UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Uma Comparação Experimental de Ensembles de Árvores de Decisão e de Regras Utilizando a Técnica de Boosting

Flávia Cristina Bernardini
Maria Carolina Monard

Nº 161

RELATÓRIOS TÉCNICOS

São Carlos – SP
Abr./2002

SYSNO 1239113
DATA / /
ICMC - SBAB

Uma Comparação Experimental de Ensembles de Árvores de Decisão e de Regras Utilizando a Técnica de Boosting*

Flávia Cristina Bernardini

Maria Carolina Monard

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {fbernardi, mcmonard}@icmc.sc.usp.br

Resumo

Um dos temas atuais em aprendizado de máquina supervisionado está relacionado com o estudo de métodos de construção de ensembles de classificadores, os quais podem ser mais precisos que os classificadores individuais que compõem o ensemble. Vários métodos têm sido propostos para construir ensembles, entre eles métodos que manipulam o conjunto de exemplos de treinamento. Existem várias técnicas para manipular o conjunto de treinamento, entre elas a técnica de Boosting. O objetivo deste trabalho é analisar experimentalmente a técnica de Boosting utilizando dois algoritmos de aprendizado de máquina: um que induz árvores de decisão e outro que induz regras.

Palavras-Chave: Aprendizado de Máquina, Ensemble, Boosting.

Abril 2002

*Trabalho realizado com auxílio da FAPESP

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

[1]Este documento foi elaborado com o LaTeX, sua bibliografia é mantida com o auxílio da ferramenta BibView (Prati et al., 1999) e gerada automaticamente pelo BibTeX

1 Introdução

A qualidade das hipóteses induzidas pelos atuais sistemas de Aprendizado de Máquina (AM) depende principalmente da quantidade e da qualidade dos atributos e exemplos utilizados no treinamento. Freqüentemente, resultados experimentais obtidos sobre grandes bases de dados, nas quais muitos atributos irrelevantes estão presentes, resultam em hipóteses de baixa precisão. Por outro lado, muitos dos sistemas de aprendizado de máquina conhecidos não estão preparados para trabalhar com uma quantidade muito grande de exemplos. Assim, uma das áreas de pesquisa mais ativas em aprendizado de máquina tem girado em torno de técnicas que sejam capazes de ampliar a capacidade dos algoritmos de aprendizado para processar muitos exemplos de treinamento, atributos e classes (Dietterich, 2000b; Michalski et al., 1998). Esses problemas são típicos da área de mineração de dados em grandes bases (Data Mining) (Cabena et al., 1998; Mannila, 1996; Mitchell, 1999; Weiss and Indurkhya, 1998).

Para que os conceitos sejam aprendidos a partir de grandes bases de dados, utilizando AM, pode-se utilizar duas abordagens. A primeira realiza uma seleção de exemplos e atributos mais relevantes (Blum and Langley, 1997), e a segunda é a abordagem de ensembles, tratada neste trabalho. Nessa abordagem primeiramente são retiradas, por exemplo, L amostras (subconjuntos) do conjunto de exemplos disponível para treinamento. Logo após, cada um desses L subconjuntos é submetido a algum algoritmo de AM criando assim L classificadores, os quais podem ser construídos em paralelo. Um ensemble é um conjunto de classificadores cujas decisões individuais são combinadas de alguma forma para classificar um novo caso.

Vários métodos têm sido propostos para construir ensembles, entre eles métodos que manipulam o conjunto de exemplos de treinamento. Existem várias técnicas para manipular o conjunto de treinamento, entre elas a técnica de Boosting. O objetivo deste trabalho é analisar experimentalmente a técnica de Boosting utilizando dois algoritmos de aprendizado de máquina: um que induz árvores de decisão e outro que induz regras. Os experimentos foram realizados utilizando oito conjuntos de dados do mundo real extraídos da UCI (Blake et al., 1998).

Quanto à organização, este trabalho está organizado da seguinte forma: Na Seção 2, é feita uma revisão sobre aprendizado de máquina supervisionado. Na Seção 3, é descrita a abordagem de ensembles, com uma breve descrição sobre os principais métodos de Ensembles e uma descrição mais detalhada sobre a técnica Boosting. Na Seção 4, são descritos os experimentos realizados. Na Seção 5, são descritos os resultados experimentais obtidos. Na Seção 5 é feita uma análise dos resultados obtidos nos experimentos. E na Seção 6 é apresentada a conclusão do trabalho realizado.

2 Aprendizado de Máquina Supervisionado

No problema padrão de AM supervisionado, ao algoritmo de aprendizado de máquina é dado um conjunto de exemplos S de treinamento, com N exemplos $T_i, i = 1, ..., N$, escolhidos de um domínio X com uma distribuição D fixa, desconhecida e arbitrária, da forma $\{(\mathbf{x}_1,y_1),...,(\mathbf{x}_N,y_N)\}$ para alguma função desconhecida $y = f(\mathbf{x})$, como mostra a Tabela 1 na página seguinte. Os $\mathbf{x}_i$ são tipicamente vetores da forma $<x_{i1},x_{i2},...,x_{iM}>$, com valores discretos ou numéricos, como peso, altura, cor, idade, e assim por diante. Assim, x_{ij} refere-se ao valor do atributo (ou *feature*)

j, denominado $\mathbf{X}_j$ do exemplo T_i, como mostra a Tabela 1. Os valores y_i referem-se ao valor do atributo Y, freqüentemente denominado classe.

	X_1	X_2	...	X_M	Y
T_1	x_{11}	x_{12}	...	x_{1M}	y_1
T_2	x_{21}	x_{22}	...	x_{2M}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
T_N	x_{N1}	x_{N2}	...	x_{NM}	y_N

Tabela 1: Conjunto de Exemplos no Formato Atributo-Valor

Os valores de y são tipicamente pertencentes a um conjunto discreto de classes, representado por $\{C_1, ..., C_K\}$, quando se trata de *classificação* ou ao conjunto de números reais em caso de *regressão*.

Dado um conjunto S de exemplos de treinamento a um algoritmo de aprendizado, um *classificador* $\mathbf{h}$ será induzido. O classificador consiste de uma *hipótese* feita sobre a verdadeira (mas desconhecida) função f. Dados novos exemplos $\mathbf{x}$, o classificador $\mathbf{h}$ prediz o valor correspondente y.

Dado um conjunto $\mathcal{H}$ de classificadores, denotados por $\mathbf{h}_1$,...,$\mathbf{h}_L$, pode-se combinar suas decisões individuais — tipicamente através de uma votação com ou sem peso — para classificar novos exemplos, formando, assim, um **ensemble** de classificadores.

3 Ensembles de Classificadores

Uma das áreas ativas em aprendizado de máquina supervisionado estuda métodos de construção de ensembles de classificadores. Um resultado interessante é que ensembles de classificadores podem ser mais precisos que os classificadores individuais que compõem o ensemble.

Uma condição para que um ensemble de classificadores seja mais preciso que seus componentes é que os classificadores que compõem o ensemble sejam distintos (Hansen and Salamon, 1990). Um classificador preciso é um classificador que faz a predição da classe de um novo exemplo $\mathbf{x}$ com uma margem de erro menor do que simplesmente adivinhar o valor de y dado $\mathbf{x}$. Dois classificadores são distintos se cometem erros diferentes em novos conjuntos de exemplos.

Para melhor compreender essa condição, considera-se um ensemble de três classificadores $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$ e um novo caso (ou exemplo) $\mathbf{x}$, como mostra a Figura 1 na página seguinte (Dietterich, 2000a). Esse novo exemplo $\mathbf{x}$ será classificado por cada classificador $\mathbf{h}_1, \mathbf{h}_2$ e $\mathbf{h}_3$ com uma das classes do conjunto discreto de classes $\{C_1, ..., C_K\}$. Seja $\mathbf{h}_1(\mathbf{x})$ a classificação atribuída a esse novo exemplo $\mathbf{x}$ pelo classificador $\mathbf{h}_1$, $\mathbf{h}_2(\mathbf{x})$ pelo classificador $\mathbf{h}_2$ e $\mathbf{h}_3(\mathbf{x})$ pelo classificador $\mathbf{h}_3$.

Se os três classificadores são idênticos, então, quando $\mathbf{h}_1(\mathbf{x})$ está errado, $\mathbf{h}_2(\mathbf{x})$ e $\mathbf{h}_3(\mathbf{x})$ também estão. Entretanto, se os erros cometidos pelos classificadores forem não correlacionados, então quando $\mathbf{h}_1(\mathbf{x})$ está errado, $\mathbf{h}_2(\mathbf{x})$ e $\mathbf{h}_3(\mathbf{x})$ podem estar corretos, de forma que o voto majoritário pode classificar corretamente o exemplo $\mathbf{x}$. Em geral, dado um ensemble composto por L classificadores $\mathbf{h}_1, ..., \mathbf{h}_L$, para cada novo exemplo $\mathbf{x}$ a ser classificado por esses classificadores, tem-se uma série de L ensaios. Considerando que cada um desses ensaios é independente e que cada

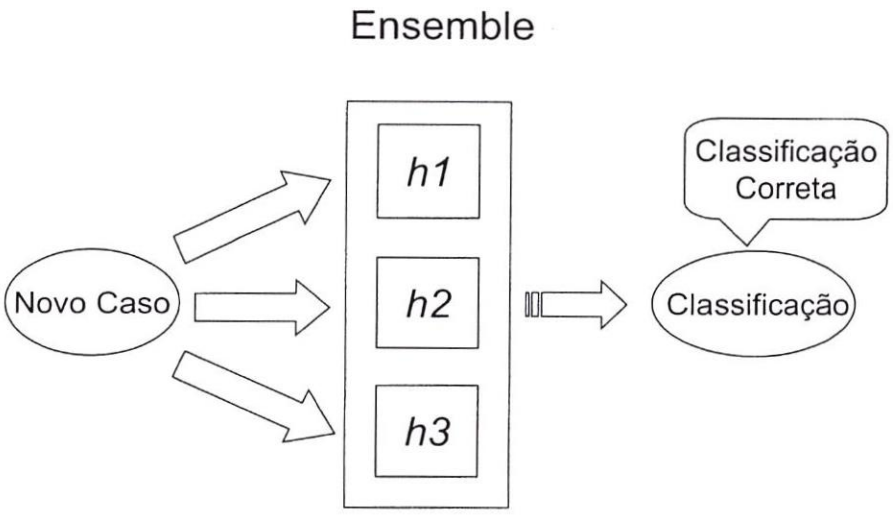

Figura 1: Ensemble de Três Classificadores

ensaio é um sucesso na classificação de $\mathbf{x}$ com probabilidade[1] p ou uma falha com probabilidade $1-p$, então a probabilidade do número de sucessos ser l em L ensaios é dada por

$$P(Z=l) = \binom{L}{l} p^l (1-p)^{L-l} \tag{1}$$

Mais precisamente, se as taxas de erro de L classificadores $\mathbf{h}_1, ..., \mathbf{h}_L$ são todas iguais a $p < \frac{1}{2}$ e se os erros são independentes, então a probabilidade do voto majoritário estar errado, ou seja, a probabilidade do ensemble ter mais de 50% de classificadores que falharam, é dada por

$$P(Z > \frac{L}{2}) = 1 - \sum_{l=0}^{\frac{L}{2}} \binom{L}{l} p^l (1-p)^{L-l} \tag{2}$$

que corresponde à área sob o gráfico da distribuição binomial na qual mais que $\frac{L}{2}$ hipóteses estão erradas.

Na Figura 2, são mostrados *ensembles* ideais formados com hipóteses independentes, todas com taxas de erro de 0.3, 0.4 e 0.45 respectivamente. Pode ser observado que quanto maior o valor da taxa de erro das hipóteses, maior é a taxa de erro do ensemble, mas ainda assim a taxa de erro do ensemble é menor que a taxa de erro de cada hipótese que o compõe. Por exemplo, para um ensemble simulado com 21 hipóteses, cada uma delas possuindo uma taxa de erro de 0.3, a área de curva para 11 ou mais hipóteses simultaneamente incorretas é 0.026, o qual é muito menor que a taxa de erro das hipóteses individuais, conforme pode ser observado na Figura 2 na próxima página. Observa-se também que quanto maior o número de hipóteses, menor é a taxa de erro do ensemble.

[1] É considerado que essa probabilidade é a taxa de erro do ensaio, ou seja, da hipótese $\mathbf{h}_l$

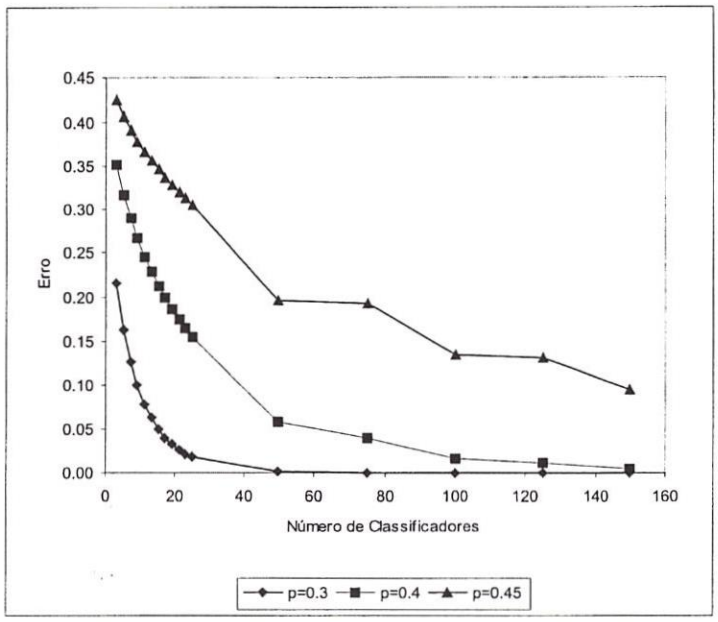

Figura 2: Probabilidade do Voto Majoritário Estar Errado com L Hipóteses, $L = 1, ..., 150$, com Taxa de Erro de Cada Hipótese Igual a 0.3, 0.4 e 0.45

Por outro lado, se as hipóteses individuais têm erros não correlacionados mas com taxas de erro excedendo 0.5, então a taxa de erro do ensemble aumentará como resultado da votação. A Figura 3 na página seguinte, mostra a curva do erro de um ensemble ideal composto por hipóteses independentes, todas com taxa de erro de 0.7. Pode ser observado que incrementando o número de hipóteses, maior é o erro cometido pelo ensemble.

