearest Neighbor* (ENN). Segundo o ENN, um exemplo do conjunto de dados é eliminado se sua classe não é igual à da maioria de seus k vizinhos mais próximos, em que k tipicamente é igual a 3. Esse procedimento elimina ruídos e dados próximos à borda de decisão.

Uma variação desse algoritmo, denominada *Repeated ENN* (RENN), aplica o ENN repetidamente até que todos os dados remanescentes tenham a maioria de seus vizinhos com a mesma classe. Esse procedimento aumenta a margem de separação entre as classes e suaviza a fronteira de decisão.

Tomek (1976b) prôpos uma extensão do algoritmo ENN, denominada *All-kNN* (AkNN), que pode ser aplicada na detecção e remoção de ruídos e de dados de borda. Esse algorimo procede da seguinte maneira: para $i = 1, \ldots, k$, sinalizar como "ruim" qualquer dado classificado incorretamente por seus i vizinhos mais próximos. Depois de completo o loop, remover todos os dados sinalizados.

As três técnicas anteriormente descritas (ENN, RENN e AkNN) são referenciadas como filtros de ruídos em (Wilson & Martinez, 2000).

Ainda no contexto de técnicas baseadas em distância, o trabalho de Wilson & Martinez (2000) apresenta um levantamento de técnicas para a redução do número de exemplos em conjuntos de dados a serem utilizados por algoritmos de AM baseados em instâncias. Elas também incorporam procedimentos de remoção de ruídos, embora outros dados, como redundantes, também possam ser eliminados com a sua aplicação. Embora o artigo de Wilson & Martinez (2000) apresente um levantamento de um total de 15 técnicas para o pré-processamento de conjuntos de dados rotulados, apenas aquelas apontadas como mais eficazes na remoção de ruídos são descritas a seguir.

Para compreensão dos algoritmos propostos por Wilson & Martinez (2000), seja S o conjunto de treinamento original e S' o subconjunto de S obtido com a aplicação da etapa de pré-processamento. Cada dado $\mathbf{x}$ de S (e também de S') tem k vizinhos mais próximos. Define-se o "inimigo mais próximo" de $\mathbf{x}$ como o exemplo de S (S') com classe oposta à

de $\mathbf{x}$ mais próxima a este dado. Exemplos que têm $\mathbf{x}$ como um de seus k vizinhos mais próximos são denominados associados de $\mathbf{x}$. A partir dessas definições, seguem descrições das técnicas propostas em Wilson & Martinez (2000):

- *Decremental Reduction Optimization Procedure 1* (DROP1): inicialmente, tem-se $S' = S$. Um dado $\mathbf{x}$ é removido de S' se este procedimento não afeta a classificação da maioria dos exemplos associados a $\mathbf{x}$ presentes em S'. Esse algoritmo é capaz de remover ruídos e centros de agrupamentos;

- DROP2: eventualmente, o algoritmo DROP1 pode deixar ruídos nos dados. Se os vizinhos de um exemplo ruidoso são eliminados antes dele, seus associados podem vir a incluir dados da mesma classe do ruído, fazendo com que a eliminação dele leve à classificação incorreta de seus associados. Com isso, o dado ruidoso é mantido. Para lidar com esse caso, o algoritmo DROP2 elimina um dado $\mathbf{x}$ de S' se a maioria de seus associados em S (não mais em S') são classificados corretamente sem $\mathbf{x}$. Também há uma mudança na ordem em que os exemplos são removidos. Inicialmente os dados em S' são ordenados pela distância em relação a seu inimigo mais próximo. A ordem de checagem para remoção é então dada pelos exemplos mais afastados de seus inimigos mais próximos;

- DROP3: segundo o algoritmo DROP2, pontos centrais são removidos antes de pontos de borda. Com isso, pontos de borda ruidosos podem ser mantidos. A técnica DROP3 lida com essa situação realizando um passo de filtragem de ruídos antes de ordenar os dados. Esse passo consiste de uma regra similar à do ENN: qualquer dado classificado incorretamente por seus k vizinhos mais próximos é removido;

- DROP4: esse algoritmo é idêntico ao DROP3, exceto que ao invés de aplicar o ENN cegamente, o passo de filtragem de ruídos remove um dado somente se ele é classificado incorretamente por seus k vizinhos mais próximos e não vai de encontro à classificação de outros exemplos;

- DROP5: esse algoritmo modifica o DROP2 de forma que a ordem para remoção é iniciada com os dados mais próximos de seu inimigo mais próximo. Esse procedimento serve como um passo de 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 haver reduções no conjunto de dados.

- *Decremental Enconding Length* (DEL): este algoritmo é similar ao DROP3, com exceção do fato de empregar também uma heurística na decisão sobre remoções

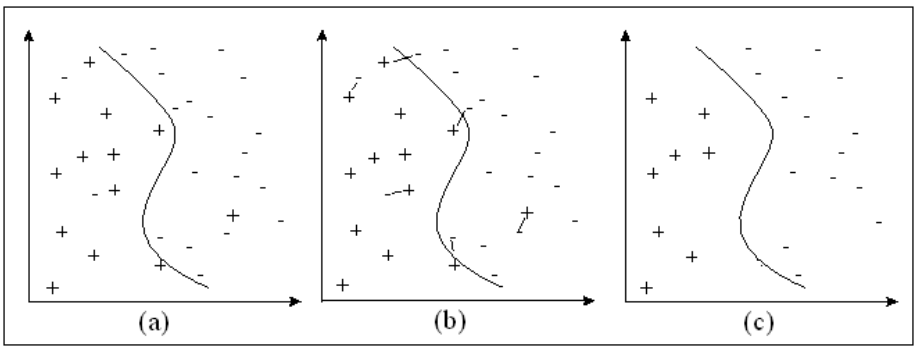

Figura 8: Aplicando *Tomek links* a um conjunto de dados; (a) Conjunto de dados original; (b) *Tomek links* identificados; (c) *Tomek links* removidos (Batista et al., 2000)

de dados no passo de filtragem de ruídos. A heurística utilizada baseia-se em um cálculo do custo acarretado pela remoção do exemplo.

Outra heurística baseada em distâncias, que pode ser utilizada na identificação de dados ruidosos e de borda, é dada pelo conceito de *Tomek links* (Tomek, 1976a). Segundo essa heurística, dados dois exemplos $\mathbf{x}_1$ e $\mathbf{x}_2$ de classes diferentes e sendo $d(\mathbf{x}_1, \mathbf{x}_2)$ a distância entre eles, o par $(\mathbf{x}_1, \mathbf{x}_2)$ é considerado um *Tomek link* se não há um dado $\mathbf{x}_3$ tal que $d(\mathbf{x}_1, \mathbf{x}_3) < d(\mathbf{x}_1, \mathbf{x}_2)$ e $d(\mathbf{x}_2, \mathbf{x}_3) < d(\mathbf{x}_2, \mathbf{x}_1)$. Os pares que formam um *Tomek link* podem representar casos de borda ou com rótulo incorreto. Na Figura 8 é ilustrado o processamento realizado pelo *Tomek links*.

Em todas as técnicas baseadas em distância apresentadas, a busca pelos vizinhos mais próximos a um determinado dado e a realização de classificações eventualmente necessárias utilizam como base o algoritmo k-NN.

Outros dois trabalhos que utilizam estratégias baseadas em distâncias são: Knorr et al. (2000) e Ramaswamy et al. (2000), que se concentram na identificação de *outliers*. Em Knorr et al. (2000), dadas duas constantes C e d, um dado $\mathbf{x}$ é considerado um *outlier* se não mais que C pontos no conjunto de dados estão a uma distância menor ou igual a d em relação a $\mathbf{x}$. Em (Ramaswamy et al., 2000), define-se o *outlier* de acordo com sua distância D^k a seu C-ésimo vizinho mais próximo. Um dado $\mathbf{x}$ é considerado um *outlier* se não mais que $n - 1$ outros exemplos têm D^k maior que $\mathbf{x}$. Com esse procedimento, tem-se um ranqueamento dos n dados com maior probabilidade de serem *outliers*.

