X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

IDENTIFICAÇÃO DE RUÍDO EM DADOS DE EXPRESSÃO GÊNICA

Giampaolo L. Libralon
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) - Campus São Carlos, São Carlos, SP, Brasil
glibralon@cefetsp.br

Ana Carolina Lorena, André C. P. L. F. de Carvalho
Centro de Matemática, Computação e Cognição - CMCC/UFABC, Santo André, SP, Brasil
Instituto de Ciências Matemáticas e de Computação - ICMC/USP, São Carlos, SP, Brasil
ana.lorena@ufabc.edu.br, andre@icmc.usp.br

Resumo – Ruído pode ser definido como um exemplo em um conjunto de dados que aparentemente é inconsistente com o restante dos dados existentes, pois não segue o mesmo padrão dos demais. Ruídos em conjuntos de dados podem reduzir o desempenho das técnicas de Aprendizado de Máquina (AM) empregadas e aumentar o tempo de construção da hipótese induzida, assim como sua complexidade. Algoritmos para a detecção e remoção de ruídos podem aumentar a confiabilidade de conjuntos de dados ruidosos. Os dados em Bioinformática são conhecidos por apresentarem uma grande quantidade de ruídos. Esses ruídos são gerados por diversas razões, como erros humanos e pela imprecisão inerente a técnicas de coleta de dados. Neste trabalho, diferentes algoritmos para detecção de ruído são investigados para minimizar a interferência dos ruídos em conjuntos de dados que tratam problemas da área de Bioinformática.

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

Abstract – Noise can be defined as an example which seems to be inconsistent with the remaining ones in a data set. The presence of noise in data sets can decrease the performance of Machine Learning (ML) techniques in the problem analysis and also increase the time taken to build the induced hypothesis and its complexity. Algorithms to detect and remove noise may increase trustworthiness of noisy data sets. Bioinformatics data are known to present a large amount of noise. These noise data are generated by several reasons, like human mistakes and the imprecision inherent to techniques for data collection. In this work, different techniques for noise detection are investigated to minimize the interference of noisy data in training data sets of Bioinformatics problems.

Keywords – Machine Learning, Noise Detection, Gene Expression Problems, Ensembles.

1. INTRODUÇÃO

Caracterizado por não apresentar os mesmos padrões que a grande maioria dos demais exemplos de um conjunto de dados, um ruído pode ser um exemplo que assume um rótulo diferente dos exemplos mais semelhantes a ele, ou um exemplo cujo valor para pelo menos um atributo é muito superior ou inferior aos outros valores observados para o mesmo atributo. Vale ressalvar que exemplos considerados ruído podem, em alguns casos, representar dados corretos, que seguem padrões diferenciados dos demais exemplos presentes no conjunto de dados [1].

Dados são geralmente coletados por meio de medições realizadas em um domínio de interesse, associando o valor de uma variável com uma determinada propriedade existente no ambiente ou problema em estudo. As relações existentes entre os diversos elementos do problema em análise são representadas por relacionamentos numéricos entre as variáveis existentes no conjunto de dados. Sendo assim, erros de medições, dados incompletos, amostras distorcidas, falhas humanas ou dos equipamentos utilizados, dentre muitos outros fatores, contribuem para a contaminação dos dados. E isto é particularmente verdadeiro para conjuntos que apresentam dados com elevada dimensionalidade. A presença de ruído em um conjunto de dados de treinamento utilizado por um algoritmo de Aprendizado de Máquina (AM) pode reduzir a acurácia de classificação do modelo induzido, além de aumentar sua complexidade e seu tempo de treinamento [2].

Diversos algoritmos de AM induzem classificadores capazes de predizer a classe de novos dados que não foram apresentados durante o processo de aprendizado. O desempenho de um classificador é diretamente influenciado pela qualidade do conjunto de dados apresentado durante seu treinamento. Embora grande parte dos algoritmos de AM apresente uma robustez diante de dados ruidosos, a existência de ruídos no conjunto de dados interfere na eficácia do modelo obtido. Uma hipótese induzida a partir de um conjunto de dados livre de ruídos possui maior probabilidade de apresentar complexidade menor, além de ser mais acurada na classificação de dados desconhecidos. Por esse motivo, a remoção de exemplos ruidosos dos conjuntos de treinamento pode aumentar a confiabilidade e a qualidade dos dados.

Neste trabalho, são avaliados e comparados diferentes algoritmos para a detecção de ruído, assim como um conjunto de combinações destes. As combinações de algoritmos de detecção de ruído desenvolvidas têm como objetivo melhorar o desempenho final apresentado pelo classificador. Para facilitar a compreensão dos experimentos realizados, este artigo está organizado

da seguinte forma: na Seção 2 é feita uma breve discussão sobre o problema de detecção de ruídos. A Seção 3 apresenta os algoritmos investigados. Na Seção 4 são apresentados os conjuntos de dados utilizados nos experimentos. A metodologia empregada, assim como os experimentos e resultados obtidos são apresentados na Seção 5 e, a Seção 6, apresenta as principais conclusões obtidas.

2. DETECÇÃO DE RUÍDO

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

A qualidade de um conjunto de dados pode ser determinada basicamente por fatores internos e externos. O fator interno indica se os atributos e as classes foram bem selecionados e definidos para caracterizar esse conjunto, ao passo que o externo se refere a erros adicionados aos atributos e classes, seja de modo artificial e voluntário ou àqueles introduzidos de maneira acidental. Nesse sentido, em AM é possível caracterizar ruídos em dois grupos: ruídos presentes nos atributos e nas classes. O primeiro é representado por erros que são introduzidos nos valores dos atributos. O segundo, por erros nas classes, seja por um mesmo dado com mais de uma classificação ou por dados que são rotulados por classes incorretas [2].