Para concluir, considerando a probabilidade do voto majoritário estar errado no modelo ideal de ensembles, definido pela equação 2, a chave para o sucesso dos métodos de criação de ensembles está em construir classificadores individuais com taxas de erro abaixo de 0.5. Uma observação é que as simulações mostradas tratam situações ideais, nas quais todas as hipóteses que compõem o ensemble são ensaios independentes, ou seja, não correlacionados. Em aprendizado de máquina, o que se tenta fazer é tornar os ensaios (as hipóteses) não correlacionados, ou, se a correlação existir — e, na prática, geralmente essa correlação existe —, torná-la mínima.

3.1 Razões para a Construção de Ensembles

Os ensembles são construídos para se tentar melhorar o poder de predição dos algoritmos de AM. Mas quais os motivos que levam a construir bons ensembles? As razões são, fundamentalmente, três (Dietterich, 2000a):

1. *Estatística.* Um algoritmo de aprendizado pode ser visto como um algoritmo de busca no espaço de hipóteses $\mathcal{H}$ para identificar a melhor hipótese. O problema estatístico aparece

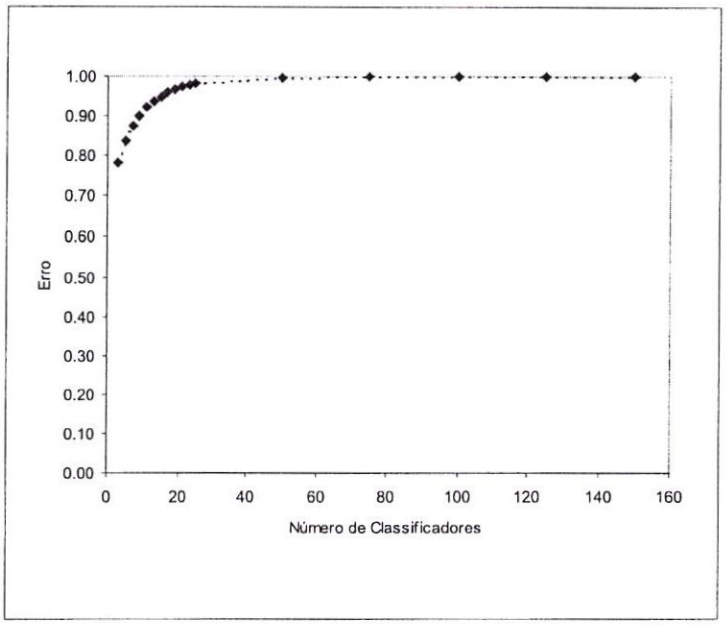

Figura 3: Probabilidade do Voto Majoritário Estar Errado com L Hipóteses, $L = 1, ..., 150$, com Taxa de Erro de Cada Hipótese Igual a 0.7

quando a quantidade de dados disponíveis para treinamento é muito pequena comparada ao tamanho do espaço de hipóteses. Sem dados suficientes, o algoritmo de aprendizado pode encontrar muitas hipóteses diferentes em $\mathcal{H}$, as quais têm a mesma precisão sobre os dados de treinamento. Construindo um ensemble com todos estes classificadores, o algoritmo pode calcular a média de seus votos e reduzir o risco de escolher o classificador errado. A Figura 4 (topo à esquerda) ilustra essa situação. A curva externa denota o espaço de hipóteses $\mathcal{H}$ e a interna denota o conjunto de hipóteses que têm uma boa precisão sobre o conjunto de treinamento. O ponto rotulado com a letra f é a hipótese verdadeira, e pode-se observar que tirando a média da precisão das hipóteses, podemos encontrar uma boa aproximação de f.

2. *Computacional.* Muitos algoritmos de aprendizado trabalham fazendo alguma busca local a qual pode parar em algum ótimo local. Por exemplo, alguns algoritmos de redes neurais utilizam métodos de busca local — como grandiente descendente — para encontrar pesos localmente ótimos para a rede; algoritmos de árvore de decisão aplicam regras gulosas de particionamento para gerar a árvore de decisão. Nos casos nos quais há quantidade suficiente de dados de treinamento — o que indica que não há problema estatístico —, é computacionalmente difícil para o algoritmo de aprendizado encontrar a melhor hipótese. De fato, o problema de encontrar a menor árvore de decisão que seja consistente com um conjunto de treinamento é NP-hard (Hyafil and Rivest, 1976). Similarmente, encontrar os pesos para a menor rede neural possível consistente com os exemplos de treinamento é também NP-hard (Blum and Rivest, 1988). Por outro lado, construindo-se um ensemble executando várias vezes o algoritmo de busca local partindo, a cada iteração, de diferentes

pontos, pode-se obter uma melhor aproximação da verdadeira (e desconhecida) função f que seja mais precisa que qualquer um dos classificadores individuais, como é ilustrado na Figura 4 (topo à direita).

3. *Representacional.* Às vezes, não é possível representar a verdadeira função f pelas hipóteses em $\mathcal{H}$. Entretanto, unindo-se as hipóteses ou dando-se pesos a cada uma delas e unindo-as, pode ser possível expandir o espaço das funções representáveis. A Figura 4 (inferior) ilustra esta situação.

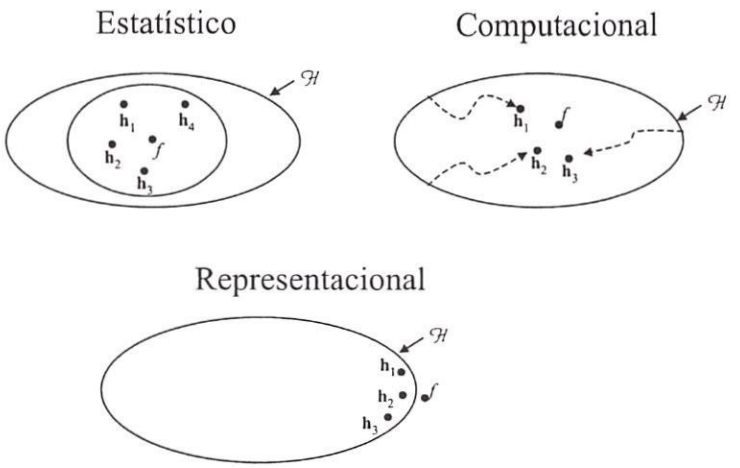

Figura 4: Três Motivos Fundamentais que Explicam Porquê os Ensembles Funcionam Melhor que um Único Classificador

3.2 Métodos de Construção de Ensembles

Muitos métodos de construção de ensembles têm sido desenvolvidos. Esses métodos podem ser divididos, basicamente, nos seguintes cinco métodos de propósitos gerais (Dietterich, 2000a):

1. Enumeração do espaço de hipóteses $\mathcal{H}$;
2. Manipulação dos exemplos de treinamento S;
3. Manipulação do conjunto de atributos X;
4. Manipulando do conjunto de classes C; e
5. Inserção de aleatoriedade nos algoritmos de aprendizado de máquina.

A descrição de cada um desses métodos pode ser encontrada em (Bernardini, 2001). Neste trabalho é considerada especialmente a técnica de Boosting, descrita a seguir, a qual é uma das técnicas que manipula os exemplos de treinamento.

3.2.1 Manipulando os Exemplos de Treinamento

A idéia geral deste método é manipular os exemplos de treinamento para gerar múltiplas hipóteses. Neste caso, executa-se o algoritmo de aprendizado várias vezes, cada vez com um subconjunto de treinamento diferente. Esta técnica trabalha especialmente bem quando se utilizam algoritmos de aprendizado **instáveis**. Algoritmos instáveis são aqueles que geram classificadores bastante diferentes mesmo ocorrendo pequenas alterações no conjunto de treinamento, ou seja, não é necessário que grandes alterações ocorram no conjunto de treinamento para que classificadores diferentes sejam gerados. Algoritmos de aprendizado de árvores de decisão, de redes neurais e de regras são instáveis, enquanto que algoritmos de regressão linear, vizinho mais próximo (*nearest neighbor*) e *linear threshold* são geralmente bastante estáveis.

As principais técnicas que manipulam o conjunto de treinamento para a construção de um ensemble podem ser divididos em duas famílias (Breiman, 1996b; Bauer and Kohavi, 1999):

1. *Família P &C.* Técnicas P&C — *Perturb and Combine* (Perturbar e Combinar) — "perturbam" o conjunto de treinamento ou o método de construção da hipótese para induzir diferentes hipóteses as quais são combinadas para gerar o ensemble. Técnicas pertencentes à esta família são **Bagging** (Breiman, 1996a), **Wagging** (Bauer and Kohavi, 1999) e **Cross-validated Comitees** (Parmanto et al., 1992).

2. *Família ARC.* ARC — *Adaptively Resample and Combine* (Reamostrar e Combinar Adaptativamente) — refere-se à família de algoritmos que combinam e refazem a amostra de exemplos de treinamento adaptativamente. Técnicas pertencentes a esta família são **Boosting** (Freund and Schapire, 1997; Breiman, 1996b), **Arcing** (Bauer and Kohavi, 1999) e **Windowing** (Quinlan, 1988).

A seguir será descrita a técnica de boosting, a qual será utilizada neste trabalho.

3.2.2 Boosting

O Boosting (Bauer and Kohavi, 1999) refere-se ao problema geral de produzir predições precisas combinando-se hipóteses rústicas e imprecisas, as quais são geralmente produzidas por algoritmos de aprendizado fraco. Esta técnica torna algoritmos de aprendizado fracos em fortes. Um algoritmo de aprendizado **forte** é um algoritmo que, dado $\epsilon, \delta > 0$ e acesso randômico aos exemplos, gera hipóteses com probabilidade $1 - \delta$ e com erro ϵ tal que

$$P(erro(\mathbf{h}) \leq \epsilon) = 1 - \delta, \text{ com } \epsilon, \delta \rightarrow 0 \tag{3}$$

Assim, quanto menores os valores de ϵ e δ, mais forte é o algoritmo. Além disso, o tempo de execução deve ser polinomial em $1/\epsilon$, $1/\delta$ e outros parâmetros relevantes, como o tamanho da amostra, e o tamanho, ou complexidade, do conceito a ser aprendido.

Um algoritmo de aprendizado **fraco** satisfaz a mesma condição da Equação 3 mas somente para $\epsilon \geq 1/2 - \gamma$ onde $\gamma > 0$ é também uma constante, ou decrementa com $1/p$, onde p é um polinômio

dos parâmetros relevantes. Ou seja, o valor de ϵ pode se aproximar de 0.5, e a probabilidade de isto ocorrer é grande (Freund and Schapire, 1997).

O algoritmo de boosting projetado por Freund e Schapire (Freund and Schapire, 1997) — AdaBoost.M1, mostrado no Algoritmo 1 — utiliza um indutor e um conjunto de exemplos de treinamento $S = \{(\mathbf{x}_i, y_i), i = 1, ..., N\}$. Para cada exemplo $(\mathbf{x}_i, y_i)$ existe um peso w_i, ou seja, existe um vetor de pesos W com N elementos. Esse vetor é inicializado com $w_i = 1/N$, $i = 1, ..., N$. Após, para cada iteração $l = 1, ..., L$, o algoritmo de Boosting executa os seguintes passos:

- O algoritmo de aprendizado constrói uma hipótese $\mathbf{h}_l$ utilizando o conjunto de treinamento S e o vetor de pesos W;
- O erro α_l, isto é, o erro($\mathbf{h}_l$) no Algoritmo 1, dessa hipótese sobre o conjunto de treinamento é calculado como sendo a soma dos pesos w_{l_i} dos exemplos erroneamente classificados pela hipótese $\mathbf{h}_l$;
- Se $\alpha_l \geq 1/2$, o processo termina. Senão, os pesos $w_{(l+1)_i}$ para a próxima iteração são recalculados como sendo

$$w_{(l+1)_i} = w_{l_i} \frac{\alpha_l}{1 - \alpha_l}$$

  se $\mathbf{h}_l$ acerta na classificação do exemplo i.

A hipótese $\mathbf{h}^*$ é obtida pela votação das hipóteses geradas $\mathbf{h}_l, l = 1, ..., L$. Se $\mathbf{h}_l$ classifica um exemplo $\mathbf{x}$ como sendo da classe C_k, o voto total para a classe C_k é incrementado de $\log((1\text{-}\alpha_l)/\alpha_l)$. $\mathbf{h}^*$ então associa ao exemplo $\mathbf{x}$ a classe com maior valor total de voto.

Com esta técnica, a taxa de erro de $\mathbf{h}^*$ cai exponencialmente para próximo de 0 (zero) conforme o valor de L aumenta. Assim, se forem consideradas as hipóteses com uma taxa de erro menor que 50% nos exemplos de teste, ou seja, uma seqüência de classificadores fracos, pode-se criar um classificador forte $\mathbf{h}^*$ que é, no mínimo, tão preciso quanto, e usualmente mais preciso que, o menos preciso dos classificadores fracos (Freund and Schapire, 1997).