Um problema comum a todas as técnicas baseadas em distância está na definição da métrica a ser utilizada no cálculo da distância entre os dados. É demonstrado que, para espaços de grande dimensão, várias métricas, como a Euclidiana, são pouco informativas (Beyer et al., 1999; Aggarwal et al., 2001; Hinneburg et al., 2000). Isso se deve ao fato de que os dados neste caso tendem a ser esparsos.

Seqüência meta:

A T G C T C C G

Produtos de seqüenciamento:

A G G C T C C G	A A T G C T C C G	A G C T C C G
(a)	(b)	(c)

Figura 9: Exemplos de erros de seqüenciamento relacionados aos nucleotídeos; (a) substituição; (b) inserção; (c) remoção

6 Experimentos e Resultados

Esta seção descreve experimentos realizados com o objetivo de avaliar o desempenho de algumas das técnicas listadas na seção anterior sobre conjuntos de dados de Bioinformática. Inicialmente investigou-se o efeito da aplicação dessas técnicas em dados genômicos, provenientes da aplicação de reconhecimento de sítios de *splicing*. A seguir, a etapa de pré-processamento foi aplicada a uma base de dados de expressão gênica para a classificação funcional de genes. Os resultados verificados em cada uma dessas aplicações, assim como descrições sobre as mesmas, são apresentados nas próximas seções.

6.1 Reconhecimento de Sítios de *Splicing*

O processo de seqüenciamento de DNA é sucetível a uma série de erros em cada uma de suas etapas (Setubal & Meidanis, 1997). Os erros mais simples são aqueles relacionados aos nucleotídeos, compreendendo substituições, inserções e remoções (Figura 9). Esses tipos de erros ocorrem aproximadamente 1 a 5 vezes a cada 100 nucleotídeos sequenciados (Setubal & Meidanis, 1997). Outra fonte comum de erro é a contaminação das amostras de dados sendo sequenciadas com fragmentos de DNA de organismos vetores[1] que participam do seqüenciamento.

No contexto de reconhecimento de padrões em seqüências genômicas, outro tipo de erro é a presença de dados incorretamente classificados em conjuntos de dados disponíveis para serem utilizados por técnicas de AM supervisionado.

Em organismos eucariotos, o reconhecimento completo dos genes envolve a identificação das regiões dos mesmos realmente codificam proteínas, os *exons*, e dos *introns*, porções de DNA que intermediam os *exons* e não são codificadoras. A aplicação de reconhecimento de sítios de *splicing* pode ser resumida a, dada uma seqüência de nucleotídeos de

[1] O vetor é um organismo utilizado para carregar segmentos de DNA estranho com o propósito de produzir mais cópias desse material genético (Lewin, 2001).

tamanho fixo de DNA ou mRNA, determinar se uma determinada região dessa seqüência (freqüentemente seu centro) possui uma fronteira do tipo intron-exon (IE), exon-intron (EI) ou se nenhum desses sítios está presente (N).

Sabe-se que as regiões próximas a cada um dos sítios de *splicing* possuem algumas seqüências consenso (Lewin, 2001; Vouet et al., 2002). Como as bases de AM para reconhecimento desses sítios possuem seus exemplos alinhados de forma a ter o sinal de interesse em uma mesma posição (Craven & Shavlik, 1994), é provável que seqüências de uma mesma classe compartilhem algumas características em termos de composição. Esse fator motiva a escolha das técnicas baseadas em distância no pré-processamento desses dados. Essas observações podem ser generalizadas a outras aplicações de reconhecimento de padrões em seqüências biológicas, como, por exemplo, a identificação de promotores e de sítios de início de transcrição em cadeias de DNA.

A partir das motivações apresentadas, dois estudos foram conduzidos. No primeiro, várias das técnicas baseadas em distância apresentadas na seção anterior foram avaliadas. Outro trabalho, realizado em cooperação com os pesquisadores Prof. Dr. Gustavo E. A. P. A. Batista e a Prof. Dra. Maria Carolina Monard, focalizou o uso da técnica *Tomek links* no pré-processamento dos dados.

Como os dados empregados nesses estudos possuem atributos nominais (os nucleotídeos), utilizou-se a métrica VDM (Seção 3.1) para calcular as distâncias entre os dados. O uso dessa métrica, que se baseia na distribuição dos atributos por classe, pode ser considerado adequado para a captura das semelhanças composicionais entre seqüências de nucleotídeos de uma mesma classe, além de ser capaz de lidar com diferenças comuns entre essas seqüências.

6.1.1 Avaliação de Várias Técnicas Baseadas em Distância

Este estudo compreendeu a aplicação e comparação das técnicas ENN, RENN, AkNN, DROP1 a DROP5 e DEL no pré-processamento de dois conjuntos de dados para o reconhecimento de sítios de *splicing*. Embora as técnicas DROP1 a DROP5 e DEL não sejam estritamente dedicadas à filtragem de ruídos, o seu uso pode levar a simplificações nos conjuntos de dados, por meio da eliminação de exemplos redundantes.

As avaliações foram realizadas segundo o poder preditivo de SVMs e ADs treinadas sobre os conjuntos de dados originais e pré-processados.

Conjuntos de Dados. Os conjuntos de dados utilizados nos experimentos foram o *splice* da UCI (Blake & Merz, 1998) e uma parte da base HS3D (*Homo Sapiens Splice Site Dataset*), de Pollastro & Rampone (2002). Na Tabela 1 é apresentada uma descrição desses conjuntos de dados, ilustrando o número total de exemplos (♯ Exemplos), o número

de atributos contínuos e nominais (♯ Atributos cont. e nom.) e a proporção de exemplos por classe (Classe %). Em ambos os conjuntos de dados, elementos com atributos com valor diferente dos nucleotídeos A, T, C ou G foram eliminados.

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

Conjunto	**♯ Exemplos**	**♯ Atributos (cont.,nom.)**	**Classe**	**%**
UCI	3175	60	IE, EI	25
		(0,60)	N	50
HS3D	5000	140	IE, EI	25
		(0,140)	N	25

Tabela 1: Descrição dos conjuntos de dados utilizados nos experimentos

Originalmente, a base HS3D possui 641613 exemplos. Devido a seu grande volume e ao fato dessa base ser altamente desbalanceada (com 1% de dados nas classes EI e IE e 99% dos exemplos pertencentes à classe N), apenas uma parte de seus dados, aleatoriamente selecionada, foi utilizada nos experimentos. Seguiu-se a mesma proporção de dados por classe do conjunto de dados UCI nessa seleção.

Como as SVMs lidam somente com atributos numéricos, os dados utilizados no treinamento e teste dessa técnica tiveram de ser codificados para um formato numérico. Utilizou-se para tal uma codificação canônica sugerida no trabalho de Pedersen & Nielsen (1997). Segundo essa codificação, cada nucleotídeo (atributo) das seqüências é representado por quatro bits, da seguinte maneira: A = (0 0 0 1), C = (0 0 1 0) e G = (0 1 0 0) e T = (1 0 0 0).

Metodologia. Inicialmente, avaliou-se o efeito da realização da etapa de pré-processamento dos dados no desempenho das SVMs. As técnicas ENN, RENN, AkNN, DROP1 a DROP5 e DEL foram então aplicadas aos conjuntos de treinamento obtidos pelo *10-fold cross-validation*. Para tal, utilizou-se o código disponibilizado pelos autores Wilson & Martinez (2000) em `ftp://axon.cs.byu.edu/pub/randy/ml/drop`. Diferentes valores do parâmetro k das técnicas de pré-processamento, que determina o número de vizinhos mais próximos no k-NN, foram testados: 1, 3 (valor default da ferramenta), 5, 7, e 9.

SVMs foram então treinadas sobre os conjuntos de dados originais e pré-processados. Para indução das SVMs, utilizou-se o software SVMTorch II (Collobert & Bengio, 2001). Diversos tipos de funções Kernel foram empregados: Polinomiais de graus 1 a 5 e Gaussianos com diversos valores de desvio-padrão (5, 10, 25 e 50). Variou-se também o valor da constante de regularização das SVMs, referenciado neste documento por B. Essa constante relaciona-se à maneira como as SVMs lidam com ruídos, impondo maiores ou

menores restrições às margens da SVM. Quanto menor o valor de B, mais suave é a fronteira de decisão. Os valores de B empregados foram: 25, 50, 100, 150 e 200.