Existem diversas técnicas de pré-processamento que podem ser aplicadas na detecção e remoção de ruídos. Neste trabalho, a identificação de ruídos é realizada por meio de algoritmos baseados em distância [3]. Algoritmos baseados em distância se destacam pela simplicidade de implementação e por não fazerem suposições, a priori, sobre o modelo de distribuição dos dados investigados. Um de seus problemas está no fato de consumir elevada memória e tempo na computação das distâncias entre todos os exemplos analisados. Nesse sentido, a complexidade computacional é diretamente proporcional à dimensionalidade dos dados e ao número de registros existente [4].

3. ALGORITMOS INVESTIGADOS

Algoritmos baseados em distância como, por exemplo, aqueles desenvolvidos com base no algoritmo k-*nearest neighbor* (k-NN), baseiam-se em medidas de similaridade existentes entre os dados analisados. O algoritmo k-NN é o representante mais simples da classe de técnicas de AM supervisionadas baseadas em instâncias [5]. Este algoritmo determina a classificação de um exemplo de acordo com a classe da maioria de seus k vizinhos mais próximos.

Os algoritmos baseados em distância investigados neste trabalho são *AllkNN*, *Edited Nearest Neighbor* (ENN), *Repeated ENN* (RENN), *Decremental Reduction Optimization Procedures* (DROPs) 1 a 5 e o algoritmo *Decremental Encoding Length* (DEL). Todos os algoritmos selecionados possuem o algoritmo k-NN [5] como base, e utilizam medidas de similaridade na identificação de possíveis ruídos.

Para os algoritmos investigados, diferentes valores do parâmetro k, que determina o número de vizinhos mais próximos no k-NN, foram avaliados, seguindo uma progressão geométrica que contém o valor 3 (valor *default* do código utilizado), nesse sentido, os valores testados para o parâmetro k foram 1, 3 e 9. Outra consideração importante diz respeito à definição da medida de similaridade a ser utilizada na computação das distâncias entre os dados. É demonstrado que, para espaços com grandes dimensões, várias métricas, como a Euclidiana, são pouco significativas. Isto se deve ao fato de que os dados tendem a ser esparsos [6]. Como os dados avaliados neste trabalho possuem alta dimensionalidade, a medida de distância utilizada é a *Heterogeneous Value Difference Metric* (HVDM), indicada por [7] como adequada para utilização em conjuntos de dados com as características descritas.

Para a descrição dos algoritmos baseados em distância, considere o seguinte: Seja *T* o conjunto de treinamento original e *S* o subconjunto de *T*, obtido com a aplicação de um dos algoritmos de eliminação de ruídos baseados em distância. Suponha que *T* contém *n* dados $\mathbf{x}_1$**, ...,** $\mathbf{x}_n$. Cada dado **x** de *T* (e também de *S*) possui k vizinhos mais próximos.

[8] propôs o algoritmo ENN. Para a utilização desse algoritmo, supõe-se, inicialmente, *S* = *T*. Um exemplo é removido do conjunto de dados *S* se sua classe não é igual à da maioria de seus k vizinhos mais próximos. Esse algoritmo elimina ruídos e dados próximos à borda de decisão, utilizada na separação dos dados em classes. O algoritmo RENN é uma variação do algoritmo ENN, em que o algoritmo ENN é aplicado repetidamente ao conjunto de dados até que todos os exemplos remanescentes tenham a maioria de seus vizinhos pertencentes à mesma classe. Isso permite que as classes sejam mais bem definidas pois define a fronteira de decisão, permitindo uma melhor generalização.

O algoritmo AllkNN foi proposto por [9] e trata-se de outra extensão do algoritmo ENN. Esse algoritmo procede da seguinte maneira: para *i = (1, . . . , k)*, marcar como "incorreto" qualquer exemplo classificado erroneamente por seus *i* vizinhos mais próximos. Depois da análise de todos os exemplos existentes, remover os exemplos previamente marcados.

Os algoritmos DEL e DROPs 1 a 5 foram propostos por [10]. Estes são algoritmos para eliminação de ruídos baseados em distância que visam apresentar tolerância aos ruídos, elevada taxa de acerto na generalização, não sensibilidade à ordem em que os dados são apresentados e alta taxa de eliminação de ruídos com consequente redução do conjunto de dados original. Esses algoritmos apresentam uma etapa de identificação e remoção de ruídos e uma etapa de detecção e remoção de dados redundantes presentes no conjunto de dados analisado. Nesse sentido, o conjunto de dados resultante será constituído apenas de exemplos que não sejam considerados redundantes ou ruído pelo algoritmo avaliado.

4. CONJUNTOS DE DADOS

Os conjuntos de dados selecionados representam problemas de análise de expressão gênica e se caracterizam por apresentar elevada dimensionalidade e a presença de ruído, fato que dificulta a classificação de seus dados por diversos algoritmos de AM. A seguir é apresentada uma breve descrição de cada um dos conjuntos utilizados.

ExpGen: Apresenta medidas do nível de expressão para classificação funcional de genes. Esse conjunto contém dados sobre 79 experimentos distintos, que definem a função dos genes no meio celular. O número de genes utilizados foi reduzido de 2467 para 207 por meio de seleção dos atributos mais significativos [11]. Cinco classes distintas estão presentes nesse conjunto.