4 Experimentos Realizados

Foram realizados vários experimentos em domínios do mundo real para testar a performance do algoritmo de boosting para criação de ensembles utilizando dois algoritmos de AM, um que induz árvores de decisão e outro que induz regras, ambos implementados no sistema *See5*. A maioria dos conjuntos de dados utilizados são provenientes do repositório da UCI (Blake et al., 1998).

Os oito conjuntos de dados escolhidos têm diferentes tipos de atributos, os quais são nominais, contínuos ou uma combinação destes. Três destes conjuntos de dados também possuem valores desconhecidos.

Algoritmo 1 AdaBoost.M1

Parâmetros: Exemplos: um conjunto de n exemplos rotulados $\{(\mathbf{x}_i, y_i), i = 1, ..., N\}$
Indutor: um algoritmo de aprendizado
L: número de classificadores

1: **procedure** boost(N,Exemplos,Indutor,l)
2: **for all** $i := 1$ to N **do**
3: Inicializar os pesos: $w_i := 1/N$
4: **end for**
5: **for** l:=1 to L **do**
6: **for** i:=1 to N **do**
7: Normalizar os pesos: $p_i := w_i/(\sum_{j=1}^{N} w_j)$
8: **end for**
9: $\mathbf{h}_L$:=Indutor(Exemplos,p)
10: Computar o erro da hipótese: erro($\mathbf{h}_L$):=$\sum_{\mathbf{x}_i \in S_l} p_i || \mathbf{h}_l(\mathbf{x}_i) \neq y_i ||$
11: **if** erro($\mathbf{h}_l$) $\geq 1/2$ **then**
12: $L := l - 1$
13: O erro da hipótese atual é maior que ou igual a 1/2: **exit loop**
14: **end if**
15: β_l :=erro($\mathbf{h}_l$)/(1−erro($\mathbf{h}_l$))
16: **for** i:=1 to N **do**
17: **if** $\mathbf{h}_l(\mathbf{x}_i) = y_i$ **then**
18: Computar novos pesos: $w_i := w_i \beta_i$
19: **end if**
20: **end for**
21: **end for**
22: $\mathbf{h}^*(\mathbf{x}) := \text{argmax}_{y \in \{C_1, ..., C_K\}} \sum_{l=1}^{L} \mathbf{h}_l(\mathbf{x})$
23: **return** $\mathbf{h}^*$

4.1 Descrição Geral dos Conjuntos de Dados

Segue uma descrição resumida dos oito conjuntos de dados utilizados. Uma descrição mais detalhada pode ser encontrada em (Lee et al., 1999).

Bupa Este conjunto de dados consiste da predição de quando um paciente do sexo masculino têm problemas no fígado baseado em vários exames de sangue e na quantidade de consumo de álcool.

Pima Neste conjunto de dados, todos os pacientes são do sexo feminino com pelo menos 21 anos descendentes dos Índios Pima que vivem próximo a Phoenix, Arizona, EUA. O problema consiste em predizer quando um paciente é diabético.

CMC Os exemplos neste dataser consistem de mulheres casadas que não estavam grávidas ou não sabiam quando se deu a entrevista. O problema é predizer a escolha do método contraceptivo corrente (*none*, *long-term methods* ou *short-term methods*) de uma mulher baseado em suas características demográficas e sócio-econômicas.

WBC Neste conjunto de dados, o problema é predizer quando uma amostra de tecido de tumor de uma mama de um paciente é maligno ou benigno.

Hungaria Este conjunto de dados é utilizado para diagnosticar doenças do coração.

CRX Este conjunto de dados contém aplicações de cartões de créditos. Todos os valores e nomes dos atributos foram mudados para símbolos menos significativos para proteger a confidência dos dados.

Hepatiti Este conjunto de dados é utilizado para predizer a expectativa de vida dos pacientes com hepatiti.

Anneal Este conjunto de dados foi doado por David Sterling e Wray Buntine.

4.2 Características dos Conjuntos de Dados

A Tabela 2 sumariza as características dos conjuntos de dados utilizados neste estudo. Ela mostra, para cada conjunto de dados, o número de exemplos(#Exemplos), número e porcentagem de exemplos com valores duplicados (que aparecem mais de uma vez) ou conflitantes (que possuem o mesmo atributo-valor mas têm diferentes classificações), número de atributos(#Atributos) contínuos e nominais, distribuição de classes, o erro majoritário e se o conjunto de dados tem ao menos um valor desconhecido[2].

Conjunto de Dados	# Exemplos	#Duplicados ou conflitantes (%)	# Atributos (cont.,nom.)	Classe	Classe %	Erro Majoritário	Valores Desconhecidos

continua na próxima página

[2]Esta informação foi obtida utilizando a ferramenta *info* da $\mathcal{MLC}$++.

continuação da página anterior

Conjunto de Dados	# Exemplos	#Duplicados ou conflitantes (%)	# Atributos (cont.,nom.)	Classe	Classe %	Erro Majoritário	Valores Desconhecidos
Bupa	345	4 (1.16%)	6 (6,0)	1	42.03%	42.03%	N
				2	57.97%	no valor 2	
Pima	769	1 (0.13%)	8 (8,0)	0	65.02%	34.98%	N
				1	34.98%	no valor 0	
CMC	1473	115 (7.81%)	9 (2,7)	1	42.70%	57.30%	N
				2	22.61%	no valor 1	
				3	34.69%		
WBC	699	8 (1.15%)	10 (10,0)	2	65.52%	34.48%	Y
				4	34.48%	no valor 2	
Hungaria	294	1 (0.34%)	13 (13,0)	presence	36.05%	36.05%	Y
				absence	63.95%	no valor absence	
CRX	690	0 (0.0%)	15 (6,9)	yes	44.49%	44.49%	N
				no	55.51%	no valor no	
Hepatiti	155	0 (0%)	19 (6,13)	die	20.65%	20.65%	Y
				live	79.35%	no valor live	
Anneal	898	12 (1.37%)	38 (6,32)	1	42.70%	57.30%	N
				2	22.61%	no valor 1	
				3	34.69%		

Tabela 2: Sumário das Características dos Conjuntos de Dados

Os conjuntos de dados estão na ordem ascendente do número de atributos, e são mostrados também nessa ordem nas próximas tabelas e gráficos. A Figura 5 mostra a dimensão dos conjuntos de dados, ou seja, o número de atributos e o número de exemplos de cada conjunto de dados. Deve ser observado que a respeito da larga variação, o número de exemplos na Figura 5 é representada por $\log_{10}(\#\text{Exemplos})$.

4.3 Descrição dos Experimentos

O objetivo dos experimentos realizados é medir e analisar o erro de cada ensemble gerado utilizando a técnica de Boosting e dois algoritmos de AM, implementados no sistema *See5* — um que induz árvores de decisão e outro que induz regras.

Para utilizar o sistema *See5*, o usuário deve primeiramente identificar o conjunto de dados, e então interagir com o sistema para selecionar opções de construção dos classificadores, para induzir árvores de decisão ou regras, e predizer novos casos com os classificadores construídos (Quinlan, 2002). Boosting é uma das dessas opções do sistema. Para implementar esta técnica de construção de ensembles, o algoritmo implementado foi baseado no algoritmo AdaBoost (*Adaptive Boosting*) (Quinlan, 2002), desenvolvido por Freund and Schapire, 1997), descrito anteriormente. Um ponto interessante a ser observado é que o processo de construção do ensemble continua para um determinado número de iterações L, mas pára se em uma determinada iteração um dos classificadores é extremamente preciso ($\text{erro}(\mathbf{h}_l) = 0$) ou extremamente impreciso ($\text{erro}(\mathbf{h}_l) \geq 1/2$). Este critério difere do critério de parada do Algoritmo 1 na página 9 — Algoritmo AdaBoost.M1 —, o qual somente pára se $\text{erro}(\mathbf{h}_l) \geq 1/2$. Na implementação do *See5*, no caso de parar quando $\text{erro}(\mathbf{h}_l) = 0$, a hipótese $\mathbf{h}_l$ é inserida no ensemble, e em caso de $\text{erro}(\mathbf{h}_l) \geq 1/2$, $\mathbf{h}_l$ é descartada (Quinlan, 1996).

A técnica de Boosting implementada na ferramenta *See5* foi utilizada gerando-se para cada con-

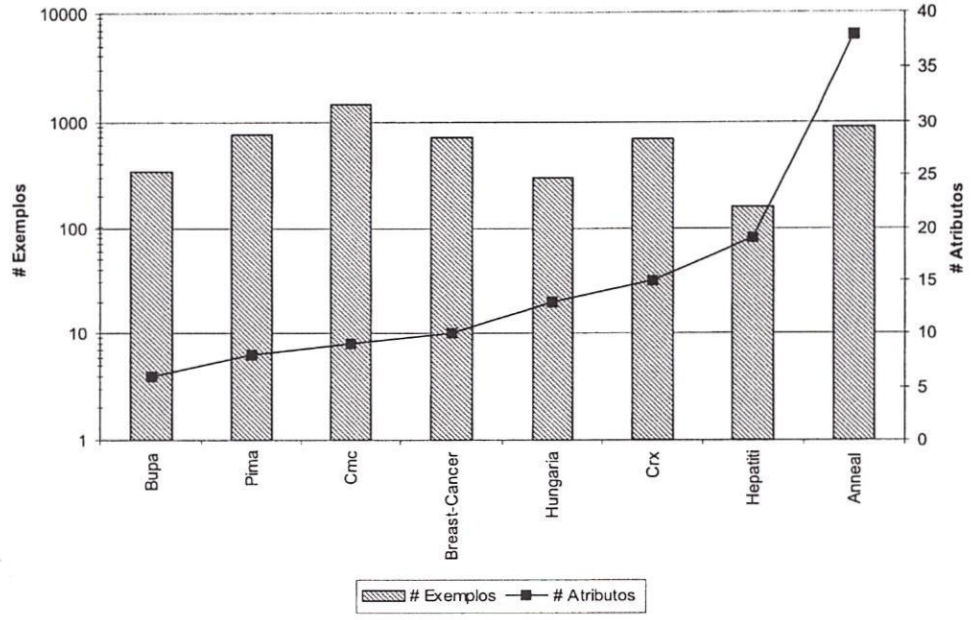

Figura 5: Dimensão dos Conjuntos de Dados

junto de dados 35 ensembles de classificadores. Os 25 primeiros ensembles consistem, respectivamente, de 1 até 25 classificadores e os outros 10 ensembles consistem de 30, 35, 40, 45, 50, 55, 60, 65, 70 e 75 classificadores respectivamente. Para cada conjunto de dados e para cada ensemble construído, foi analisado o desempenho da técnica de Boosting utilizando 10-fold Cross-validation. Os resultados obtidos são descritos a seguir.

5 Resultados Experimentais

A Tabela 3 mostra as taxas de erro (e) e o desvio padrão (dp) para cada um dos 35 ensembles gerados utilizando a técnica de Boosting e o algoritmo de indução de árvores de decisão do sistema *See5*, como algoritmo de aprendizado de máquina.

L	Bupa ($e \pm dp$)	Pima ($e \pm dp$)	CMC ($e \pm dp$)	WBC ($e \pm dp$)	Hungaria ($e \pm dp$)	CRX ($e \pm dp$)	Hepatiti ($e \pm dp$)	Anneal ($e \pm dp$)
1	34.50 ± 2.2	26.70 ± 1.0	49.90 ± 1.1	5.30 ± 0.6	21.80 ± 2.5	14.60 ± 1.0	20.10 ± 2.7	7.00 ± 0.9
2	36.00 ± 2.7	24.20 ± 1.4	49.70 ± 1.1	5.40 ± 0.7	22.10 ± 1.2	13.30 ± 1.1	16.60 ± 2.5	5.20 ± 0.8
3	32.40 ± 3.1	26.30 ± 1.8	48.70 ± 1.3	5.00 ± 0.6	18.80 ± 2.9	14.90 ± 1.8	18.80 ± 2.6	4.00 ± 0.9
4	33.70 ± 1.6	25.50 ± 1.2	49.80 ± 1.1	4.10 ± 0.7	22.10 ± 3.2	12.90 ± 1.4	18.80 ± 3.1	5.20 ± 0.7
5	30.10 ± 2.6	24.80 ± 1.4	47.50 ± 1.4	4.00 ± 0.5	23.80 ± 2.0	14.80 ± 2.0	18.80 ± 2.0	4.50 ± 0.7
6	29.00 ± 1.8	26.80 ± 1.2	49.10 ± 1.6	4.60 ± 0.9	20.10 ± 2.4	13.90 ± 1.2	16.00 ± 2.4	4.50 ± 0.5
7	30.70 ± 2.5	25.40 ± 2.1	48.40 ± 0.9	3.30 ± 0.5	18.30 ± 3.6	12.90 ± 1.2	17.40 ± 2.3	4.80 ± 0.7
8	29.80 ± 2.0	25.20 ± 1.4	50.70 ± 1.8	4.00 ± 0.4	20.40 ± 2.4	12.80 ± 1.1	15.60 ± 3.5	4.30 ± 0.6
9	33.60 ± 2.9	25.10 ± 1.6	48.80 ± 1.8	2.90 ± 0.4	19.40 ± 2.0	14.20 ± 1.2	15.50 ± 1.7	5.30 ± 0.8