ADs, induzidas pelo algoritmo C4.5 (Quilan, 1988), também foram aplicadas ao problema, tendo sido treinadas sobre os dados originais e pré-processados, como no caso das SVMs.

Para conversão dos diversos conjuntos de dados gerados durante os experimentos para os formatos exigidos pelo código de Wilson & Martinez (2000), do SVMTorch II e do C4.5, e também para determinação dos melhores modelos de SVMs e ADs em cada caso, foram implementadas rotinas na linguagem de programação Perl.

Resultados. Na Tabela 2 são apresentados os melhores modelos de SVMs (Kernel e B) para cada conjunto de dados, o melhor valor de k para as técnicas de pré-processamento (o valor que levou às menores taxas de erros no posterior treinamento das SVMs), as taxas médias de erro das SVMs induzidas (Erro) e o número médio de dados nas partições de treinamento utilizadas em cada caso (N^o dados). Esse número fornece uma idéia da quantidade de dados eliminada com a aplicação das técnicas de pré-processamento. As siglas "Gs" e "Pol" representam funções Kernel do tipo Gaussiano e Polinomial, respectivamente, e os valores de seus parâmetros encontram-se indicados em parênteses. As melhores e piores taxas de erro em cada conjunto de dados encontram-se destacadas em negrito e itálico, respectivamente.

	UCI					**HS3D**				
	k	**Kernel**	B	**Erro**	**N^o dados**	k	**Kernel**	B	**Erro**	**N^o dados**
original	-	Gs (5)	25	**2,9±1,2**	2857,5±0,5	-	Gs (10)	200	**11,3±0,8**	4500,0±0,0
ENN	5	Pol (4)	25	3,0±1,4	2444,5±3,6	5	Pol (2)	25	12.5±1.1	3410.7±22.1
RENN	3	Pol (4)	25	3,0±1,4	2436,5±8,2	1	Pol (2)	25	12,4±1,2	3382,3±21,5
AkNN	3	Pol (4)	25	3,1±1,3	2403,5±9,6	1	Pol (2)	25	12,7±0,8	2988,2±449,6
DROP1	9	Pol (2)	25	17,0±1,5	169,6±17,7	9	Pol (2)	25	*25,4±2,5*	736,0±23,3
DROP2	1	Pol (2)	25	11,2±2,6	249,3±4,2	5	Pol (2)	25	21,1±1,4	688,8±104,4
DROP3	9	Gs (10)	25	13,3±1,8	234,1±10,0	9	Pol (2)	25	23,1±1,5	735,4±21,4
DROP4	9	Gs (10)	25	12,2±1,2	296,1±6,4	5	Pol (2)	25	24,7±2,2	743,9±115,2
DROP5	9	Pol (2)	25	11,0±1,7	249,3±6,8	5	Pol (2)	25	22,6±1,3	752,3±16,4
DEL	1	Gs (10)	25	*18,9±2,8*	170,7±7,3	1	Gs (10)	25	22,0±11,7	1310,9±12,1

Tabela 2: Resultados das SVMs

Os resultados verificados nos experimentos com ADs encontram-se na Tabela 3. Essa tabela ilustra o melhor número de vizinhos mais próximos k das técnicas de pré-processamento, as taxas médias de erros das ADs (Erro) e o número médio de dados nas partições de treinamento empregadas em cada caso (N^o dados). Como na Tabela 2, as melhores e piores taxas de erros encontram-se destacadas em negrito e itálico, respectivamente.

	UCI			HS3D		
	k	**Erro**	**N^o dados**	k	**Erro**	**N^o dados**
original	-	8,1±1,5	2857,5±0,5	-	23,9±1,7	4500,0±0,0
ENN	3	6,5±1,4	2450,8±5,0	9	**21,7±1,7**	3432,4±10,5
RENN	9	**6,5±1,2**	2410,2±6,7	7	21,9±2,0	3343,1±13,1
AkNN	9	**6,5±1,2**	2370,6±10,8	5	22,9±1,9	2808,8±425,3
DROP1	1	*31,8±3,0*	99,5±6,3	7	*35,4±2,1*	683,4±29,5
DROP2	9	26,8±3,5	343,5±16,0	1	31,9±4,1	688,8±104,4
DROP3	9	26,8±5,9	234,1±9,9	3	30,2±3,0	648,8±11,5
DROP4	9	30,3±5,1	296,1±6,4	7	34,8±3,2	806,2±128,6
DROP5	3	29,1±3,5	189,4±11,3	9	28,8±2,9	856,5±24,1
DEL	7	31,6±4,5	134,9±7,2	9	32,5±3,5	650,2±27,6

Tabela 3: Resultados das ADs

Discussões. Aplicando o teste t para dados pareados descrito em (Nadeau & Bengio, 2003) às taxas de erros das SVMs (Tabela 2), verifica-se que a utilização das técnicas de pré-processamento ENN, RENN e AkNN manteve os resultados observados sobre os dados originais em ambos conjuntos de dados (UCI e HS3D), com 95% de confiança. Pode-se também considerar que os modelos de SVMs obtidos sobre os conjuntos de dados pré-processados segundo essas técnicas foram simplificados (de uma Gaussiana, que possui dimensão VC infinita (Burges, 1998), a uma Polinomial, que possui complexidade controlada por seu grau (Burges, 1998; Haykin, 1999)). No caso das técnicas DROP1 a DROP5 e DEL, houve uma clara piora dos resultados. Esse fator pode ser atribuído às reduções drásticas nos conjuntos de dados com a aplicação dessas técnicas, conforme ilustrado nas colunas N^o dados da Tabela 2. O exame dessa coluna, tanto na Tabela 2 quanto na Tabela 3, também indica que as técnicas AkNN, DROP2 e DROP4 apresentaram maiores variações no número de dados removidos nas partições, o que pode ser observado nas altas taxas de desvio-padrão obtidas.

O emprego das técnicas de pré-processamento ENN, RENN e AkNN também trouxe benefícios em termos de reduções no tempo despendido no posterior treinamento das SVMs. As reduções verificadas foram de aproximadamente 20% no conjunto UCI e 75% no conjunto HS3D. Os experimentos foram conduzidos em um processador dual de 2.4 GHz e 1 GB de memória.

No conjunto HS3D também verificou-se, nesses casos, uma diminuição da ordem de 40% no número de SVs dos modelos gerados. Como a equação do hiperplano ótimo no espaço de características é determinado unicamente por esses dados, um número menor de SVs leva a menores tempos de predição. No conjunto UCI, o número de SVs das SVMs geradas sobre os conjuntos pré-processados pelas técnicas ENN, RENN e AkNN se

mantiveram similares ou aumentaram um pouco (cerca de 2% de crescimento).

Para as ADs, o exame da Tabela 3 indica que a aplicação das técnicas de pré-processamento ENN, RENN e AkNN levou a reduções nas taxas de erros, embora elas não possam ser consideradas estatisticamente significantes a 95% de confiança estatística. Grandes reduções no tamanho médio das ADs também foram verificadas com o emprego dessas técnicas (com mais de 50% de reduções). Esses resultados indicam simplificações nos modelos induzidos, com manutenção, e até mesmo pequenas quedas, das taxas de erro. Além disso, reduções nas ADs refletem em ganhos de compreensibilidade dos modelos obtidos, no sentido de que árvores menores podem ser consideradas mais fáceis de serem examinadas.

No caso das técnicas DROP1 a DROP5 e DEL verificou-se grandes acréscimos nas taxas de erros das ADs. Novamente, pode-se atribuir esse comportamento a reduções indesejadas nos conjuntos de dados com a aplicação desses pré-processamentos.