Leukemia: Referenciado na literatura como St. Jude Leukemia [12], possui dados de expressão gênica obtidos a partir de imagens de *microarrays* que representam 6 subtipos diagnosticáveis de leucemia linfóide aguda pediátrica e um grupo com amostras que não se enquadram nos grupos anteriores. O conjunto de dados original contém 12558 atributos (genes), porém foi, neste trabalho, pré-processado e redimensionado como descrito em [12].

Lung: Apresenta dados sobre amostras de câncer de pulmão. Os três diferentes tipos de câncer analisados são os adenocarcinomas, os carcinomas de células escamosas (*squamous cell carcinomas*) e os carcinóides, além de uma quarta classe que indica não ser possível a classificação do câncer detectado em nenhum dos três tipos anteriores. A versão utilizada para esse conjunto foi a mesma utilizada por [13], gentilmente cedida por esses autores.

Colon16: Consiste no conjunto de dados Colon, que apresenta informações sobre pacientes com câncer de colo [14], após ter sua dimensionalidade reduzida de acordo com procedimento apresentado em [15] para seleção dos genes que melhor representam o domínio do problema, de modo a avaliar se a redução de atributos pode auxiliar no processo de identificação de ruídos. Do total, foram selecionados os 16 genes mais significativos para a discriminação de pacientes com e sem câncer de colo (classes).

Golub64: Consiste no conjunto de dados Golub, que possui informações sobre dados de expressão gênica de amostras de pacientes com leucemia aguda [16], após ter sua dimensionalidade reduzida de acordo com procedimento apresentado em [15], em que os 64 genes mais significativos foram selecionados. As classes de leucemia aguda representadas são leucemia linfóide aguda (ALL) e leucemia mielóide aguda (AML). As principais características desses conjuntos estão resumidas na Tabela 1.

Tabela 1: Descrição dos conjuntos de dados selecionados.

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

5. EXPERIMENTOS E RESULTADOS

Os experimentos realizados foram divididos em duas etapas: pré-processamento e classificação. Na primeira, de pré-processamento, ocorre a detecção do ruído por meio dos algoritmos avaliados, de modo a reduzir a quantidade destes nos dados de treinamento. A segunda etapa, de classificação, prevê a indução de classificadores para os novos conjuntos de dados, livres de ruído, e permite verificar se houve melhora no desempenho dos classificadores.

Na segunda etapa, os classificadores avaliados são as Máquinas de Vetores de Suporte (SVMs), o algoritmo C4.5 e o algoritmo RIPPER. As SVMs são baseadas na teoria de aprendizado estatístico e procuram dividir dados em classes por meio de um hiperplano, com a maior margem de separação possível. Este hiperplano pode, então, ser utilizado na classificação de novos dados [17].

O algoritmo C4.5 é um algoritmo de aprendizado simbólico que induz árvores de decisão (ADs) a partir de um conjunto de dados de treinamento, por meio de um processo de aprendizado [5]. A complexidade de uma AD é avaliada de acordo com seu tamanho, quanto maior o tamanho da árvore, maior a complexidade. Algumas vantagens das ADs são a compreensibilidade das regras de classificação produzidas, a facilidade de manutenção, flexibilidade e velocidade de treinamento. Em contrapartida, são pouco robustas a exemplos de elevada dimensionalidade (com grande quantidade de atributos).

O algoritmo RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*) [18] é um algoritmo de indução de regras que foi projetado para obter pequenas taxas de erro de classificação em conjuntos de dados que apresentam ruído e possuem elevada dimensionalidade. Algoritmos de indução de regras são mais flexíveis e incrementais que algoritmos geradores de ADs, uma vez que, com a inclusão de novos dados, novas regras podem ser adicionadas ou modificadas sem afetar as já existentes no conjunto de regras.

Para obter as estimativas do erro de predição, o método k*-fold cross validation* foi utilizado em todos os experimentos, com o número de partições (valor de k), definido como dez. As duas etapas de experimentos realizadas resultam no porcentual de erro da predição de cada classificador para cada conjunto de dados antes e após a eliminação de ruídos. Com isso, é possível verificar se houve melhora no desempenho dos classificadores com a eliminação dos ruídos. Para avaliar estatisticamente os resultados

obtidos nos experimentos, foram aplicados os testes estatísticos de Friedman e de comparações múltiplas de Dunn [19], com 95% de confiança.

Para treinamento das SVMs, utilizou-se o software SVMTorch II [20]. Para o treinamento das ADs, foi utilizado o algoritmo C4.5 [21], o treinamento do algoritmo RIPPER foi realizado pelo simulador Weka, desenvolvido pela Universidade de Waikato [22]. Em todos os experimentos realizados, os valores dos parâmetros desses algoritmos foram definidos como os valores *default* de cada uma das ferramentas utilizadas.

Inicialmente, os algoritmos baseados em distância ENN, RENN, AllkNN, DROPs 1 a 5 e DEL, são aplicados aos conjuntos originais, produzindo conjuntos de dados pré-processados, ou seja, sem os dados considerados ruído por cada um dos algoritmos utilizados. Em seguida, os conjuntos livres de ruído, resultantes dos pré-processamentos, assim como os conjuntos de dados originais, são submetidos aos classificadores SVM, C4.5 e RIPPER, de modo a analisar os resultados obtidos com a etapa de eliminação de ruídos.

Na Tabela 2 é apresentada a porcentagem de exemplos dos conjuntos originais que foram considerados ruído pelos algoritmos investigados. Pode-se observar, de maneira geral, que os algoritmos baseados em distância consideraram como ruído uma quantidade significativamente elevada dos dados existentes nos conjuntos avaliados.