continua na próxima página

continuação da página anterior

L	Bupa ($e \pm dp$)	Pima ($e \pm dp$)	CMC ($e \pm dp$)	WBC ($e \pm dp$)	Hungaria ($e \pm dp$)	CRX ($e \pm dp$)	Hepatiti ($e \pm dp$)	Anneal ($e \pm dp$)
10	27.00 ± 2.2	26.50 ± 1.4	50.40 ± 1.6	3.90 ± 0.6	22.00 ± 2.4	14.50 ± 1.1	14.70 ± 2.3	4.50 ± 0.8
11	28.40 ± 2.1	25.40 ± 1.4	50.80 ± 1.3	3.70 ± 0.8	18.30 ± 1.4	12.80 ± 1.5	16.80 ± 1.8	3.90 ± 0.5
12	27.60 ± 3.4	24.50 ± 1.3	49.00 ± 1.3	3.00 ± 0.7	19.40 ± 1.5	13.00 ± 1.0	16.80 ± 2.9	5.70 ± 1.2
13	26.70 ± 1.7	25.20 ± 1.7	49.60 ± 1.3	3.10 ± 0.7	22.40 ± 2.6	12.80 ± 1.6	17.40 ± 2.5	6.20 ± 0.5
14	27.80 ± 2.4	24.70 ± 1.0	48.70 ± 1.6	3.30 ± 0.5	22.10 ± 1.5	13.50 ± 1.5	14.20 ± 2.3	5.00 ± 0.9
15	29.00 ± 2.7	26.10 ± 1.4	51.10 ± 1.3	3.70 ± 0.6	19.00 ± 2.9	13.00 ± 2.1	16.30 ± 3.1	5.00 ± 0.6
16	28.40 ± 2.1	24.40 ± 1.4	49.60 ± 1.0	3.40 ± 0.8	20.40 ± 2.9	13.80 ± 1.1	16.00 ± 3.4	5.30 ± 0.8
17	29.80 ± 2.2	24.50 ± 1.4	48.50 ± 1.1	3.30 ± 0.6	20.40 ± 2.5	14.90 ± 1.2	16.20 ± 2.9	5.60 ± 0.9
18	29.00 ± 2.2	25.50 ± 1.3	50.60 ± 1.0	3.60 ± 0.6	20.70 ± 2.4	13.60 ± 0.9	16.80 ± 4.3	5.50 ± 0.6
19	29.30 ± 2.7	24.80 ± 1.6	49.70 ± 1.3	3.70 ± 0.7	20.80 ± 1.6	12.50 ± 1.4	16.00 ± 2.3	5.10 ± 0.3
20	29.90 ± 2.7	24.80 ± 1.0	49.00 ± 1.3	3.30 ± 0.7	21.40 ± 1.5	14.50 ± 1.6	12.90 ± 2.4	4.60 ± 0.8
21	25.50 ± 2.2	24.50 ± 2.0	49.80 ± 1.5	3.10 ± 0.5	20.10 ± 1.3	13.90 ± 1.7	15.50 ± 3.1	5.60 ± 0.7
22	27.50 ± 2.0	25.20 ± 1.3	49.80 ± 1.1	3.70 ± 0.6	21.50 ± 3.6	13.90 ± 1.5	17.40 ± 2.3	5.80 ± 0.8
23	28.40 ± 1.7	25.90 ± 1.4	49.90 ± 0.6	2.90 ± 0.7	22.10 ± 1.9	14.30 ± 1.8	16.20 ± 1.5	5.00 ± 0.8
24	28.70 ± 1.8	25.00 ± 0.9	48.30 ± 1.2	3.70 ± 0.8	19.70 ± 2.6	12.80 ± 1.4	16.80 ± 3.6	5.60 ± 0.7
25	25.50 ± 1.5	26.40 ± 1.7	50.90 ± 1.1	3.60 ± 0.8	21.80 ± 1.9	13.20 ± 1.4	17.40 ± 1.9	5.90 ± 0.4
30	29.30 ± 2.6	25.20 ± 1.3	49.70 ± 1.9	3.90 ± 0.8	20.80 ± 2.3	12.50 ± 1.4	17.60 ± 3.2	4.60 ± 0.7
35	26.40 ± 2.5	22.60 ± 0.8	49.80 ± 0.8	3.30 ± 0.8	20.80 ± 2.7	13.00 ± 1.9	18.80 ± 2.6	4.60 ± 0.7
40	26.40 ± 2.5	25.60 ± 1.0	49.20 ± 1.4	2.70 ± 0.7	18.80 ± 2.0	12.80 ± 0.9	16.00 ± 1.9	4.60 ± 0.7
45	25.20 ± 1.4	24.10 ± 1.5	49.10 ± 1.3	3.00 ± 0.7	21.10 ± 1.6	12.50 ± 0.8	14.20 ± 1.9	4.30 ± 0.7
50	29.00 ± 1.7	24.40 ± 1.7	50.70 ± 1.6	2.90 ± 0.8	20.00 ± 1.6	12.50 ± 1.3	15.40 ± 2.3	5.20 ± 0.8
55	29.90 ± 1.4	24.70 ± 0.9	48.30 ± 0.7	3.70 ± 0.6	20.10 ± 2.0	13.60 ± 1.5	14.30 ± 2.7	5.80 ± 0.9
60	25.20 ± 2.4	25.00 ± 2.3	48.60 ± 1.3	3.40 ± 0.7	19.40 ± 1.9	12.50 ± 1.0	15.50 ± 2.2	4.30 ± 0.5
65	27.20 ± 2.5	25.10 ± 1.1	49.50 ± 1.5	3.00 ± 0.4	20.10 ± 2.2	12.60 ± 1.6	14.80 ± 1.9	4.20 ± 0.7
70	25.50 ± 1.5	26.30 ± 0.9	50.20 ± 1.6	2.70 ± 0.7	22.10 ± 2.1	13.20 ± 1.8	14.10 ± 2.2	4.80 ± 0.6
75	26.40 ± 1.9	22.80 ± 1.4	48.70 ± 1.3	3.30 ± 0.7	19.40 ± 2.6	14.20 ± 1.4	18.00 ± 2.0	5.30 ± 0.9

Tabela 3: Resultados Obtidos utilizando Árvores de Decisão

A Tabela 4 é similar mas utilizando o algoritmo de indução de regras do sistema $\mathcal{S}$ee5 como algoritmo de aprendizado de máquina.

L	Bupa ($e \pm dp$)	Pima ($e \pm dp$)	CMC ($e \pm dp$)	WBC ($e \pm dp$)	Hungaria ($e \pm dp$)	CRX ($e \pm dp$)	Hepatiti ($e \pm dp$)	Anneal ($e \pm dp$)
1	33.60 ± 2.3	26.10 ± 0.8	48.80 ± 1.1	4.30 ± 0.6	21.40 ± 2.6	14.50 ± 1.2	19.50 ± 1.8	5.90 ± 0.7
2	33.70 ± 2.5	25.70 ± 1.0	48.50 ± 0.7	5.60 ± 0.9	21.80 ± 1.8	14.20 ± 1.5	16.60 ± 3.1	4.50 ± 0.9
3	31.50 ± 2.6	24.80 ± 1.3	47.30 ± 1.2	4.60 ± 0.6	19.80 ± 2.6	14.50 ± 1.6	17.40 ± 2.3	3.70 ± 0.7
4	32.50 ± 2.3	25.40 ± 0.9	48.20 ± 0.9	4.10 ± 0.8	20.40 ± 2.8	12.30 ± 1.3	18.20 ± 3.0	3.60 ± 0.5
5	29.60 ± 2.2	25.10 ± 1.6	47.80 ± 0.9	3.40 ± 0.5	20.40 ± 1.7	14.90 ± 1.5	18.80 ± 2.5	4.00 ± 0.8
6	29.20 ± 2.1	27.10 ± 1.0	47.30 ± 1.5	3.70 ± 0.9	19.40 ± 2.3	14.10 ± 1.1	14.10 ± 2.3	3.10 ± 0.5
7	29.00 ± 2.3	24.10 ± 1.9	46.20 ± 1.1	3.10 ± 0.5	21.40 ± 3.6	12.30 ± 1.0	19.30 ± 1.9	3.00 ± 0.5
8	29.20 ± 1.5	23.30 ± 1.5	48.00 ± 1.4	3.70 ± 0.6	19.40 ± 1.8	13.90 ± 1.4	17.00 ± 3.6	3.10 ± 0.4
9	31.30 ± 3.2	25.40 ± 1.5	46.30 ± 1.6	3.00 ± 0.4	19.40 ± 1.5	13.30 ± 1.1	14.20 ± 1.6	3.60 ± 0.7
10	25.00 ± 2.5	24.40 ± 1.1	46.00 ± 1.4	3.60 ± 0.7	19.30 ± 2.1	14.10 ± 1.1	14.70 ± 2.5	3.40 ± 0.7
11	27.60 ± 2.0	23.80 ± 1.3	47.90 ± 1.2	3.60 ± 0.6	17.00 ± 1.9	12.90 ± 1.0	16.10 ± 1.8	3.30 ± 0.6
12	25.90 ± 3.0	25.00 ± 1.1	46.80 ± 1.3	3.00 ± 0.7	20.50 ± 1.7	12.80 ± 1.1	17.50 ± 3.0	3.00 ± 0.6
13	26.40 ± 2.1	23.50 ± 1.4	47.50 ± 1.0	3.60 ± 0.9	20.70 ± 2.7	11.90 ± 1.3	18.00 ± 2.4	3.10 ± 0.4
14	25.50 ± 2.4	22.80 ± 1.4	46.70 ± 1.4	3.30 ± 0.6	19.70 ± 1.6	13.90 ± 1.5	15.50 ± 2.2	2.80 ± 0.4
15	27.50 ± 2.1	26.30 ± 1.4	47.60 ± 1.6	3.30 ± 0.7	18.40 ± 3.3	12.60 ± 0.1	15.60 ± 2.9	2.60 ± 0.6
16	28.10 ± 1.2	24.70 ± 1.2	46.20 ± 1.2	3.30 ± 0.8	19.40 ± 2.5	12.80 ± 1.2	16.70 ± 3.3	3.30 ± 0.5
17	29.60 ± 2.0	24.20 ± 1.6	45.80 ± 1.3	3.30 ± 0.7	17.30 ± 1.6	13.60 ± 1.1	16.80 ± 3.1	3.10 ± 0.6
18	26.10 ± 2.3	25.00 ± 1.1	48.30 ± 0.9	3.40 ± 0.6	18.00 ± 1.6	12.80 ± 0.7	16.80 ± 4.3	3.20 ± 0.3
19	27.00 ± 2.8	25.10 ± 1.4	47.90 ± 1.4	3.30 ± 0.6	19.10 ± 1.8	12.80 ± 1.3	17.30 ± 2.5	3.10 ± 0.4
20	29.30 ± 2.4	23.70 ± 0.9	47.30 ± 0.9	3.00 ± 0.6	20.10 ± 1.6	14.10 ± 1.8	14.90 ± 2.5	3.20 ± 0.4
21	26.10 ± 2.0	23.50 ± 2.0	46.90 ± 1.0	3.00 ± 0.5	19.40 ± 2.0	13.50 ± 1.4	15.50 ± 2.9	3.60 ± 0.5
22	27.80 ± 2.3	25.00 ± 1.0	47.90 ± 1.3	3.00 ± 0.6	20.50 ± 2.9	15.10 ± 1.5	18.10 ± 1.8	3.30 ± 0.5

continua na próxima página

continuação da página anterior

L	Bupa ($e \pm dp$)	Pima ($e \pm dp$)	CMC ($e \pm dp$)	WBC ($e \pm dp$)	Hungaria ($e \pm dp$)	CRX ($e \pm dp$)	Hepatiti ($e \pm dp$)	Anneal ($e \pm dp$)
23	27.00 ± 2.1	25.10 ± 1.2	46.80 ± 0.6	3.00 ± 0.5	19.80 ± 2.0	13.90 ± 1.4	16.10 ± 1.7	3.70 ± 0.4
24	26.60 ± 2.3	23.50 ± 1.1	45.80 ± 1.5	3.40 ± 0.8	19.10 ± 3.1	12.60 ± 1.5	16.00 ± 2.7	3.70 ± 0.5
25	23.50 ± 1.3	26.00 ± 1.4	49.00 ± 1.4	3.30 ± 0.8	20.10 ± 1.9	13.20 ± 1.2	16.10 ± 1.7	3.50 ± 0.4
30	28.40 ± 2.2	24.10 ± 1.0	47.00 ± 1.8	3.60 ± 0.7	19.40 ± 1.8	12.20 ± 1.0	16.90 ± 3.0	3.20 ± 0.5
35	27.00 ± 2.1	23.70 ± 0.7	46.10 ± 0.7	3.30 ± 0.8	19.80 ± 2.7	13.00 ± 1.9	17.50 ± 2.7	3.10 ± 0.5
40	25.20 ± 2.7	23.90 ± 1.1	47.90 ± 1.4	2.70 ± 0.7	17.80 ± 2.2	11.70 ± 1.2	17.40 ± 1.6	2.80 ± 0.6
45	24.30 ± 1.9	24.10 ± 1.4	46.00 ± 1.4	2.70 ± 0.6	20.70 ± 2.0	11.90 ± 0.8	16.80 ± 1.4	3.00 ± 0.4
50	28.40 ± 1.6	23.90 ± 1.6	47.60 ± 1.5	2.90 ± 0.7	18.70 ± 1.7	12.50 ± 1.4	16.60 ± 2.3	3.00 ± 0.4
55	28.70 ± 1.8	25.00 ± 1.1	46.80 ± 1.2	3.10 ± 0.6	18.30 ± 2.1	13.20 ± 1.5	13.60 ± 2.5	3.80 ± 0.9
60	25.20 ± 3.0	24.10 ± 1.6	47.20 ± 1.2	3.10 ± 0.8	18.70 ± 1.9	12.20 ± 1.1	16.20 ± 2.2	3.60 ± 0.4
65	27.20 ± 2.6	24.30 ± 1.1	47.00 ± 1.2	3.00 ± 0.4	19.40 ± 2.0	12.30 ± 1.8	16.00 ± 1.6	2.90 ± 0.5
70	25.80 ± 1.6	24.60 ± 1.2	46.60 ± 1.3	3.00 ± 0.6	20.40 ± 2.1	12.60 ± 1.6	14.00 ± 2.2	2.80 ± 0.4
75	26.40 ± 1.5	24.30 ± 1.4	47.00 ± 1.3	3.10 ± 0.6	19.40 ± 2.6	14.10 ± 1.5	16.70 ± 2.7	3.20 ± 0.4

Tabela 4: Resultados Obtidos utilizando Regras

Os resultados das Tabelas 3 e 4 são também mostrados graficamente para cada conjunto de dados, cada algoritmo de aprendizado de máquina e cada ensemble gerado utilizando a técnica de Boosting. No eixo das ordenadas de cada gráfico, tem-se o número de classificadores de cada ensemble, e no eixo das abcissas, tem-se as respectivas taxas de erro. Os gráficos são mostrados nas Figuras 6, 7, 8, 9, 10, 11, 12 e 13.