De forma geral, os resultados obtidos sugerem que a aplicação das técnicas de filtragem de ruídos ENN, RENN e AkNN manteve os dados mais expressivos nos conjuntos de dados, o que foi refletido na manutenção das taxas de erros verificadas, que, no caso das ADs, foram até mesmo reduzidas. Como outros benefícios, foram observadas simplificações nos classificadores induzidos. Esse fator vem de encontro ao princípio de Occam, que afirma que, entre diversos modelos com um mesmo erro, deve-se escolher aquele de menor complexidade.

No caso das técnicas DROP1 a DROP5 e DEL, reduções muito drásticas foram observadas nos dados de treinamento. Isso refletiu em prejuízos no desempenho alcançado pelas técnicas de aprendizado. Logo, essas estratégias de pré-processamento não foram capazes de manter os dados mais significativos nos conjuntos de dados, e provavelmente eliminaram dados seguros. Elas se mostraram, portanto, pouco criteriosas na remoção de dados das bases de dados biológicas consideradas.

Comparando ainda os resultados das ADs e das SVMs em termos de taxas de erros, pode-se verificar um melhor desempenho das SVMs. Esse fato pode ser confirmado estatisticamente, a 95% de confiança. As maiores taxas de erros das ADs podem ser conferidas a sua dificuldade em lidar com conjuntos de dados de grande dimensão, como os investigados nesses experimentos. Em contrapartida, as SVMs são robustas em relação esse tipo de dados. Porém, se o interesse é compreensibilidade, o uso de ADs é mais indicado, pois a forma como o conhecimento é extraído e representado pelas SVMs pode ser considerado de difícil interpretação. Ainda para as ADs, o uso das técnicas de pré-processamento ENN, RENN e AkNN possibilita maiores ganhos em termos de compreensibilidade.

Conclusões. Por meio dos experimentos e resultados reportados, pode-se verificar que

os maiores benefícios alcançados na aplicação de uma etapa de pré-processamento sobre dados genômicos se deram em simplificações nos modelos de AM induzidos. Em alguns casos, reduções nas taxas de erros também foram verificadas. Porém, mais experimentos, inclusive sobre outros problemas, são necessários para uma avaliação mais ampla da adequabilidade e benefícios da aplicação dos procedimentos de pré-processamento nesse domínio.

Como sugestão de trabalho futuro, modificações podem ser realizadas na medida de distância empregada nas técnicas de pré-processamento baseadas em distância. Na medida da distância entre seqüências de DNA, pode-se recorrer, por exemplo, a técnicas de alinhamento. O alinhamento pode ser definido como um procedimento no qual uma (ou mais) seqüência é "colocada sobre outra" de forma a tornar clara a correspondência entre catacteres similares ou subcadeias das seqüências analisadas (Setubal & Meidanis, 1997). Por esse processo, pode-se quantificar a similaridade entre duas (ou mais) seqüências, ou seja, quanto elas são semelhantes.

No contexto de reconhecimento de sinais em seqüências genômicas em particular, é comum que as seqüências com o sinal de interesse possuam regiões consenso (Stormo, 2000). Na identificação de promotores de organismos procariotos, por exemplo, vários sistemas buscam identificar regiões como o *TATAbox* (rica nos nucleotídeos A e T), presentes 10 nucleotídeos acima do sítio de início de tradução (Stormo, 2000). Como já discutido, as próprias bases de dados em AM para reconhecimento de sinais possuem seus dados alinhados de forma a possuir o sítio de interesse em uma mesma posição (Craven & Shavlik, 1994). Esses fatores justificam e motivam o uso de uma métrica baseada em alinhamentos. A utilização dessa métrica pode levar também à obtenção de uma estratégia de pré-processamento mais próxima da realidade biológica. Esse tipo de modificação pode ainda beneficiar técnicas de pré-processamento baseadas em agrupamentos, que utilizam medidas de similaridade na formação de aglomerados de dados.

Parte do trabalho experimental apresentado nesta seção foi apresentado no I ICOBICOBI (I *Iternational Conference on Bioinformatics and Computational Biology*) (Lorena & Carvalho, 2003a). Outra fração, no periódico *Genetics and Molecular Biology* (Lorena & Carvalho, 2004).

6.1.2 Avaliação da Heurística *Tomek links*

Este estudo consistiu na aplicação da heurística *Tomek lins*, descrita na Seção 5, em uma etapa de pré-processamento sobre dados utilizados no reconhecimento de sítios de *splicing*. Avaliou-se a capacidade dessa técnica em eliminar dados ruidosos e manter aqueles mais informativos, por meio da avaliação do desempenho de duas técnicas de

AM treinadas sobre os conjuntos de dados pré-processados. As técnicas de AM aplicadas foram as ADs e as SVMs.

Conjunto de Dados. O conjunto de dados utilizado nos experimentos foi o *splice* da UCI (Blake & Merz, 1998), descrito na Tabela 1. Para o caso das ADs, utilizou-se os dados originais, compostos de 60 atributos nominais assumindo os valores A, T, C e G. Para as SVMs, que requerem atributos com valores numéricos, utilizou-se a codificação canônica apresentada na Seção 6.1.1.

Metodologia. O problema de identificação de sítios de *splicing* foi reformulado como dois subproblemas binários. Baseado no conhecimento do domínio, a divisão realizada foi a seguinte: inicialmente gerou-se classificadores para distingüir seqüências que têm sítios de *splicing* (IE+EI) das que não possuem (N). Se a presença de um sítio de *splicing* é verificada, um segundo classificador discrimina seu tipo, se IE ou EI. Para simplicidade de referência, o subproblema IE+EI versus N será referenciado como "detecção de sítios de *splicing*" (DSS) e o subproblema IE versus EI será nomeado "identificação de sítios de *splicing*" (ISS). Esta reformulação do problema multiclasses foi realizada devido ao fato das SVMs serem originalmente binárias e também da implementação da técnica de pré-processamento empregada ser binária. Cada um dos conjuntos obtidos com a realização dessa separação foi ainda dividido segundo o método *10-fold cross-validation*.

A técnica de pré-processamento empregada nos experimentos foi baseada na heurística *Tomek links* (Tomek, 1976a), descrita na Seção 5. Essa heurística permite a identificação de dados com classificação incorreta (ruídos) e casos de borda, dados considerados pouco confiáveis. Esse pré-processamento foi aplicado a cada um dos conjuntos de treinamento das partições geradas pelo método *10-fold cross-validation*. Esse procedimento levou à eliminação de aproximadamente 6% dos dados de treinamento no experimento DSS e 5% para o caso do subproblema ISS.

A indução de classificadores segundo as técnicas ADs e SVMs foram realizadas com as ferramentas C4.5 (Quilan, 1988) e SVMToch II (Collobert & Bengio, 2001), respectivamente.

Para o caso das SVMs, diversos tipos de funções Kernel foram testadas. As funções consideradas foram: Polinomiais de diferentes graus d (1 a 5) e Gaussianas com valores variados de desvio-padrão σ (0.01, 0.1, 1, 5, 10, 50 e 100). Outros parâmetros foram mantidos com os valores default do SVMTorch II (Collobert & Bengio, 2001).

Relizados os experimentos, o efeito da aplicação do pré-processamento foi analisado para cada técnica de AM individualmente e, em seguida, para o conjunto das técnicas de

	Experimento DSS			Experimento ISS		
Conjunto de dados	**Total**	**IE+EI**	**N**	**Total**	**IE**	**EI**
Original	**4,4±0,6**	**1,9±0,6**	**6,7±0,9**	**4,4±1,5**	**5,3±2,3**	**3,3±2,8**
Pré-processado	4,5±0,9	1,9±1,0	6,8±1,0	4,5±1,6	5,3±2,4	3,7±2,4

Tabela 4: Taxas de erros das ADs antes e após etapa de pré-processamento dos dados

aprendizado consideradas.

Resultados e Discussões. Como discutido anteriormente, para o caso das ADs, uma propriedade importante é sua compreensibilidade, que é em geral melhor para árvores pequenas. Uma avaliação pertinente neste caso diz respeito então ao ganho de compreensibilidade das ADs induzidas após a fase de pré-processamento.