Tabela 2: Percentual de ruídos identificados pelos algoritmos avaliados.

	k	Leukemia	Lung	ExpGen	Golub64	Colon16
AllkNN	1	25.75	23.08	26.23	22.08	36.77
	3	27.61	24.00	27.53	22.91	39.67
	9	30.09	24.57	29.08	23.19	40.32
ENN	1	25.17	22.99	25.56	20.97	34.19
	3	25.29	22.59	25.65	21.39	31.93
	9	27.77	23.09	26.76	20.97	32.25
RENN	1	25.69	23.20	25.84	21.12	35.16
	3	26.60	24.06	25.94	21.80	33.22
	9	28.90	23.80	27.92	20.97	33.70
DROP1	1	94.10	96.45	94.97	95.84	92.74
	3	92.60	94.36	92.99	92.91	91.13
	9	87.46	88.93	88.02	84.72	81.77
DROP2	1	91.31	95.78	93.28	94.72	85.00
	3	89.97	93.50	90.91	92.23	85.80
	9	84.06	88.37	85.84	84.72	78.06
DROP3	1	93.40	95.78	94.10	94.82	95.00
	3	91.53	93.75	92.41	92.91	91.13
	9	87.25	87.76	87.00	84.58	81.93
DROP4	1	92.63	95.58	93.91	94.82	94.35
	3	90.00	93.35	91.50	92.08	90.48
	9	84.55	88.27	86.23	84.58	81.13
DROP5	1	91.80	94.82	94.15	95.84	95.32
	3	89.81	93.09	92.12	92.50	91.45
	9	84.71	87.71	88.30	84.58	81.61
DEL	1	90.34	96.45	91.12	98.61	86.45
	3	92.41	95.58	90.96	95.84	94.84
	9	86.55	90.35	87.73	87.50	84.68

As taxas de erro e o desvio padrão (em porcentagem) obtidos nos experimentos realizados com os algoritmos investigados aplicados individualmente para detecção de ruído são apresentados nas tabelas 3 a 5, respectivamente, para os classificadores SVM, C4.5 e RIPPER. Para facilitar a leitura e a comparação dos resultados, os conjuntos pré-processados que apresentaram desempenho superior ou semelhante ao obtido para o conjunto original são apresentados em **negrito**. Os desvios-padrão são apresentados entre parênteses.

Os resultados da Tabela 3 indicam a obtenção de pequenas melhorias nos resultados obtidos para os conjuntos de dados Leukemia e ExpGen. É possível observar que os algoritmos baseados em distância avaliados, mesmo com a remoção de exemplos considerados ruidosos, não melhoraram o desempenho das SVMs. Pelo contrário, para a maioria dos conjuntos de dados avaliados, o aumento verificado na taxa de erro foi expressivo.

Na Tabela 4 podem ser observados os resultados obtidos para o classificador C4.5. É possível notar que os resultados obtidos foram satisfatórios, uma vez que o desempenho apresentado por esse classificador após a etapa de eliminação de ruído realizada pelos diferentes algoritmos investigados foi, em sua maioria, semelhante ou superior àquele apresentado para os conjuntos originais. Somente com os conjuntos Leukemia e ExpGen a etapa de eliminação de ruídos não reduziu as taxas de erro de classificação. Para os demais conjuntos, a melhora no desempenho aconteceu para os diferentes algoritmos avaliados.

A melhoria dos resultados obtidos para o classificador RIPPER, apresentados na Tabela 5, foram similares aos obtidos pelo classificador C4.5. Para os dois casos, considerando todos os conjuntos de dados avaliados, a etapa de eliminação de ruídos propiciou uma melhora no desempenho do classificador. A única exceção pode ser observada para o conjunto de dados Leukemia, em que a etapa de eliminação de ruídos não reduziu as taxas de erro obtidas.

Após a análise individual dos algoritmos para detecção de ruído, foram desenvolvidas combinações com o objetivo de obter taxas de erro inferiores às obtidas individualmente pelos algoritmos investigados. Todas as combinações desenvolvidas seguiram a metodologia de votação por maioria, em que cada dado é considerado ruído se, para n algoritmos avaliados, pelo menos m deles, em que $m > n/2$, classificam o dado como provável ruído.

Tabela 3: Erro médio dos algoritmos avaliados para o classificador SVM.