Deve ser observado que um dos interesses futuros deste trabalho está relacionado com a melhoria da precisão de ensembles quando são utilizados poucos classificadores para compor o ensemble. A nossa idéia é utilizar ensembles cujos classificadores têm seu conhecimento expresso em uma linguagem simbólica, tais como regras, a fim de selecionar as melhores regras, ou combinação de regras, de cada classificador no ensemble, para fornecer uma explicação da predição de um novo exemplo $\mathbf{x}$. Dessa forma, um ensemble de classificadores não agiria como uma caixa preta. Entretanto, para conduzir experimentos controláveis e pela teoria de ensembles mostrar que poucos classificadores oferecem uma melhora na predição comparado a um único classificador, será utilizado um baixo número de classificadores para compor o ensemble. Assim, para melhor focalizar esse problema, gráficos com as taxas de erros dos ensembles com 1 até 7 classificadores são mostrados na Figura correspondente.

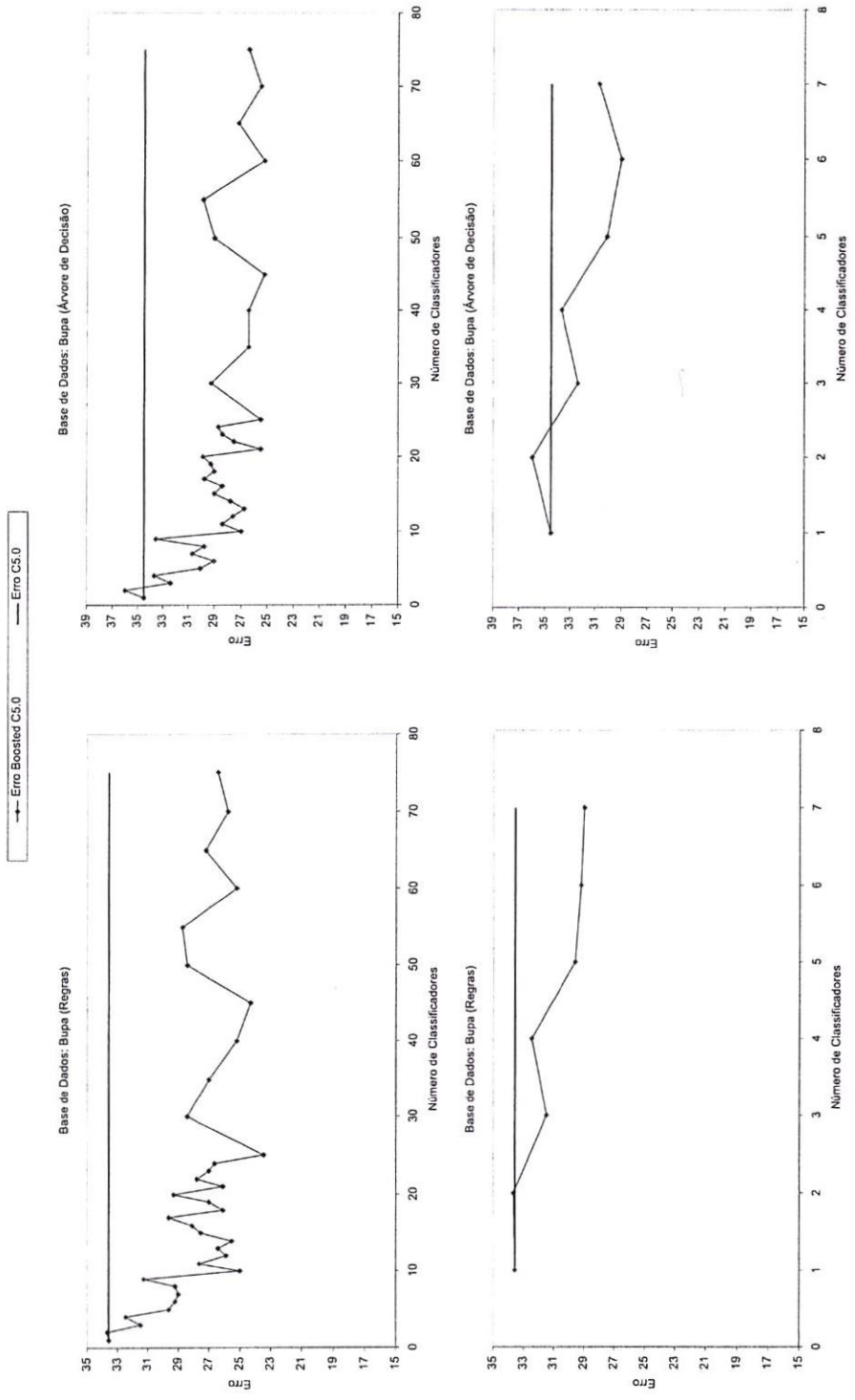

Figura 6: Bupa: Taxas de Erros de Ensembles Gerados.

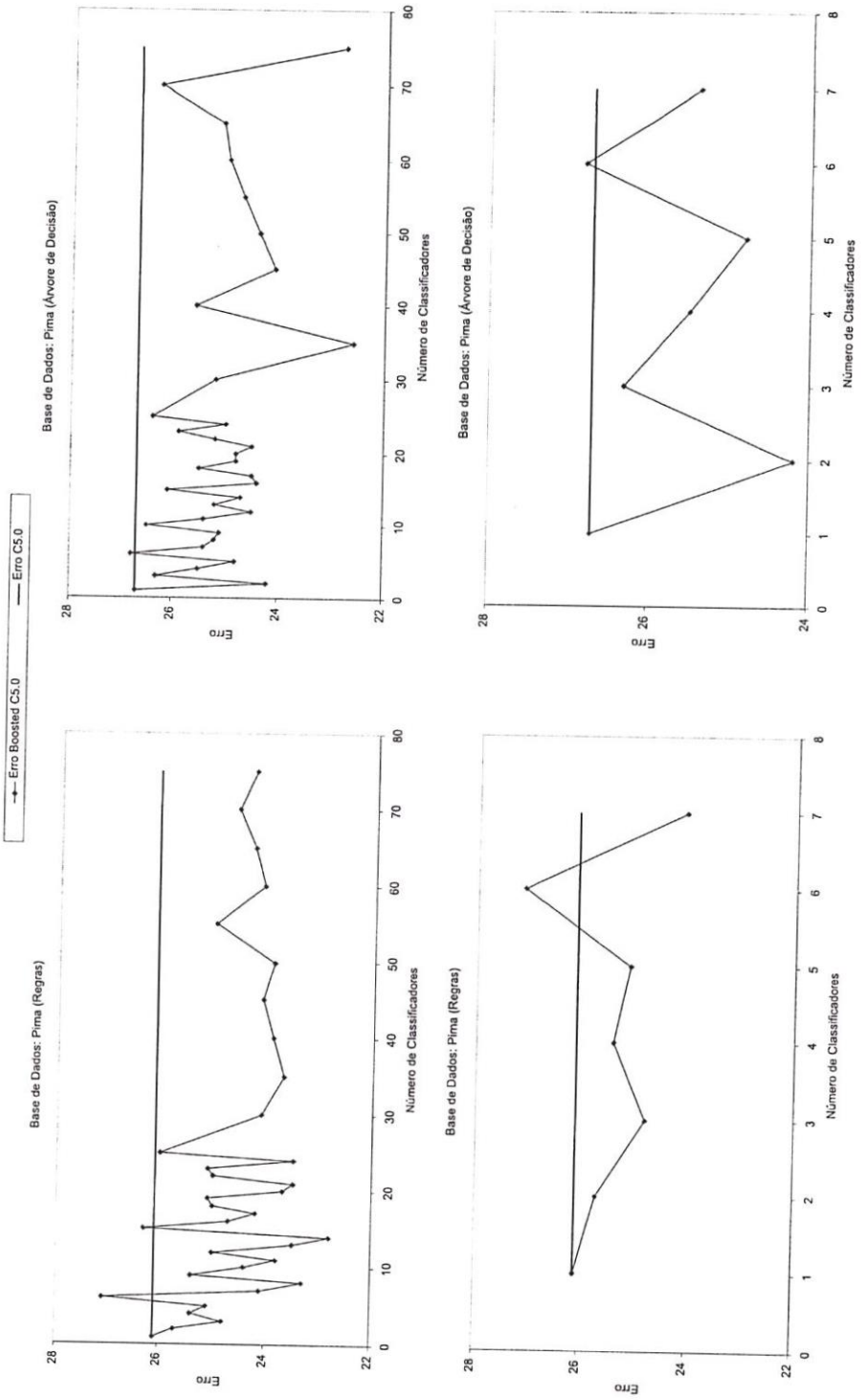

Figura 7: Pima: Taxas de Erros de Ensembles Gerados.

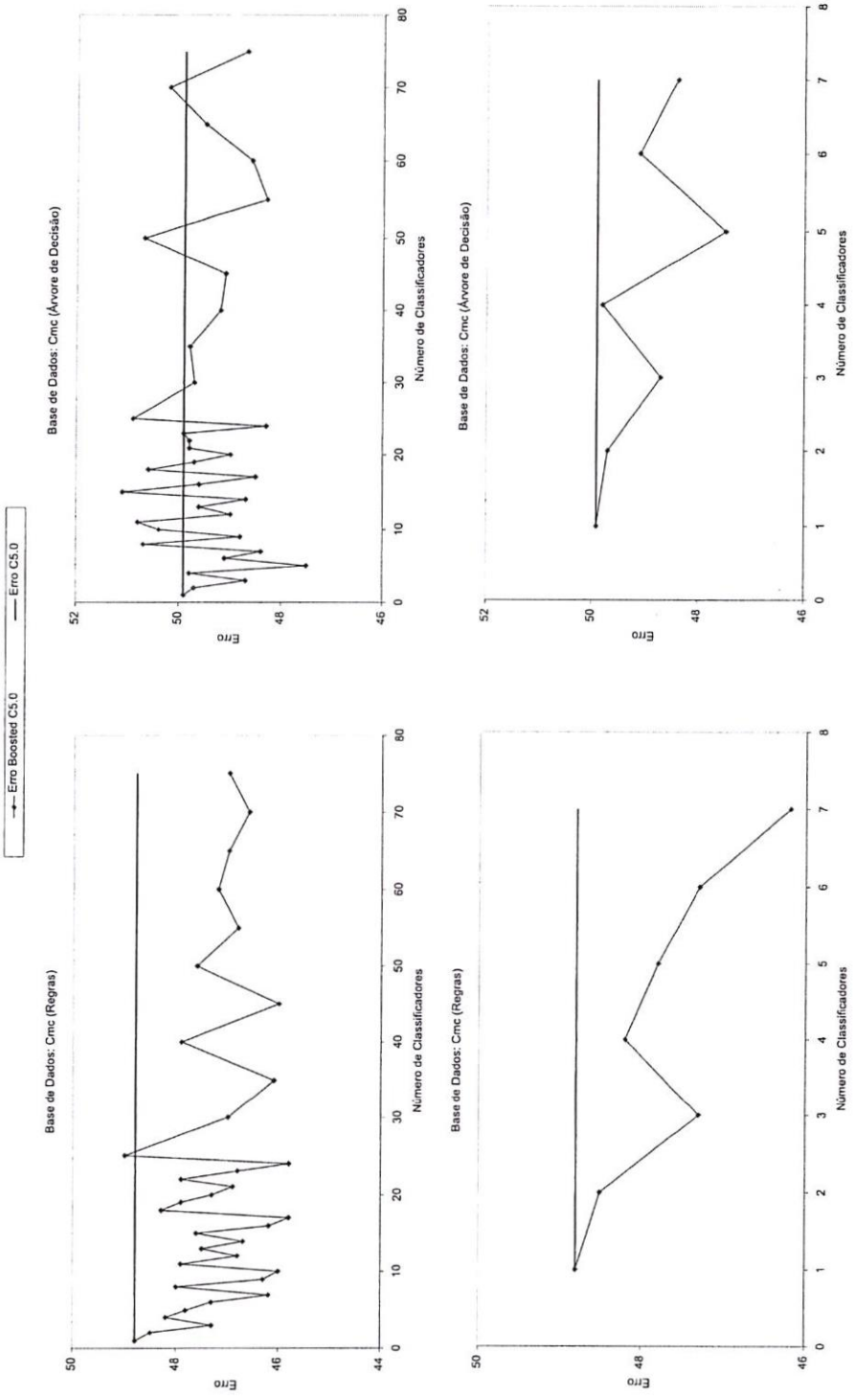

Figura 8: CMC: Taxas de Erros de Ensembles Gerados.