No experimento DSS, o tamanho médio das ADs induzidas caiu de $230,6 \pm 8,0$ nós para $209,0 \pm 14,3$ nós com a aplicação do pré-processamento. Verifica-se, assim, uma redução de aproximadamente 10%. Para o experimento ISS, a redução verificada foi de $88,2 \pm 9,6$ nós para $81,8 \pm 8,8$ nós (aproximadamente 10%). Em ambos casos, portanto, tem-se um indicativo de algum ganho de compreensibilidade. Esses dados referem-se às ADs obtidas após o processo de poda, em que ramos da árvore pouco representativos são eliminados. Por meio da poda, portanto, a influência de dados ruidosos é minimizada. Para obter-se uma melhor medida das reduções de árvores obtidas com o pré-processamento, mediu-se também as reduções obtidas antes do procedimento de poda. Para o caso do experimento DSS, o tamanho médio das árvores foi reduzido de $690,6 \pm 16,0$ nós para $591,4 \pm 25,0$ nós, o que fornece uma medida de aproximadamente 15% de redução com o pré-processamento. No caso do subproblema ISS, essa diminuição foi de aproximadamente 33% ($295,4 \pm 17,8$ para $197,0 \pm 16,9$ nós). Esses fatos indicam que a heurística *Tomek links* pode ser considerada responsável por uma redução no nível de dados pouco informativos.

Na Tabela 4 é apresentado o desempenho das ADs induzidas. Nessa tabela apresenta-se, para cada experimento (DSS e ISS), a taxa de erro total (Total), assim como as taxas de erro observadas em cada classe (IE+EI, N, IE e EI). As menores taxas de erro encontram-se destacadas em negrito. Pode-se observar que os erros obtidos em ambos experimentos após a aplicação do pré-processamento são similares aos verificados sem a realização dessa fase. Aplicando-se o teste estatístico pareado de Nadeau & Bengio (2003), pode-se verificar que as diferenças nos resultados reportados não são estatisticamente significantes, com 95% de confiança. Logo, o desempenho geral da técnica foi mantido, enquanto ganhos em termos de compreensibilidade e simplificação do modelo foram obtidos.

No caso das SVMs, os melhores resultados para os dados originais no experimento DSS foram alcançados pelo Kernel Gaussiano com um desvio-padrão de 5. Para os dados

	Experimento DSS			Experimento ISS		
Conjunto de dados	**Total**	**IE+EI**	**N**	**Total**	**IE**	**EI**
Original	3,6±0,9	1,3±0,8	2,2±0,6	**1,9±1,1**	**1,7±1,1**	**2,1±2,3**
Pré-processado	**2,9±0,8**	**1,0±0,8**	**2,0±0,4**	2,1±1,2	1,9±0,9	2,5±2,4

Tabela 5: Taxas de erros das SVMs antes e após etapa de pré-processamento dos dados

pré-processados, os melhores resultados foram obtidos com um Kernel Polinomial de terceiro grau (um modelo mais simples). No experimento ISS, o melhor Kernel para ambos conjuntos de dados (original e pré-processado) foi um Polinomial de quinto grau.

Para as SVMs, outra medida computada foi a redução no número de SVs dos modelos gerados, que correspondem aos dados mais representativos para o processo de classificação. Como o classificador final nas SVMs é definido por esses dados, a presença de exemplos ruidosos pode influenciar a sua determinação. Além disso, um número de SVs menor reflete em maiores velocidades na fase de classificação de novos dados. No subproblema DSS, a fase de pré-processamento levou a uma redução de aproximadamente 10% no número de SVs. No caso do experimento ISS, esse decréscimo foi de aproximadamente 6%.

Também verificou-se uma redução no tempo de treinamento das SVMs com o pré-processamento, o qual foi de 7% para o experimento DSS e 15% no caso do ISS. Os experimentos foram conduzidos em um processador Pentium II dual de 300 MHz e 128 MB de memória.

Similarmente à Tabela 4, na Tabela 5 são apresentadas as taxas de erro das SVMs na fase de teste. Pode-se observar que o desempenho geral das SVMs foi mantido (enquanto ganhos foram alcançados em reduções no tempo de treinamento e número de SVs). Para o experimento DSS, em particular, os resultados obtidos com a realização da etapa de pré-processamento foram melhores, com 95% de nível de confiança.

Como última análise, realizou-se uma comparação estatística entre os resultados alcançados pelas técnicas de AM empregadas. Foi verificado que as SVMs foram superiores às ADs em todos casos, com significância de 95%. Como no primeiro experimento aprensentado nesta seção, esse resultado pode ser atribuído à pouca robustez das ADs em relação a dados com um grande número de atributos.

Conclusões. Como conclusões do trabalho realizado, tem-se que a aplicação de uma etapa de pré-processamento com a heurística *Tomek links* levou a melhoras no processo de indução das técnicas de AM, sem perda significante de generalidade (para o caso das SVMs no subproblema DSS houve até mesmo reduções significativas nas taxas de erro).

Para as ADs, a maior contribuição foi obtida em reduções no tamanho médio das

árvores induzidas, o que resulta em ganhos de compreensibilidade e simplificações dos modelos induzidos. Para as SVMs, decréscimos significantes foram observados no tempo de treinamento. Resultados semelhantes foram verificados em experimentos com Redes Neurais (RNs) do tipo MLP (*Multilayer Perceptron*) (Braga et al., 2000) sob o subproblema ISS, em que o pré-processamento dos dados levou a uma redução de aproximadamente 20% no tempo de treinamento das redes. Esses resultados são apresentados em (Lorena et al., 2002b). Para as SVMs, obteve-se também uma redução no número de SVs, sugerindo que o hiperplano ótimo no espaço de características pode ter se tornado menos orientado por dados pouco confiáveis. Esse decréscimo também traz benefícios em termos de aumentos na velocidade da fase de predição.

Apesar das melhoras mencionadas, o custo da etapa de pré-processamento deve ser levado em consideração. Para n exemplos com m atributos, essa fase tem complexidade da ordem de $m \cdot n^2$. Este é, portanto, um processo custoso. Essa fase é, porém, aplicada uma única vez, independente do número de técnicas de AM empregadas.

Além disso, uma deficiência na aplicação da heurística *Tomek links* utilizada deve ser apontada. Esta decorre do fato da implementação corrente não ser capaz de determinar precisamente qual dos exemplos em um *Tomek link* constitui o ruído. O procedimento adotado foi de eliminar ambos os dados, o que leva à eliminação de casos seguros. O exame de uma vizinhança ao redor dos exemplos que formam um *Tomek link* pode ser uma alternativa para solução desse problema.

Como última conclusão, pode-se afirmar que os dados empregados nos experimentos não possuem um nível elevado de ruídos. Isso pode ser devido a possíveis pré-processamentos anteriormente aplicados a esses dados, uma vez que eles foram utilizados em diversos trabalhos prévios (ex. (Noordewier et al., 1991; Towell, 1991; Towell & Shavlik, 1992)).

Parte deste trabalho foi apresentado no VII Simpósio Brasileiro de Redes Neurais (VII SBRN) (Lorena et al., 2002b). Outra fração, no I Workshop Brasileiro de Bioinformática (WOB 2002) (Lorena et al., 2002a).

6.2 Classificação Funcional de Genes

Esta seção apresenta estudos complementares na avaliação do emprego de técnicas de pré-processamento em Bioinformática. Os pré-processamentos foram então aplicados a um conjunto de dados de medidas 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 tal, recorre-se à análise do nível de expressão deste gene em diferentes experimentos laboratoriais. Tem-se então valores numéricos a serem analisados, os quais

		Classe %				
♯ Exemplos	**♯ Atributos (cont.,nom.)**	**1**	**2**	**3**	**4**	**5**
207	79 (79,0)	58,0	5,3	16,9	13,0	6,8

Tabela 6: Descrição do conjunto de dados

seguem algum padrão de acordo com sua a classificação.

Assim como na seção anterior, o efeito da aplicação dos pré-processamentos foi avaliado pela verificação da diferença de desempenho de SVMs e ADs treinadas sobre os dados origianis e pré-processados, respectivamente.