	k	Leukemia	Lung	ExpGen	Golub64	Colon16
Original		7,36(5,36)	4,53(4,32)	6,53(4,78)	1,25(3,95)	11,19(7,78)
AllkNN	1	**7,95(3,91)**	29,42(3,74)	7,69(5,13)	34,82(12,17)	35,47(18,63)
	3	**7,95(3,91)**	29,42(3,74)	8,19(3,90)	34,82(12,17)	35,47(18,63)
	9	**7,94(2,96)**	29,42(3,74)	8,69(3,76)	34,82(12,17)	35,47(18,63)
ENN	1	**7,34(3,65)**	29,42(3,74)	8,67(4,34)	34,82(12,17)	35,47(18,63)
	3	8,27(3,61)	29,42(3,74)	**7,71(3,32)**	34,82(12,17)	35,47(18,63)
	9	11,00(4,83)	29,42(3,74)	9,64(5,48)	34,82(12,17)	35,47(18,63)
RENN	1	**7,65(3,93)**	29,42(3,74)	8,67(4,34)	34,82(12,17)	35,47(18,63)
	3	11,01(5,08)	29,42(3,74)	**7,71(3,32)**	34,82(12,17)	35,47(18,63)
	9	13,43(6,26)	29,42(3,74)	9,17(4,16)	34,82(12,17)	35,47(18,63)
DROP1	1	13,47(3,71)	77,37(28,34)	8,64(7,02)	65,18(12,17)	43,57(23,14)
	3	12,52(9,31)	80,37(26,76)	8,21(5,58)	60,89(16,50)	41,19(22,25)
	9	11,65(7,34)	76,55(26,55)	9,64(5,00)	65,18(12,17)	46,90(23,89)
DROP2	1	11,29(3,79)	77,87(25,77)	7,19(5,12)	48,75(20,06)	64,52(18,63)
	3	14,09(6,50)	78,52(26,95)	8,62(6,67)	56,96(18,71)	63,09(19,77)
	9	**8,87(5,06)**	69,26(29,61)	9,21(6,25)	65,18(12,17)	61,19(21,03)
DROP3	1	13,44(5,76)	79,08(24,65)	8,19(5,13)	50,18(20,10)	54,52(23,63)
	3	11,31(3,23)	78,05(26,77)	8,17(5,96)	60,89(16,50)	64,52(18,63)
	9	14,35(4,71)	61,73(35,09)	8,69(4,92)	65,18(12,17)	56,90(22,99)
DROP4	1	14,05(5,63)	72,58(28,28)	7,69(5,69)	50,18(20,10)	46,43(23,81)
	3	13,45(7,31)	71,21(30,51)	9,59(6,79)	59,46(17,45)	64,52(18,63)
	9	**8,89(5,32)**	81,05(28,04)	8,19(5,50)	65,18(12,17)	60,24(21,56)
DROP5	1	11,04(5,54)	42,26(26,67)	7,67(6,40)	65,18(12,17)	54,52(23,63)
	3	12,84(7,39)	36,26(20,85)	**6,71(5,10)**	47,32(19,90)	50,24(24,11)
	9	**8,26(2,50)**	48,71(30,89)	8,19(5,03)	63,75(13,93)	39,76(21,56)
DEL	1	13,42(5,87)	83,87(21,27)	**6,24(6,78)**	65,18(12,17)	64,52(18,63)
	3	14,34(6,02)	73,26(30,71)	**5,74(4,37)**	65,18(12,17)	57,86(22,64)
	9	9,16(4,96)	52,52(31,29)	8,17(5,04)	65,18(12,17)	53,57(23,81)

Tabela 4: Erro médio dos algoritmos avaliados para o classificador C4.5.

	k	Leukemia	Lung	ExpGen	Golub64	Colon16
Original		16,01(6,19)	7,50(5,81)	7,12(3,87)	12,20(8,09)	34,04(28,66)
AllkNN	1	18,95(5,07)	**6,06(5,15)**	**8,16(7,46)**	15,19(10,15)	**21,43(15,90)**
	3	**17,47(5,51)**	**7,56(4,21)**	**8,65(4,91)**	15,19(10,15)	**19,77(15,38)**
	9	18,70(5,58)	**7,56(4,21)**	12,52(5,07)	15,19(10,15)	**19,77(15,38)**
ENN	1	**17,43(4,75)**	**5,56(4,97)**	9,18(6,10)	15,19(10,15)	**18,10(16,61)**
	3	**16,83(7,05)**	**8,08(4,83)**	12,03(7,47)	13,94(9,56)	**18,10(16,61)**
	9	**17,43(6,19)**	8,08(6,32)	12,49(7,07)	15,19(10,15)	**18,11(14,63)**
RENN	1	**17,74(4,87)**	**6,06(4,58)**	9,66(6,32)	**12,33(9,98)**	**19,77(15,38)**
	3	**16,83(6,72)**	**7,58(4,87)**	12,50(7,10)	**9,65(9,50)**	**21,43(15,90)**
	9	**16,81(4,98)**	9,13(4,59)	12,49(7,07)	15,19(10,15)	**19,77(18,98)**
DROP1	1	52,70(12,04)	62,85(15,54)	37,46(24,40)	14,12(11,69)	**24,77(21,27)**
	3	33,62(10,81)	56,00(17,17)	31,45(6,35)	16,62(12,89)	**20,72(18,34)**
	9	33,37(7,62)	50,76(8,17)	19,64(10,31)	**6,97(7,36)**	**25,96(15,07)**
DROP2	1	37,36(5,37)	50,13(19,90)	26,88(21,09)	18,41(13,64)	42,62(16,94)
	3	37,64(5,10)	47,65(20,36)	42,20(22,49)	**11,08(8,95)**	**35,46(20,90)**
	9	29,93(7,43)	44,58(18,86)	27,40(14,36)	**8,22(11,43)**	**34,77(23,06)**
DROP3	1	42,02(10,87)	62,39(17,94)	36,14(20,67)	12,69(10,54)	47,16(29,16)
	3	32,45(7,17)	41,42(20,52)	35,74(14,42)	13,40(15,96)	39,52(28,69)
	9	27,83(6,07)	34,97(5,68)	21,07(16,43)	12,51(13,82)	**33,11(25,04)**
DROP4	1	36,75(7,12)	60,39(16,84)	34,64(22,06)	12,69(10,54)	**35,25(23,92)**
	3	32,16(9,79)	47,26(15,61)	45,06(14,69)	16,26(15,85)	42,85(27,67)
	9	31,51(7,92)	44,65(16,66)	22,97(17,41)	12,51(13,82)	**34,53(21,54)**
DROP5	1	42,51(5,58)	43,35(21,18)	38,52(21,15)	**9,65(11,18)**	**36,91(21,27)**
	3	36,17(11,32)	38,52(12,82)	44,83(23,98)	19,30(9,41)	**35,72(25,71)**
	9	31,52(9,16)	31,84(12,23)	25,53(15,22)	16,62(12,89)	**22,39(16,95)**
DEL	1	44,95(10,74)	49,14(16,06)	69,19(23,59)	34,84(12,17)	36,45(25,21)
	3	35,17(11,23)	51,72(13,81)	40,20(21,03)	**12,51(10,01)**	36,19(29,56)
	9	30,94(5,21)	40,65(12,13)	29,87(19,44)	**6,79(9,29)**	34,95(12,63)