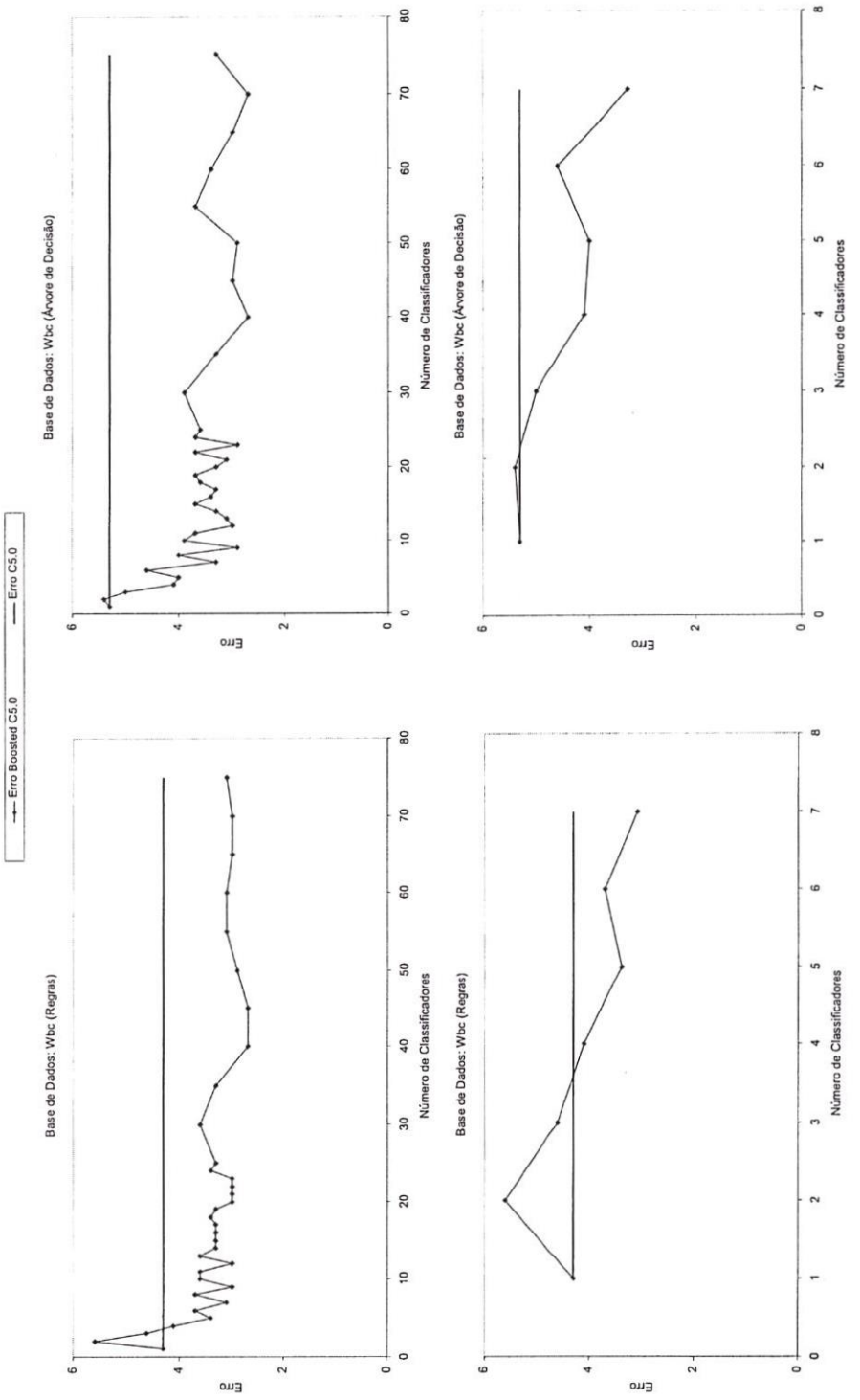

Figura 9: WBC: Taxas de Erros de Ensembles Gerados.

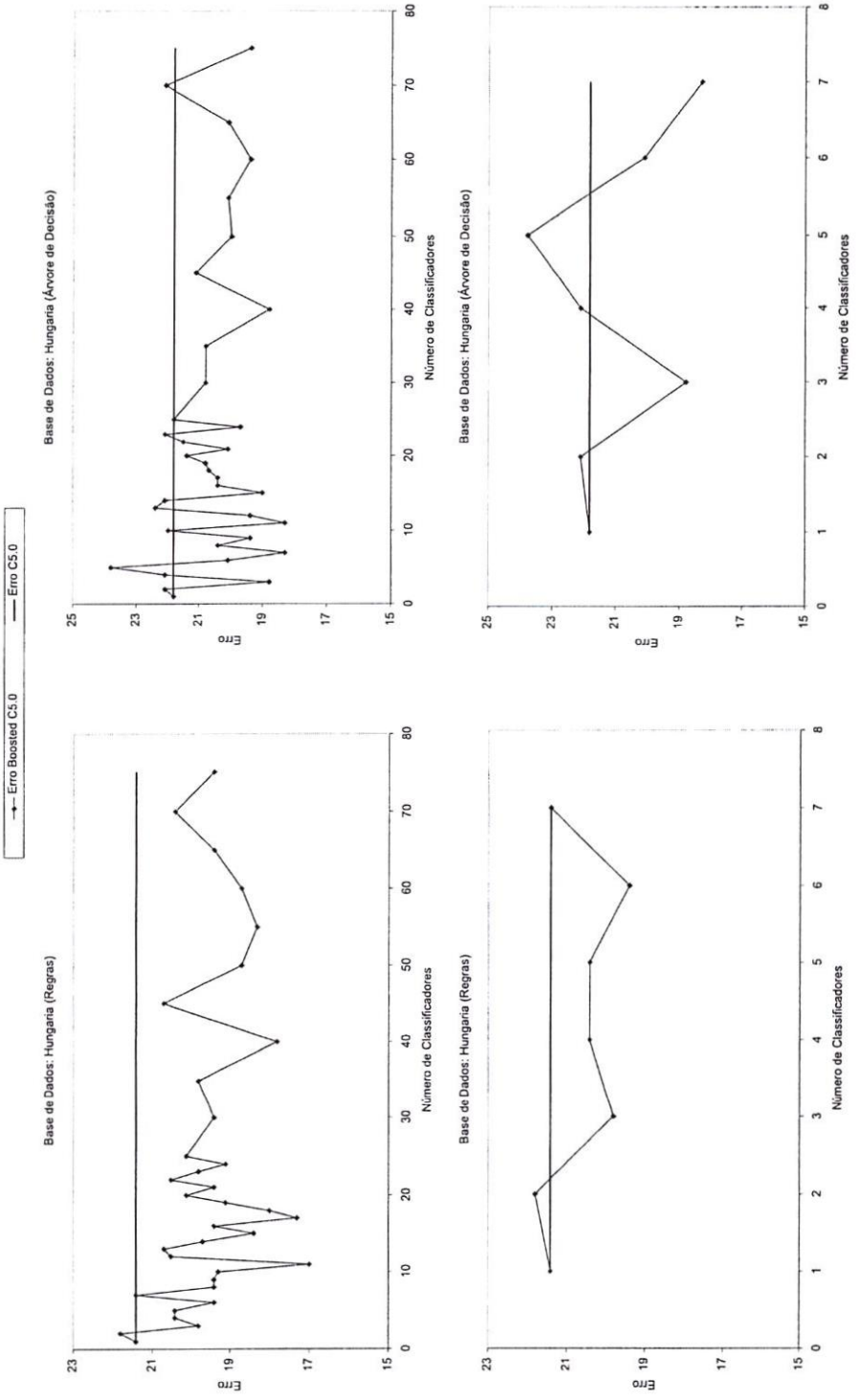

Figura 10: Hungaria: Taxas de Erros de Ensembles Gerados.

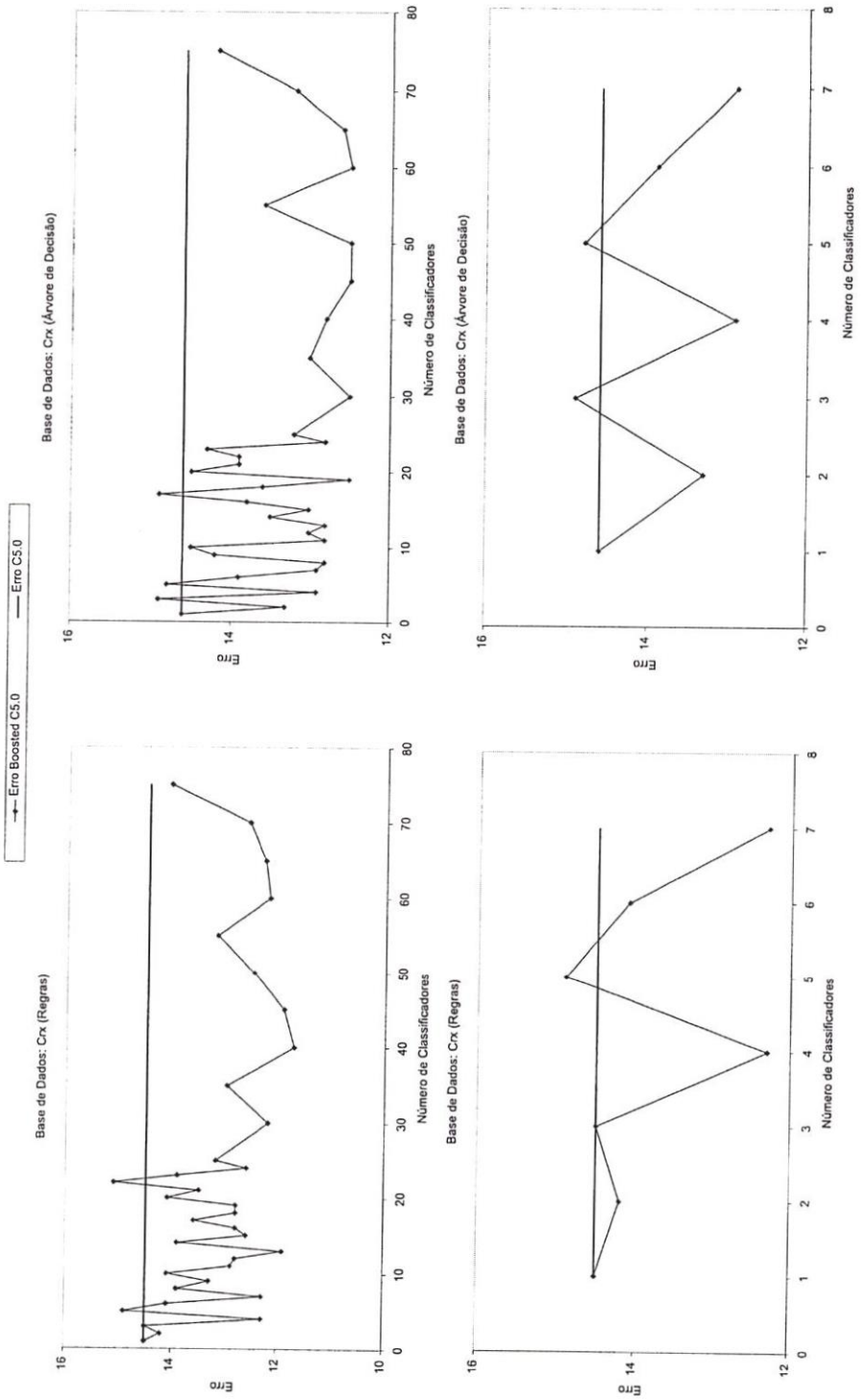

Figura 11: CRX: Taxas de Erros de Ensembles Gerados.