As técnicas aplicadas foram as mesmas apresentadas na Seção 6.1.1. São elas: ENN, RENN, AllKNN, DROP1 a DROP5 e DEL.

Como na aplicação em Bioinformática investigada neste caso os dados possuem valor numérico, a medida de distância empregada na computação dos k vizinhos mais próximos a um elemento foi a Euclidiana, definida pela Equação 1.

Conjunto de dados. O conjunto de dados empregado nas análises experimentais conduzidas foi o disponibilizado por (Brown et al., 1999). Esse conjunto apresenta medidas do nível de expressão de 2467 genes em 79 experimentos laboratoriais distintos. Desses genes, 207 encontram-se categorizados em cinco classes funcionais e foram então selecionados para realização dos experimentos neste trabalho. O conjunto de dados final empregado encontra-se descrito na Tabela 6, que apresenta o número total de exemplos no mesmo (♯ Exemplos), o número de atributos contínuos e nominais de seus dados (♯ Atributos cont. e nom.) e a distribuição de dados por classe (Classe %).

De forma a prover estimativas dos erros dos classificadores induzidos neste estudo, esse conjunto de dados foi dividido segundo o método *10-fold cross-validation*.

Metodologia. Todas as partições de treinamento geradas pelo procedimento de *cross-validation* foram pré-processadas segundo as técnicas ENN, RENN, AkNN, DROP1 a DROP5 e DEL. SVMs e ADs foram então treinadas nas partições originais e pré-processadas e seus resultados foram avaliados nas partições de teste correspondentes.

Todos os experimentos com SVMs foram realizados com a ferramenta SVMTorch II (Collobert & Bengio, 2001). Diversos tipos de Kernel e variações de parâmetros foram testados. Com relação às funções Kernel, empregou-se os tipos Polinomial (com graus de 1 a 5) e Gaussiano (com desvio-padrão de 5, 10, 35 e 50). Também variou-se o parâmetro B das SVMs. Os valores utilizados foram: 1, 50, 100, 150, 200.

A indução das ADs foi realizada com o algoritmo C4.5 (Quilan, 1988).

Na aplicação das técnicas de pré-processamento, utilizou-se o código disponibilizado

Conjunto	k	**Kernel**	B	**Erro**	**N^o dados**
original	-	G (50)	50	4,8±5,9	186,3±0,5
ENN	7	P (1)	1	*8,2±5,0*	153,1±1,5
RENN	7	P (1)	1	*8,2±5,0*	151,4±3,2
AKNN	1	G (50)	50	7,7±6,4	153,7±1,8
DROP1	3	P (1)	1	7,3±5,7	15,5±2,1
DROP2	3	P (1)	1	6,7±7,5	19,8±1,6
DROP3	1	P (1)	1	7,2±6,1	13,2±1,9
DROP4	1	P (1)	1	6,7±6,5	13,6±1,5
DROP5	3	G (25)	50	6,2±5,5	17,3±2,7
DEL	1	P (1)	1	**3,8±4,9**	19,4±1,3

Tabela 7: Resultados dos pré-processamentos para as SVMs

por Wilson & Martinez (2000). Também variou-se os valores da constante k, que determina o número de vizinhos mais próximos do k-NN. Os valores empregados foram: 1, 3, 5, 7 e 11.

Os resultados destes experimentos encontram-se sintetizados a seguir.

Resultados. Na Tabela 7 são apresentados os resultados observados no treinamento e teste das SVMs. Como na Tabela 2, essa tabela apresenta, para cada conjunto pré-processado (e também para o original), o melhor valor da constante k, o tipo de Kernel e a constante B da melhor SVM (com menor taxa de erro), a taxa de erro média verificada nos conjuntos de teste e também o número médio de dados nas partições de treinamento. As melhores e piores taxas de erros verificadas em cada conjunto de dados encontram-se destacadas em negrito e itálico, respectivamente.

Na Tabela 8 são apresentados os resultados verificados no treinamento das ADs. Como na Tabela 3, são apresentados, para cada conjunto de dados, o melhor valor da constante k, a média e desvio-padrão dos erros nas partições de teste e o número médio de dados de treinamento em cada caso. As melhores e piores taxas de erros encontram-se destacadas em negrito e itálico, respectivamente.

Discussões. Na análise dos resultados obtidos no treinamento e teste das SVMs (Tabela 7), observa-se uma redução na taxa média de erro com a aplicação da técnica DEL. A diferença desse resultado em relação ao verificado sobre os dados originais não é, porém, estatisticamente significante a 95% de confiança. Além disso, o emprego da técnica DEL levou a simplificações no modelo das SVMs (de um Kernel Gaussiano a um linear - Polinomial de primeiro grau).

Para as outras técnicas consideradas, foram verificados pequenos aumentos nos erros das SVMs, embora eles não possam ser considerados estatisticamente significantes a 95%

Conjunto	k	**Erro**	**N^o dados**
original	-	**8,2±4,6**	186,3±0,5
ENN	1	9,2±6,1	155,1±1,7
RENN	1	9,7±6,3	154,5±1,8
AKNN	1	8,2±7,5	153,7±1,8
DROP1	11	24,1±10,9	30,5±4,5
DROP2	11	17,8±8,6	34,7±2,2
DROP3	11	24,2±12,9	28,8±4,7
DROP4	11	20,2±11,3	33,3±1,8
DROP5	11	27,1±16,4	28,2±3,3
DEL	5	*29,5±14,1*	22,4±4,9

Tabela 8: Resultados dos pré-processamentos para as ADs

de confiança. Na maioria dos casos também houve simplificações nos modelos das SVMs. Tem-se para esses casos a situação de manutenção das taxas de erros com modelos mais simples.

Com relação aos tempos de treinamento das SVMs, para todas as técnicas de pré-processamento eles foram iguais ou superiores aos verificados sobre os dados originais. Logo, não se observou benefícios em termos de reduções nos tempos de treinamento. No caso do número de SVs, foram observadas reduções em todos conjuntos pré-processados, que variaram entre 20% e 60%. Tem-se então maiores velocidades na etapa preditiva das SVMs.

Um fator interessante a ser observado é o de que, neste conjunto de dados, as técnicas DROP1 a DROP5 e DEL, que levaram a maiores reduções nos conjuntos de dados, tiveram bom desempenho, até mesmo melhor do que o das técnicas ENN, RENN e AkNN. Esse resultado é oposto ao verificado para os dados genômicos da seção anterior. Logo, os algoritmos DROP1 a DROP5 e DEL foram capazes de selecionar pequenos subconjuntos dos dados (genes) de grande valor informativo para as SVMs. Para o caso da técnica DEL, essa seleção levou inclusive à obtenção de menores taxas de erros das SVMs, com um modelo mais simples.

Para as ADs não se verificou o mesmo quadro. Nesse caso, grandes reduções nos conjuntos de dados foram prejudiciais. As técnicas DROP1 a DROP5 e DEL levaram então a quedas no poder de generalização das ADs. As técnicas ENN, RENN e AkNN, por outro lado, foram capazes de manter os resultados verificados sobre os dados originais, com reduções significativas nos tamanhos das ADs induzidas (cerca de 35% de decréscimo). Tem-se então simplificações nos modelos de ADs, que relete em ganhos de compreensibilidade.

Na comparação entre os desempenhos das SVMs e das ADs, tem-se pouca diferença no caso dos conjuntos original e ENN, RENN e AkNN. Esse fato pode ser confirmado

estatisticamente, a 95% de confiança. Portanto, na presença de dados de grande dimensão numéricos e com um número suficiente de amostras, os resultados das ADs e das SVMs se mostraram semelhantes.

Conclusões. Novamente, a aplicação de uma etapa de pré-processamento, agora sobre dados de expressão gênica, levou a benefícios em termos de simplificações nos modelos induzidos pelas técnicas de AM. Não houve neste caso um consenso quanto às melhores técnicas de pré-processamento para as SVMs e as ADs simultaneamente. De fato, alguns dos resultados observados para cada uma dessas técnicas de aprendizado podem ser considerados opostos.