Tabela 5: : Erro médio dos algoritmos avaliados para o classificador RIPPER.

	k	Leukemia	Lung	ExpGen	Golub64	Colon16
Original		17,18(4,96)	12,11(5,79)	14,28(6,85)	7,67(8,87)	27,38(15,06)
AllkNN	1	21,46(8,97)	**7,55(5,37)**	**13,48(5,93)**	**9,64(9,49)**	**20,95(17,92)**
	3	19,96(9,72)	**9,55(6,81)**	**12,98(7,39)**	**6,79(9,84)**	**21,43(19,41)**
	9	22,36(10,66)	**9,55(6,81)**	15,36(8,82)	**8,21(9,79)**	**24,52(18,27)**
ENN	1	20,52(7,87)	**6,55(4,09)**	**11,52(6,74)**	11,07(11,19)	**23,10(21,18)**
	3	20,52(4,48)	**8,55(6,23)**	**13,45(8,26)**	12,68(10,53)	**23,10(18,03)**
	9	21,42(7,81)	**9,05(6,96)**	15,36(9,90)	11,07(11,19)	**26,19(19,86)**
RENN	1	22,39(8,10)	**6,05(3,93)**	**13,43(7,95)**	11,07(11,19)	**14,76(19,95)**
	3	19,64(7,44)	**10,58(6,78)**	**12,98(8,04)**	11,25(11,23)	**19,76(19,54)**
	9	22,65(9,92)	**12,11(7,08)**	**13,93(8,15)**	9,82(11,70)	**24,52(19,89)**
DROP1	1	58,14(7,44)	29,42(3,74)	61,86(21,11)	34,82(12,18)	41,43(30,91)
	3	53,89(15,75)	72,71(19,04)	61,50(20,92)	21,96(19,81)	39,52(35,09)
	9	43,40(8,03)	50,16(11,21)	38,60(16,99)	11,07(10,71)	28,57(20,57)
DROP2	1	42,81(8,16)	65,53(27,72)	55,29(16,63)	33,39(18,02)	47,62(17,60)
	3	45,24(11,58)	50,85(23,40)	48,48(13,10)	19,11(15,84)	36,90(22,95)
	9	32,08(7,05)	56,76(15,00)	43,40(9,49)	20,71(11,76)	39,29(20,12)
DROP3	1	48,61(10,40)	74,58(25,51)	64,31(19,59)	34,82(15,46)	64,52(18,63)
	3	43,72(10,07)	50,11(25,73)	45,02(14,99)	24,64(31,52)	56,19(25,13)
	9	30,53(5,31)	38,45(15,93)	43,02(14,91)	13,93(13,09)	32,38(15,32)
DROP4	1	49,83(14,27)	68,58(27,18)	62,81(18,81)	34,82(15,46)	58,33(23,44)
	3	43,47(10,25)	46,16(25,96)	51,29(16,91)	27,50(31,77)	47,86(26,45)
	9	36,98(8,89)	44,68(19,85)	39,19(12,77)	13,93(13,09)	37,14(10,95)
DROP5	1	51,40(10,55)	34,21(21,88)	73,43(24,46)	34,82(12,18)	57,86(22,64)
	3	50,12(6,93)	35,58(7,54)	71,57(17,58)	12,86(15,72)	41,90(21,86)
	9	33,97(8,05)	34,13(13,54)	47,31(19,99)	20,71(13,55)	29,29(16,42)
DEL	1	53,76(7,40)	60,08(33,84)	83,12(8,20)	35,12(13,09)	45,48(25,48)
	3	51,35(6,91)	53,26(31,96)	67,55(11,96)	34,12(12,11)	59,52(23,94)
	9	33,30(8,68)	40,87(25,62)	32,79(14,61)	8,57(12,05)	**27,86(18,03)**

Uma vez que não foi possível a obtenção de resultados superiores para todos os algoritmos baseados em distância investigados, foram desenvolvidas combinações apenas para os conjuntos de dados em que os classificadores apresentaram, para os dados pré-processados, resultados superiores ou semelhantes aos obtidos para os conjuntos de dados originais. Essas combinações foram compostas por todos os algoritmos que permitiram ao classificador apresentar os melhores resultados em seu desempenho, ou seja, pelos algoritmos apresentados em **negrito** nas tabelas 3 a 5. Nesse âmbito, para o classificador SVM, apenas duas combinações foram desenvolvidas, a saber:

- Leukemia, composta pelos algoritmos AllkNN com k=1, k=3 e k=9, ENN e RENN com k=1, e algoritmos DROPs 2, 4 e 5 com k=9;
- ExpGen, composta pelos algoritmos ENN, RENN e DROP5 com k=3 e DEL com k=1 e k=3.