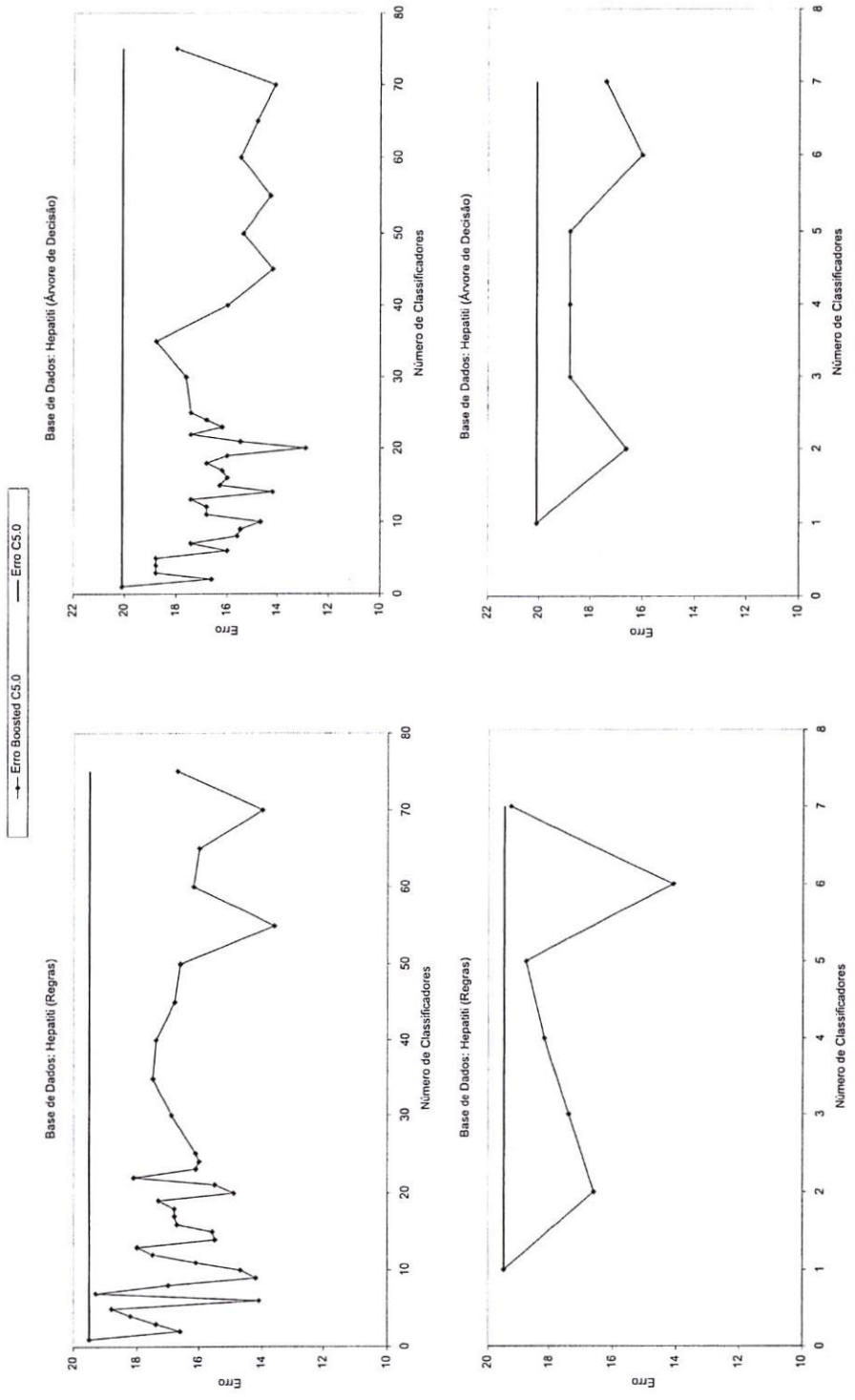

Figura 12: Hepatiti: Taxas de Erros de Ensembles Gerados.

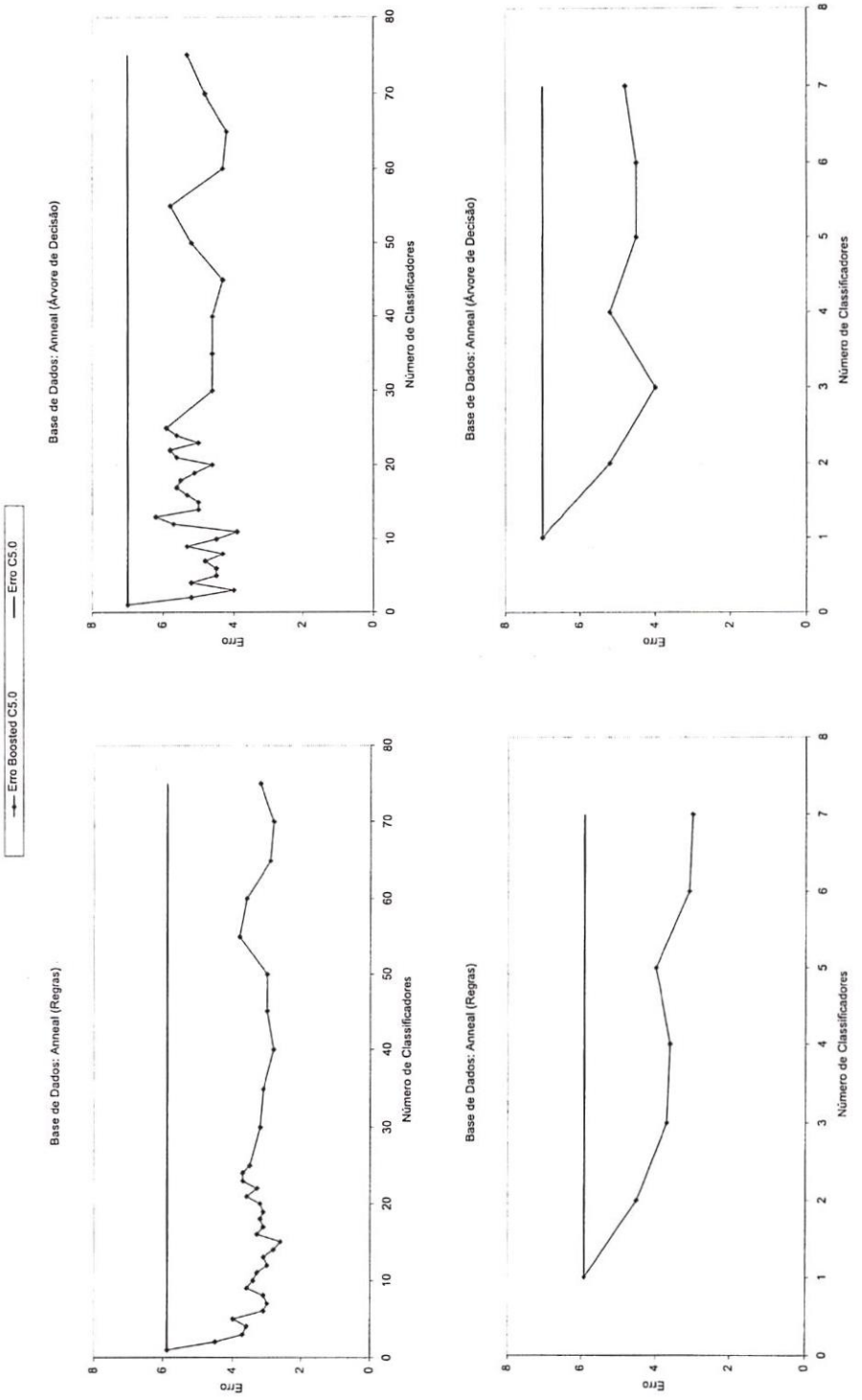

Figura 13: Anneal: Taxas de Erros de Ensembles Gerados.

6 Análise dos Resultados

Primeiramente, é interessante observar que o algoritmo que induz regras tem, em todos os conjuntos de dados, um erro médio menor que o algoritmo que induz árvores de decisão, quando utilizado independentemente, isto é, para $L = 1$. A fim de facilidar a visualização desses resultados, a Figura 14 mostra os valores correspondentes a $L = 1$ das Tabelas 3 e 4. Para verificar se o algoritmo que induz regras supera o que induz árvores de decisão significativamente, com grau de confiança de 95%, utilizamos o critério de diferença absoluta (da) em desvios padrões (Baranauskas and Monard, 2000) para cada conjunto de dados usando regras (A_R) e usando árvores de decisão (A_{AD}), dado pela Equação 4.

$$da(A_R - A_{AD}) = \frac{e_R - e_{AD}}{\sqrt{\frac{(dp_R)^2 + (dp_{AD})^2}{2}}} \tag{4}$$

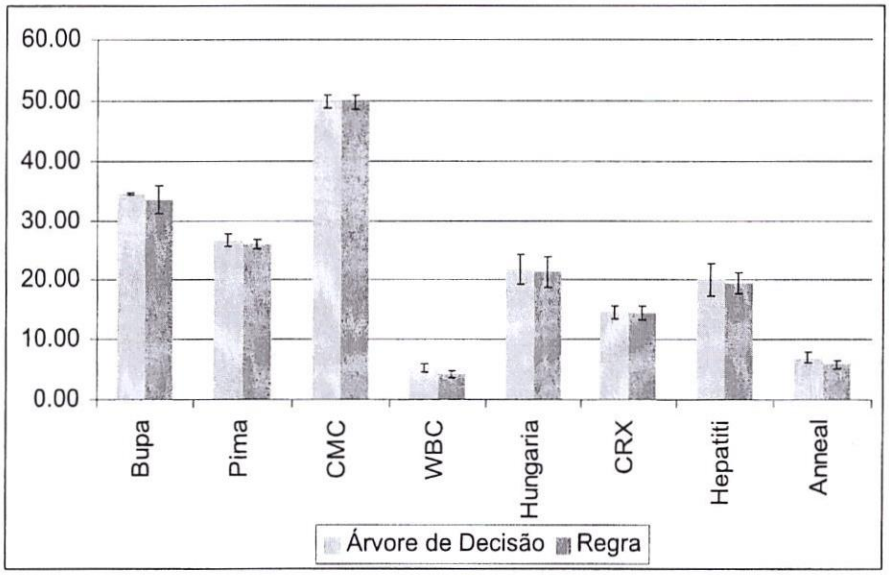

Figura 14: Resultados Obtidos para L=1 Usando Árvores de Decisão e Regras

Se $da(A_R - A_{AD}) \leq 2$ então o algoritmo que induz regras supera o que induz árvores de decisão com grau de confiança de 95%. Como pode ser observado na Tabela 5, isto não ocorre para qualquer um dos conjuntos de dados, isto é, não se pode afirmar que o algoritmo do *See5* que induz regras supera o algoritmo que induz árvores de decisão com 95% de confiança.

Também, deve ser observado que Boosting pára se um dos classificadores obtém erro zero no conjunto de treinamento. Isto significa que este classificador tem o máximo poder de voto e, na realidade, é usado como um classificador único. Por outro lado, se o algoritmo de boosting pára se um dos classificadores obtém erro maior que ou igual a 50% no conjunto de treinamento, o ensemble terá uma alta taxa de erro, pois isso significa que as taxas de erro dos classificadores anteriores foram aumentando gradativamente. A Tabela 6 mostra o número de vezes que Boosting

Conjunto de Dados	AD $(e \pm dp)$	R $(e \pm dp)$	da$(A_R - A_{AD})$
Bupa	34.50 ± 0.2	33.60 ± 2.3	-0.55
Pima	26.70 ± 1.0	26.10 ± 0.8	-0.66
CMC	49.90 ± 1.1	49.80 ± 1.1	-0.09
WBC	5.30 ± 0.6	4.30 ± 0.6	-1.67
Hungaria	21.80 ± 2.5	21.40 ± 2.6	-0.16
CRX	14.60 ± 1.0	14.50 ± 1.2	-0.09
Hepatiti	20.10 ± 2.7	19.50 ± 1.8	-0.26
Anneal	7.00 ± 0.9	5.90 ± 0.7	-1.36

Tabela 5: Resultados Obtidos no Critério de Diferença Absoluta Entre o Algoritmo que Induz Árvores de Decisão e o que Induz Regras

parou antes de construir os L classificadores das 10 vezes que foi executado para cada L, durante a execução do 10-fold cross-validation utilizando Árvores de Decisão (AD) e Regras (R). As regras foram geradas a partir das árvores de decisão e, portanto, se parou para árvores de decisão, também parou para as regras. Uma ilustração dos dados mostrados na Tabela 6 pode ser vista na Figura 15.

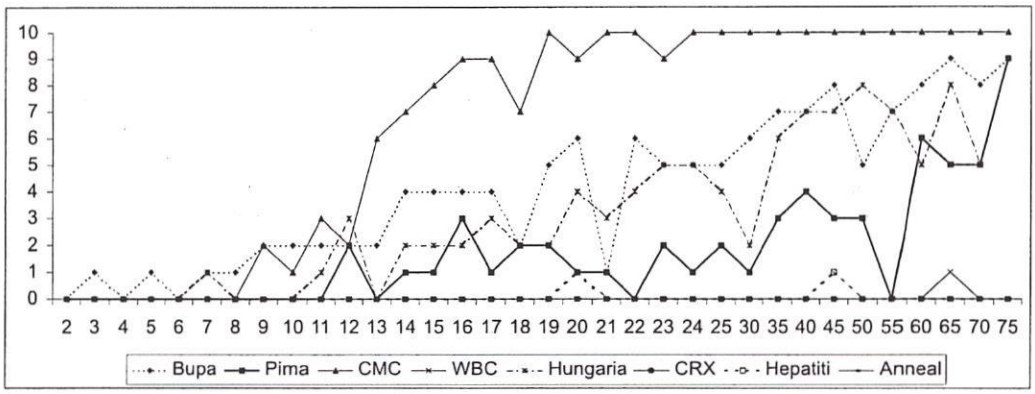

Figura 15: Número de Vezes que o Algoritmo de Boosting Parou Antes de Construir L Classificadores Durante a Execução do 10-Fold Cross-Validation

Observando a Figura 15, pode-se concluir que, para a construção de ensembles de classificadores utilizando a técnica de boosting implementada no *See5* e os conjuntos de dados já descritos, somente ensembles com até 8 classificadores têm 90% de chance de ter seus L classificadores construídos. Para $L > 8$, nada se pode afirmar. Deve ser lembrado que o algoritmo de boosting utilizado pára antes de construir as L hipóteses quando o algoritmo de AM induz uma hipótese com taxa de erro maior que ou igual a 50% ou com taxa de erro igual a 0%.

Analisando as Figuras 6 a 13 e a Tabela 6, pode-se afirmar que há uma queda na taxa de erro quanto maior o valor de L, conforme o esperado, para os conjuntos de dados Bupa, WBC e Anneal (Figuras 6, 9 e 13, respectivamente). Para os outros conjuntos de dados, observa-se máximos e mínimos locais que não permitem afirmar nada a respeito de melhorias na predição.