No caso das SVMs em particular, alguns pré-processamentos que levaram a grandes reduções nos conjuntos de dados foram capazes de produzir SVMs com taxas de erro semelhantes ou mesmo menores às verificadas sobre os dados originais. Logo, a aplicação dessas técnicas de pré-processamento se mostrou criteriosa e as mesmas foram capazes de extrair pequenas frações de dados bastante informativas para as SVMs. Embora essa observação não tenha sido confirmada, essas técnicas podem ter sido capazes de selecionar SVs, os exemplos mais informativos contidos nos conjuntos de dados para as SVMs. Esse tipo de investigação constitui uma possibilidade de trabalho futuro na aplicação de pré-processamentos aos dados de expressão considerados. Porém, também neste caso, mais experimentos são necessários para uma confirmação das tendências observadas.

7 Conclusão

Este relatório apresentou a aplicação de técnicas de filtragem de ruídos a conjuntos de dados utilizados em tarefas preditivas em Bioinformática. Os resultados observados indicam que a realização dessa etapa de pré-processamento pode trazer benefícios em termos de simplificações nos classificadores induzidos para a solução do problema preditivo.

Por meio dos experimentos e resultados reportados na seção anterior, pode-se verificar que os maiores benefícios alcançados na aplicação de uma etapa de pré-processamento sobre dados genômicos se deram em simplificações nos modelos de AM induzidos. Em alguns casos, reduções nas taxas de erros também foram verificadas. Porém, contribuições ainda maiores podem ser obtidas com o emprego da etapa de pré-processamento em conjunto com a métrica VDM sobre conjuntos de dados mais atuais. Para os dados de expressão gênica, também foi possível obter simplificações nos modelos induzidos. Os resultados neste caso foram particularmente interessantes as SVMs, em que uma redução de aproximadamente 89,6% no tamanho do conjunto de dados possibilitou a obtenção de um modelo de melhor desempenho preditivo. Isso indica que os dados mais informativos

para essa técnica foram mantidos.

Deve-se destacar também que os resultados apresentados representam investigações iniciais. Mais experimentos, inclusive sobre outros problemas em Biologia Molecular, são necessários para uma avaliação mais ampla da adequabilidade e benefícios da aplicação desses procedimentos em Bioinformática.

Além disso, outras técnicas de pré-processamento, como as baseadas em agrupamentos dos dados (essas estratégias em geral também computam distâncias entre os dados), poderiam levar a resultados semelhantes ou mesmo melhores que os observados.

Como outras possibilidades de trabalhos futuros a partir dos experimentos apresentados, modificações podem ser realizadas nas medidas de distância empregadas nas técnicas de pré-processamento baseadas em distância. Na medida da distância entre seqüências de DNA, pode-se recorrer, por exemplo, a técnicas de alinhamento. O alinhamento entre seqüências é um problema amplamente estudado pela comunidade de Bioinformática, para o qual diversas soluções eficientes foram derivadas e analisadas (Setubal & Meidanis, 1997).

A comparação entre seqüências também pode ser realizada por meio de uma medida de quanto essas diferem, o que é capturado pelo conceito de distância entre seqüências (Setubal & Meidanis, 1997). Nessa abordagem, são designados custos a operações de edição elementares e procura-se a composição de operações com menor custo capaz de transformar uma seqüência na outra. É demonstrado em Setubal & Meidanis (1997) que essa medida também pode ser obtida por alinhamentos, podendo ser solucionada, portanto, de forma eficiente.

As métricas discutidas permitem a obtenção de uma estratégia de pré-processamento mais próxima da realidade biológica. Além disso, seu uso permite lidar com a questão da alta dimensionalidade dos dados biológicos. Técnicas de pré-processamento dos dados baseadas em agrupamentos, que utilizam medidas de distância na formação de aglomerados de dados semelhantes, também podem se beneficiar dessas modificações.

Como observação final, pode-se verificar pelas análises experimentais realizadas na Seção 6 uma maior robustez das SVMs diantes de dados ruidosos. Como a função Kernel nessas técnicas equivale a uma medida de distância entre dados (Cristianini & Shawe-Taylor, 2000), diversos trabalhos foram devotados a reformular essa função de forma a acomodar as características das seqüências biológicas (Leslie et al., 2002a,b). Em (Leslie et al., 2002b), por exemplo, uma função Kernel baseada em similaridades de cadeias foi desenvolvida. Esses autores propõem uma estratégia eficiente para computação dessa função e incorporam modelos de similaridade inexata. Esses modelos acomodam a ocorrência de substituições e espaços nas seqüências, aproximando a função Kernel a uma espécie de alinhamento. Essas modificações levaram a bons resultados na classificação de seqüências

proteicas e genômicas. Logo, no caso das SVMs, pode-se recorrer a reformulações na função Kernel em conjunto com o exame dos dados para os quais se tem dificuldades em maximizar a margem na obtenção de um mecanismo para identificação de ruídos.

Estudos mais detalhados a respeito de técnicas para a identificação de ruídos podem ser encontrados em: (Wilson & Martinez, 2000; Hodge & Austin, 2004; Otey et al., 2005; Escalante, 2005).

Referências

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In Bussche, J. V. & Vianu, V., editors, *8th International Conference on Database Theory (ICDT 2001) Lecture Notes in Computer Science*, volume 1973, p. 420–434, London, UK. Springer Verlag.

Baranauskas, J. A. & 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, São Carlos, Brasil. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_102.ps.zip`.

Barnett, V. & Lewis, T. (1994). *Outliers in Statistical Data*. John Wiley and Sons.

Batista, G. E. A. P. A., Carvalho, A. C. P. L. F., & Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In *Mexican International Conference on Artificial Intelligence (MICAI) Lecture Notes in Artificial Intelligence*, volume 1793, p. 315–325. Springer-Verlag.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? In Beeri, C. & Buneman, P., editors, *7th International Conference on Database Theory (ICDT'99) Lecture Notes in Computer Science*, volume 1540, p. 217–235, Jerusalem, Israel. Springer Verlag.

Blake, C. L. & Merz, C. J. (1998). UCI repository of machine learning databases. Disponível em: `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Braga, A., Carvalho, A. C. P. L. F., & Ludemir, T. (2000). *Redes Neurais Artificiais: Teoria e Aplicações*. Editora LTC.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Chapman and Hall.

Brodley, C. E. & Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167.

Brown, M., Grundy, W., Lin, D., Christianini, N., Sugnet, C., Jr, M., & Haussler, D. (1999). Support vector machine classification of microarray gene expression data. Relatório Técnico UCSC-CRL 99-09, Department of Computer Science, University California Santa Cruz, Santa Cruz, CA.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):1–43.

Casley, D. (1992). Primer on molecular biology. Relatório técnico, U. S. Department of Energy, Office of Health and Environmental Research.

Collobert, R. & Bengio, S. (2001). SVMTorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1:143–160.

Corney, D. P. A. (2002). *Intelligent Analysis of Small Data Sets for Food Design*. Tese de Doutorado, Department of Computer Science, University College, London.

Craven, M. W. & Shavlik, J. W. (1994). Machine learning approaches to gene recognition. *IEEE Expert*, 9(2):2–10.

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Escalante, H. J. (2005). A comparison of outlier detection algorithms for machine learning. In *14th Congress on Computing, CD proceedings, Mexico City*, p. 12p.

Escalante, H. J. (2006). Kernel methods for anomaly detection and noise elimination. In *Advances in Computer Science and engineering, International Conference on Computing, CORE 2006, Editorial IPN*, volume 19, p. 69–80.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Gamberger, D., Boskovic, R., & Lavrac, N. (1999). Experiments with noise filtering in a medical domain. In *Proceedings of the 16th International Conference on Machine Learning*, p. 143–151. Morgan Kaufmann, San Francisco, CA.

Gamberger, D. & Lavrac, N. (1997). Conditions for Occam's razor applicability and noise elimination. In *Proceedings of the European Conference on Machine Learning*, p. 108–123.

Hawkins, S., He, H., Williams, G., & Baxter, R. (2002). Outlier detection using replicator neural networks. In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, p. 170–180, Aix-en-Provence, France.