Para o classificador C4.5, cinco combinações foram elaboradas:

- Leukemia, composta pelos algoritmos ENN e RENN com k=1, k=3 e k=9 e AllkNN com k=3;
- Lung, composta pelos algoritmos AllkNN com k=1, k=3 e k=9, e algoritmos ENN e RENN com k=1 e k=3;
- ExpGen, formada pelos algoritmos AllkNN com k=1 e k=3. Para evitar a possibilidade de empate, foi também adicionado à esta combinação o algoritmo ENN com k=1, por este apresentar, dentre todos os algoritmos existentes que obtiveram resultados inferiores aos obtidos para os dados originais, o melhor desempenho;
- Golub64, composta pelos algoritmos RENN com k=1 e k=3, DROP1 com k=9, DROP2 com k=3 e k=9, DEL com k=9 e DROP5 com k=1. Para evitar a possibilidade de empate, foi eliminado desta combinação o algoritmo DEL com k=3, por este apresentar o pior resultado dentre todos os algoritmos considerados;
- Colon16, com o objetivo de evitar que uma grande quantidade de algoritmos fosse considerada em sua composição, foram selecionados apenas aqueles que apresentaram resultados superiores aos obtidos para os dados originais. Sendo assim, ela foi conposta pelos algoritmos AllkNN, ENN, RENN e DROP1 com k=1, k=3 e k=9 e DROP5 com k=9.

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

- Lung, constituída pelos algoritmos AllkNN, ENN e RENN com k=1, k=3 e k=9;
- ExpGen, composta pelos algoritmos AllkNN e ENN com k=1 e k=3 e RENN com k=1, k=3 e k=9;
- Golub64, formada pelos algoritmos AllkNN com k=1, k=3 e k=9;
- Colon16, composta pelos algoritmos AllkNN, ENN e RENN com k=1, k=3 e k=9. Para evitar a possibilidade de empate, o algoritmo DEL com k=9 foi eliminado desta combinação por apresentar o pior resultado dentre todos os algoritmos considerados;

Os resultados obtidos pelas combinações desenvolvidas são apresentados na Tabela 6. Nesta tabela, em itálico podem ser observados os resultados superiores ou semelhantes aos obtidos para os dados originais e, em negrito, os resultados superiores ou semelhantes aos melhores resultados anteriormente obtidos. Os desvios-padrão são apresentados entre parênteses.

Tabela 6: Erro médio das combinações desenvolvidas para os algoritmos baseados em distância.

Classificador	Conjuntos de dados	Quantidade de algoritmos	Taxa de erro Combinação	Taxa de erro Melhor Algoritmo	Melhor Algoritmo
SVM	Leukemia	8	**7,34(3,65)**	7,34(3,65)	ENN k=1
	ExpGen	5	*6,18(5,70)*	5,74(4,37)	DEL k=3
C4.5	Leukemia	7	**15,06(6,11)**	16,81(4,98)	RENN k=9
	Lung	7	**5.06(4.08)**	5,56(4,97)	ENN k=1
	ExpGen	3	9,66(6,32)	8,16(7,46)	AllkNN k=1
	Golub64	7	*8.22(7.43)*	6,97(7,36)	DROP1 k=9
	Colon16	13	**18.10(16.61)**	18,11(14,63)	AllkNN k=3 e k=9
RIPPER	Lung	9	*8,05(5,34)*	6,05(3,93)	RENN k=1
	ExpGen	7	**11,45(6,01)**	11,52(6,74)	ENN k=1
	Golub64	3	11.07(8.94)	6,79(9,84)	AllkNN k=3
	Colon16	9	**14.76(16.56)**	14,76(19,95)	RENN k=1

A análise da Tabela 6 indica que, para o classificador SVM, as combinações apresentaram resultados superiores aos obtidos para os dados originais sem resultar, no entanto, em desempenho superior. Na avaliação do classificador C4.5, de um modo geral, todas as combinações, com exceção da desenvolvida para o conjunto ExpGen, obtiveram resultados superiores àqueles encontrados para os dados originais. Nesse sentido, para o conjunto Colon16, a combinação apresentou resultado estatisticamente igual aos melhores resultados até então encontrados para esse conjunto e, para os conjuntos Leukemia e Lung, o melhor desempenho foi obtido pelas combinações desenvolvidas. Para o conjunto Golub64, a combinação produziu resultado superior ao encontrado para os dados originais, porém inferior aos melhores resultados obtidos por alguns dos algoritmos de detecção de ruído aplicados individualmente.

A análise dos resultados para o classificador RIPPER indica que apenas a combinação desenvolvida para o conjunto Golub64 produziu resultados inferiores aos obtidos para os dados originais. As desenvolvidas para os conjuntos ExpGen e Colon16 apresentaram resultados estatisticamente semelhantes aos melhores resultados obtidos pelos algoritmos baseados em distância. A combinação desenvolvida para o conjunto Lung apresentou resultado superior ao obtido para os dados originais, porém inferior a alguns dos algoritmos de detecção de ruído baseados em distância aplicados individualmente.

Os experimentos sugerem que as diferentes estratégias de detecção de ruído investigadas foram aptas na tarefa de eliminar ruído, melhorando a acurácia preditiva dos classificadores ao utilizarem conjuntos de dados pré-processados (livres de ruído). Com exceção do algoritmo SVM, aplicado após a etapa de detecção de ruído realizada pelos algoritmos baseados em distância, que teve seu desempenho deteriorado, os demais algoritmos avaliados apresentaram desempenho superior quando aplicados aos conjuntos de dados pré-processados. Portanto, a análise dos resultados experimentais apresentados demonstra que os algoritmos baseados em distância investigados podem ser alternativas eficientes para a detecção de ruído em dados de expressão gênica.

As combinações também apresentaram bons resultados, embora não tenha sido possível a obtenção de melhoria significativa no desempenho para todos os conjuntos de dados e classificadores avaliados. Nesse sentido, as combinações demonstraram ser capazes de tratar, com desempenho satisfatório, problemas de expressão gênica representados por conjuntos de dados complexos e de elevada dimensionalidade sendo, portanto, adequadas para utilização em demais problemas relacionados, não abordados neste trabalho. Uma vantagem do uso das combinações é que, ao custo de possivelmente não obter o melhor desempenho possível com a detecção de ruído, melhorar ou manter as taxas de classificação corretas obtidas com os dados originais. Outra vantagem é reduzir a necessidade de descobrir qual a técnica de detecção mais indicada para um novo conjunto de dados.