Para os conjuntos de dados onde os resultados não são os esperados, pensou-se na hipótese de o

L	Bupa	Pima	CMC	WBC	Hungaria	CRX	Hepatiti	Anneal
2	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	1	0	0	0
8	1	0	0	0	0	0	0	0
9	2	0	2	0	0	0	0	0
10	2	0	1	0	0	0	0	0
11	2	0	3	0	1	0	0	0
12	2	2	2	0	3	0	0	0
13	2	0	6	0	0	0	0	0
14	4	1	7	0	2	0	0	0
15	4	1	8	0	2	0	0	0
16	4	3	9	0	2	0	0	0
17	4	1	9	0	3	0	0	0
18	2	2	7	0	2	0	0	0
19	5	2	10	0	2	0	0	0
20	6	1	9	0	4	0	1	0
21	1	1	10	0	3	0	0	0
22	6	0	10	0	4	0	0	0
23	5	2	9	0	5	0	0	0
24	5	1	10	0	5	0	0	0
25	5	2	10	0	4	0	0	0
30	6	1	10	0	2	0	0	0
35	7	3	10	0	6	0	0	0
40	7	4	10	0	7	0	0	0
45	8	3	10	0	7	0	1	0
50	5	3	10	0	8	0	0	0
55	7	0	10	0	7	0	0	0
60	8	6	10	0	5	0	0	0
65	9	5	10	1	8	0	0	0
70	8	5	10	0	5	0	0	0
75	9	9	10	0	9	0	0	0

Tabela 6: Número de Vezes que o Algoritmo de Boosting Parou Antes de Construir L Classificadores Durante a Execução do 10-Fold Cross-Validation

algoritmo estar criando, muitas vezes, menos que L classificadores nos ensembles requeridos, ora por obter uma hipótese com taxa de erro igual a 0(zero), ora por obter uma hipótese com taxa de erro maior que ou igual a 50%. A partir das Figuras que se referem a esses conjuntos de dados (Figuras 7, 8, 10, 11 e 12) e da Tabela 6, pode-se observar que:

1. Para o conjunto de dados Pima (Figura 7), isso é verdade somente para $L \geq 12$; para $L < 12$, sempre são construídos os L classificadores requeridos;

2. Quanto ao conjunto de dados CMC, há muita oscilação nas taxas de erro, o que pode estar sendo provocado pelo alto ruído (7,81% de exemplos duplicados ou conflitantes — Tabela 2) pois, segundo (Bauer and Kohavi, 1999), boosting não trabalha bem com ruído. Outra explicação também seria as muitas vezes que o algoritmo de boosting não consegue construir os L classificadores requeridos;

3. Os experimentos com os conjuntos de dados CRX e Hepatiti apresentaram grandes oscilações nas taxas de erro obtidas, porém o algoritmo conseguiu construir todos os L classificadores de cada ensemble gerado em praticamente todas as iterações do 10-fold cross-validation.

Para tentar explicar as oscilações que não são explicáveis através do fato de o algoritmo parar de construir hipóteses e não construir, assim, os L classificadores nas iterações do 10-fold cross-validation, olhando para os resultados obtidos, verificou-se a existência de avisos, dizendo que o poder de predição de alguns ensembles criados nas iterações do 10-fold cross-validation pode ser danoso. Na documentação do *See5* (Quinlan, 2002), nada foi encontrado a respeito desses avisos, ou seja, sob qual circunstância esses avisos são emitidos. Foi deduzido que esse aviso é emitido quando o ensemble gerado não é um bom ensemble para se utilizar em predições. A Tabela 7 mostra o número de ensembles gerados nas 10 iterações do 10-fold cross-validation que foram considerados danosos para o *See5* para cada conjunto de dados e para cada L. Esses valores são ilustrados na Figura 16.

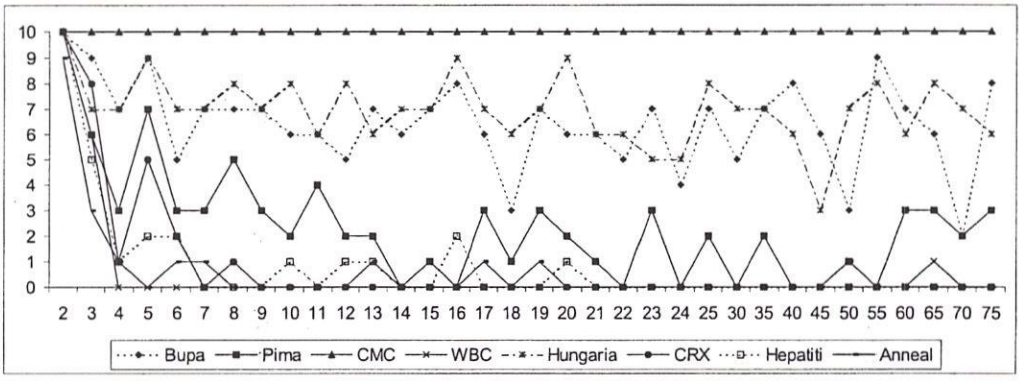

Figura 16: Número de Ensembles Danosos Gerados Durante a Execução do 10-Fold Cross-Validation

Baseando-se nas Figuras 6 a 13 e na Tabela 7, pode-se afirmar que:

1. Todos os ensembles criados com 2 classificadores são danosos. Esse resultado é esperado porque tomar decisão de classificação com apenas 2(dois) classificadores pode implicar em empates a serem resolvidos randomicamente ou resolver classificando o novo exemplo com a classe que mais possui exemplos no domínio considerado;

2. Não há relação entre as oscilações nas taxas de erro e os ensembles serem possivelmente danosos para futuras predições.

Segundo (Quinlan, 2002), após alguns testes realizados, na média um ensemble com 10 classificadores construídos com o algoritmo de boosting implementado reduz a taxa de erro nos conjuntos de teste cerca de 25%. Para verificar esse dado, foram tomados os valores das taxas de erro obtidos para $L = 1$ e para $L = 10$, sem considerar o desvio padrão, e foi calculada a porcentagem de

L	Bupa	Pima	CMC	WBC	Hungaria	CRX	Hepatiti	Anneal
2	10	10	10	10	10	10	10	9
3	9	6	10	6	7	8	5	3
4	7	3	10	0	7	1	1	1
5	9	7	10	0	9	5	2	0
6	5	3	10	0	7	2	2	1
7	7	3	10	0	7	0	0	1
8	7	5	10	0	8	1	0	0
9	7	3	10	0	7	0	0	0
10	6	2	10	0	8	0	1	0
1	6	4	10	0	6	0	0	0
12	5	2	10	0	8	0	1	0
13	7	2	10	0	6	0	1	1
14	6	0	10	0	7	0	0	0
15	7	1	10	0	7	0	0	0
16	8	0	10	0	9	0	2	0
17	6	3	10	0	7	0	0	1
18	3	1	10	0	6	0	0	0
19	7	3	10	0	7	0	0	1
20	6	2	10	0	9	0	1	0
21	6	1	10	0	6	0	0	0
22	5	0	10	0	6	0	0	0
23	7	3	10	0	5	0	0	0
24	4	0	10	0	5	0	0	0
25	7	2	10	0	8	0	0	0
30	5	0	10	0	7	0	0	0
35	7	2	10	0	7	0	0	0
40	8	0	10	0	6	0	0	0
45	6	0	10	0	3	0	0	0
50	3	1	10	0	7	0	0	1
55	9	0	10	0	8	0	0	0
60	7	3	10	0	6	0	0	0
65	6	3	10	1	8	0	0	0
70	2	2	10	0	7	0	0	0
75	8	3	10	0	6	0	0	0

Tabela 7: Número de Ensembles Danosos Gerados Durante a Execução do 10-Fold Cross-Validation

queda da taxa de erro do ensemble com 1 classificador (ou seja, um classificador único) para a taxa de erro do ensemble com 10 classificadores. Os valores obtidos são mostrados na Tabela 8.

Observando a Tabela 8, pode-se afirmar que nem sempre há uma diminuição na taxa de erro de 25% quando se constrói um ensemble com 10 classificadores.

Por outro lado, quanto à idéia de construir ensembles com reduzido número de classificadores, em relação à técnica implementada, observa-se, na Tabela 8, que há geralmente uma redução na taxa de erro, mesmo para um pequeno número de classificadores. O problema encontrado se deu nas grandes oscilações das taxas de erro de uma forma geral.

Conjunto de Dados	% Queda Árvores de Decisão	% Queda Regras
Bupa	21.74%	25.60%
Pima	0.75%	6.51%
CMC	-1.00%	5.74%
WBC	26.42%	16.28%
Hungaria	-0.92%	9.81%
CRX	0.68%	2.76%
Hepatiti	26.87%	24.62%
Anneal	35.71%	42.37%
Média ± Desvio Padrão	13.78%±15.36%	16.71%±13.45%

Tabela 8: Porcentagem de Queda nas Taxas de Erro Obtidas do Classificador Único para o Ensemble com 10 Classificadores

7 Conclusão

Os experimentos realizados com o algoritmo de boosting implementado pelo *See5* mostrou que essa técnica de construção de ensembles, como está implementada, não é satisfatória, pois foram obtidas grandes oscilações nas taxas de erro obtidas para os ensembles construídos; algumas vezes, a taxa de erro obtida para um ensemble com L classificadores, $L \geq 3$, é maior que a taxa de erro obtida por um único classificador. Entretanto, para alguns conjuntos de dados, houve melhora na predição, conforme o esperado. Inclusive, ensembles, com poucos classificadores muitas vezes tiveram suas taxas de erro menores que a taxa de erro de um classificador único.

Todavia, não se pode afirmar que a técnica de boosting não é uma boa técnica. Muitos trabalhos têm sido realizados em torno da técnica de boosting (Bauer and Kohavi, 1999; Dietterich, 2000b; Breiman, 1996b), bons resultados têm sido obtidos (Schapire et al., 1998; Breiman, 1998), e muitos trabalhos têm sido realizados para explicar as variações nos resultados obtidos quando se utiliza essa técnica (Rätsch et al., 2001). Assim, há outras técnicas de construção de ensembles que devem ser investigadas. O que se sabe é que, para algumas características do conjunto de dados a ser utilizado, alguns métodos de construção de ensembles não são aconselháveis. Por exemplo, para conjunto de dados com muito ruído, não se aconselha a utilização da técnica de boosting.

Referências

Baranauskas, J. A. and Monard, M. C. (2000). Reviewing some machine learning concepts and methods. Technical Report 102, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_102.ps.zip.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):105–139.

Bernardini, F. C. (2001). Combinação de classificadores para melhorar o poder preditivo e descritivo de *ensembles*. Monografia do Exame de Qualificação de Mestrado, ICMC-USP.

Blake, C., Keogh, E., and Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271.

Blum, A. and Rivest, R. (1988). Training a 3-node neural network is np-complete (extended abstract). In *Proceeding of the 1988 Workshop on Computacional Learning Theory*, pages 9–18, San Mateo, California. Morgan Kaufmann Publishers.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (1996b). Bias, variance and arcing classifiers. Technical Report 460, Statistics Department, University of California. ftp://ftp.stat.berkeley.edu/pub/users/breiman/.

Breiman, L. (1998). Arcing classifiers. *The Annals Of Statistics*, 26(3):801–849.

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., and Zanasi, A. (1998). *Discovering Data Mining: from Concept to Implementation*. Prentice Hall.

Dietterich, T. G. (2000a). Ensemble methods in machine learning. In *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, pages 1–15, New York. Springer Verlag. http://www.cs.orst.edu/~tgd/.

Dietterich, T. G. (2000b). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40:139–157. http://www.cs.orst.edu/~tgd/.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learninng and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Hansen, L. and Salamon, P. (1990). Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Hyafil, L. and Rivest, R. (1976). Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17.

Lee, H. D., Monard, M. C., and Baranauskas, J. A. (1999). Empirical comparison of wrapper and filter approaches for feature subset selection. Technical Report 94, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_94.ps.zip.

Mannila, H. (1996). Data mining: Machine learning, statistics, and database. Technical report, Department of Computer Science, University of Helsinki, FIN-00014 Helsinki, Finland. http://www.cs.helsinki.fi/~mannila [date: mar/2001].

Michalski, R., Bratko, I., and Kubat, M. (1998). *Machine Learning and Data Mining: Methods and Applications*. John Wiley & Son.

Mitchell, T. (1999). Machine learning and data mining. *Communications of the ACM*, 42(11):30–36. http://cs.cmu.edu/user/mitchell/ftp/tomhome.html.

Parmanto, B., Munro, P., and Doile, H. (1992). Improving commitee diagnosis with resamplig techniques. In Touretzky, D., Mozer, M., and Hesselmo, M., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 882–888, Cambridge, MA. The MIT Press.

Prati, R. C., Baranauskas, J. A., and Monard, M. C. (1999). BIBVIEW: Um sistema para auxiliar a manutenção de registros para o BIBTEX. Technical Report 95, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_95.ps.zip.

Quinlan, J. (1988). *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers.

Quinlan, J. (1996). *Bagging, Boosting and C4.5*. University of Sydney, Sydney, Australia. http://www.cse.unsw.edu.au/~quinlan/.

Quinlan, J. (2002). *See5*: An informal tutorial. Available at http://www.rulequest.com/see5-win.html [date: Abril/2002].

Rätsch, G., Onoda, T., and Müller, K. (2001). Soft margins for adaboost. *Machine Learning*, 42:287–320.

Schapire, R., Freund, Y., Bartlett, P., and Lee, W. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. In *The Annals of Statistics*, volume 26, pages 1651–1686. http://www.research.att.com/~schapire/.

Weiss, S. and Indurkhya, N. (1998). *Predictive Data Mining: a Practical Guide*. Morgan Kaufmann Publishers. http://www.data-miner.com.or.