Haykin, S. (1999). *Neural Networks - A Compreensive Foundation*. Prentice-Hall, New Jersey, 2 edition.

He, Z. & Xu, S. (2002). Outlier detection integrating semantic knowledge. In *Proceedings of the 3th International Conference on Web-Age Information Management*, p. 126–131, Beijing, China.

He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650.

Hearst, M. A., Schölkopf, B., Dumais, S., Osuna, E., & Platt, J. (1998). Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

Hinneburg, A., Aggarwal, C. C., & Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In Abbadi, A. E., Brodie, M. L., Chakrovarthy, S., Dayal, U., Kamel, N., Schlageter, G., & Whang, K.-Y., editors, *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB 2000)*, p. 506–515, Cairo, Egypt.

Hodge, V. & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126.

Jiang, Y. & Zhou, Z. H. (2004). Editing training data for kNN classifiers with neural network ensemble. In *Proceedings of the IEEE Symposium on Neural Networks*, volume 3173 of *Lecture Notes in Computer Science*, p. 356–361.

Jin, W., Tung, A. K. H., & Han, J. (2001). Mining top-n local outliers in large databases. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 293–298, San Francisco, California.

Knorr, E. M. & Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th Conference on Very Large Data Bases (VLDB)*, p. 392–403, New York, USA.

Knorr, E. M., Ng, R. T., & Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal: Very Large Data Bases*, 8(3-4):237–253.

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Perez, A., & Robles, V. (2006). Machine learning in bioinformatics. *Brief Bioinformatics*, 7:86–112.

Lavrac, N. & Gamberger, D. (2001). Saturation filtering for noise and outlier detection. In Scheffer, T. & Wrobel, S., editors, *Proceedings of the Workshop on Active Learning, Database Sampling, Experimental Design: Views on Instance Selection: Working Notes*,

p. 1–4, 12th European Conference on Machine Learning / 5th European Conference on Principles and Practice of Knowledge Discovery in Databases, Freiburg.

Leslie, C., Eskin, E., & Noble, W. (2002a). The spectrum kernel: a string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, p. 564–575.

Leslie, C., Eskin, E., Weston, J., & Noble, W. (2002b). Mismatch string kernels for SVM protein classification. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press. To appear.

Lewin, B. (2001). *Genes VII - versão traduzida*. Editora Artmed.

Lewis, R. (2001). *Human Genetics - Concepts and Applications*. Mc Graw Hill, 4 edition.

Lorena, A. C. (2006). *Investigação de Estratégias para a Geração de Máquinas de Vetores de Suporte Multiclasses*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Disponível em : `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-26052006-111406/`.

Lorena, A. C., Batista, G. E. A. P. A., Carvalho, A. C. P. L. F., & Monard, M. C. (2002a). The influence of noisy patterns in the performance of learning methods in the splice junction recognition problem. In *IEEE Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN 2002)*, p. 31–36. IEEE Computer Society Press.

Lorena, A. C., Batista, G. E. A. P. A., Carvalho, A. C. P. L. F., & Monard, M. C. (2002b). Splice junction recognition using machine learning techniques. In Bazzan, A. L. C. & de Carvalho, A. C. P. L. F., editors, *Proceedings of the 1st Brazilian Workshop on Bioinformatics (WOB)*, p. 32–39.

Lorena, A. C. & Carvalho, A. C. P. L. F. (2003a). Evaluation of noise reduction techniques in the splice junction recognition problem. 1st International Conference on Computational Biology and Bioinformatics (ICOBICOBI), May 14-16, Ribeirão Preto, Brasil, `http://www.vision.ime.usp.br/~cesar/programa/pdf/159.pdf`.

Lorena, A. C. & Carvalho, A. C. P. L. F. (2003b). Introdução às máquinas de vetores suporte (*Support Vector Machines*). Relatório Técnico 192, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_192.ps.zip`.

Lorena, A. C. & Carvalho, A. C. P. L. F. (2003c). Revisão de técnicas para geração de classificadores de margens largas multiclasses. Relatório Técnico 221, Instituto de Ciên-

cias Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_221.ps`.

Lorena, A. C. & Carvalho, A. C. P. L. F. (2003d). Utilização de técnicas inteligentes em bioinformática. Relatório Técnico 219, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_219.ps`.

Lorena, A. C. & Carvalho, A. C. P. L. F. (2004). Evaluation of noise reduction techniques in the splice junction recognition problem. *Genetics and Molecular Biology*, 27(4):665–672.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Müller, K. R., Mika, S., Rätsch, G., Tsuda, K., & Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201.

Monard, M. C. & Baranauskas, J. A. (2003). Conceitos de aprendizado de máquina. In Rezende, S. O., editor, *Sistemas Inteligentes - Fundamentos e Aplicações*, p. 89–114. Editora Manole.

Nadeau, C. & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, 52(3):239–281.

Noordewier, O., G. G. Towell, G. G., & Shavlik, J. W. (1991). Training knowledge-based neural networks to recognize genes in DNA sequences. In *Advances in Neural Information Processing Systems*, volume 3, San Mateo, CA. Morgan Kaufmann Publishers.

Nuts, R. & Rousseeuw, P. (1996). Computing depth countours of bivariate point clouds. *Journal of Computational Statistics and Data Analysis*, 23:153–168.

Otey, M., Parthasarathy, S., & Ghoting, A. (2005). An empirical comparison of outlier detection algorithms. In *Proceedings of the ACM SIGKDD Workshop on Data Mining Methods for Anomaly Detection*, p. 45–52.

Pedersen, A. G. & Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB'97)*, p. 226–233.

Pollastro, P. & Rampone, S. (2002). HS3D, a dataset of homo sapiens splice regions, and its extraction procedure from a major public database. *International Journal of Modern Physics C*, 13(8):1105–1117.

Quilan, J. R. (1988). *C4.5 Programs for Machine Learning.* Morgan Kaufmann, CA.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data and Symposium on Principles of Database Systems*, p. 427–438, Dallas, Texas, US.

Rissanen, J. (1978). Modeling by the shortest data description. *Automatica*, 14:465–471.

Rulequest-Research (2000). Rulequest research - data mining tools. Disponível em: `http://www.rulequest.com`.

Setubal, J. & Meidanis, J. (1997). *Introduction to Computational Molecular Biology.* PWS Publishing Company.

Silva, F. H. (2001). Apostila - curso de biologia molecular. INBIO - I Escola Brasileira de Inteligência Artificial e Bioinformática, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Disponível em: `http://www.icmc.usp.br/~inbio/material/Apostila_Inbio_Biomol.pdf`.

Souto, M. C. P., Lorena, A. C., Delbem, A. C. B., & Carvalho, A. C. P. L. F. (2003). *Técnicas de Aprendizado de Máquina para problemas de Biologia Molecular*, p. 103–152. Minicursos de Inteligência Artificial, Jornada de Atualização Científica em Inteligência Artificial, XXIII Congresso da Sociedade Brasileira de Computação.

Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228.

Stormo, G. D. (2000). DNA biding sites: representation and discovery. *Bioinformatics*, 16(1):16–23.

Tomek, I. (1976a). An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):448–452.

Tomek, I. (1976b). Two modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, 6(11):769–772.

Towell, G. G. (1991). *Symbolic Knowledge and Neural Networks: insertion, refinement, and extraction.* Tese de Doutorado, University of Winsconsin, Madison.

Towell, G. G. & Shavlik, J. W. (1992). Interpretation of artificial neural networks: mapping knowledge-based neural networks into rules. In *Advances in Neural Information Processing Systems*, volume 4, p. 977–984, San Mateo, CA. Morgan Kaufmann.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag, 2 edition.

Verbaeten, S. (2002). Identifying mislabeled training examples in ILP classification problems. In Wiering, M., editor, *Proceedings of 12th Belgian-Dutch Conference on Machine Learning*, p. 1–8.

Vouet, D., Vouet, J. G., & Pratt, C. W. (2002). *Fundamentos de Bioquímica - versão traduzida.* Editora Artmed.

Wilson, D. L. (1972). Asymtoptic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, 2(3):408–421.

Wilson, D. R. & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286.