6. CONSIDERAÇÕES FINAIS

Este trabalho dá continuidade ao trabalho previamente desenvolvido em [23], e busca mostrar que a presença de ruído em um conjunto de dados influencia o desempenho de classificadores de AM. Para isso, foram realizados experimentos para avaliar o desempenho de classificadores de AM para conjuntos de dados originais e pré-processados, ou seja, teoricamente livres de ruído. É importante ressaltar que o objetivo principal deste trabalho é analisar o efeito da limpeza dos dados no desempenho dos classificadores, ou seja, se os dados eliminados são efetivamente ruído ou informação útil é um problema não abordado no presente artigo.

Para os experimentos realizados, foram utilizados dados de expressão gênica e, para a detecção dos ruídos, foram investigados algoritmos baseados em distância. Os algoritmos baseados em distância identificaram e removeram, em geral, uma quantidade elevada de ruído nos conjuntos de dados avaliados. Os resultados obtidos foram satisfatórios para a maioria dos experimentos realizados, reforçando a questão da influência de ruídos no desempenho de classificadores, e a importância de sua eliminação para a qualidade e confiabilidade dos dados. A detecção de ruído valendo-se da combinação também se mostrou interessante pois, em geral, foram obtidas melhorias no desempenho dos classificadores avaliados.

REFERÊNCIAS

[1] S. Verbaeten. "Identifying mislabeled training examples in ILP classification problems." pp. 1–8. Proceedings of 12th Belgian-Dutch Conference on Machine Learning, 2002.

[2] X. Zhu and X. Wu. "Class noise vs. Attribute noise: A quantitative study of their impacts." *Artificial Intelligence Review, v. 22, n. 3*, pp. 177–210, 2004.

[3] J. Tang, Z. Chen, A. W. Fu and D. Cheung. "A robust outlier detection scheme in large data sets." Proceedings of 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 2002.

[4] V. Hodge and J. Austin. "A survey of outlier detection methodologies." *Artificial Intelligence Review, v. 22*, pp. 85–126, 2004.

[5] T. Mitchell. "Machine Learning." *McGraw Hill*, 1997.

[6] C. C. Aggarwal, A. Hinneburg and D. A. Keim. "On the surprising behavior of distance metrics in high dimensional space." pp. 420–434. 8th International Conference on Database Theory (ICDT 2001), Lecture Notes in Computer Science, v. 1973, Springer Verlag, 2001.

[7] D. R. Wilson and T. R. Martinez. "Improved heterogeneous distance functions." *Journal of Artificial Intelligence Research (JAIR)*, pp. 1–34, v. 6(1), 1997.

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

[9] I. Tomek. "Two modifications of CNN." *IEEE Transactions on Systems, Man and Cybernetics*, pp. 769–772, v. 6(11), 1976.

[10] D. R. Wilson and T. R. Martinez. "Reduction techniques for instance-based learning algorithms." *Machine Learning*, pp. 257–286, v. 38(3), 2000.

[11] M. Brown, W. Grundy, D. Lin, N. Christianini, C. J. Sugnet and D. Haussler. "Support vector machine classification of microarray gene expression data." *Technical Report UCSC-CRL 99-09, Departamento de Computação, Universidade California Santa Cruz, Santa Cruz, CA*, 1999.

[12] E. J. Yeoh, M. E. Ross, S. A. Shurtle, W. K. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C. H. Pui, W. E. Evans, C. Naeve, L. Wong and J. R. Downing. "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling." *Cancer Cell, v. 1(2)*, pp. 133–143, 2002.

[13] S. Monti, P. Tamayo, J. Mesirov and T. Golub. "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data." *Machine Learning, v. 52(1-2)*, pp. 91–118, 2003.

[14] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine. "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays." pp. 6745–6750, v. 96, 1999.

[15] B. F. Souza. "Seleção de Características em SVMs Aplicadas a Dados de Expressão Gênica." *Universidade de São Paulo (USP-São Carlos). Dissertação de Mestrado*, 2005.

[16] P. Tamayo, D. Slonim, J. Mesirow, Q. Zhu, S. Kitareewan, E. Drovsky, E. S. Lander and G. T. R. "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation." pp. 2907–2912, v. 96, 1999.

[17] V. N. Vapnik. "The Nature of Statistical Learning Theory." *Springer-Verlag, 2a. edição*, 1995.

[18] W. W. Cohen. "Fast effective rule induction." pp. 115–123. Proceedings of 12th International Conference on Machine Learning, 1995.

[19] J. Demsar. "Statistical comparisons of classifiers over multiple datasets." *Journal of Machine Learning Research*, pp. 1–30, v. 7, 2006.

[20] R. Collobert and S. Bengio. "SVMTorch: Support vector machines for large-scale regression problems." *Journal of Machine Learning Research, v. 1*, pp. 143–160, 2001.

[21] J. R. Quinlan. "C4.5: Programs for Machine Learning." *Morgan Kaufmann*, 1993.

[22] E. Frank and I. H. Witten. "Data Mining: Practical Machine Learning Tools and Techniques." *Morgan Kaufmann*, 2005.

[23] G. L. Libralon, A. C. P. L. F. Carvalho and A. C. Lorena. "Pre-processing for noise detection in gene expression classification data." *Journal of the Brazilian Computer Society, v. 15*, pp. 3–11, 2009.