Proposta de um Algoritmo para Indução de Árvores de Classificação para Dados Desbalanceados

Cláudio Frizzarini[1], **Marcelo S. Lauretto**[1]

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (EACH-USP)
Rua Arlindo Bettio, 1000 – 03828-000 – São Paulo – SP – Brazil

{cfrizzarini,marcelolauretto}@usp.br

Abstract. *Among the data mining tools embedded in decision support systems and* Business Intelligence *environments, classification trees have the advantages of being conpeptually simple and easily interpreted. However, many classification problems involve unbalanced datasets and, in such cases, low frequent classes tend to be negleted by algorithms driven to global error rates minimization. This work introduces a new algorithm for induction of decision trees for unbalanced datasets, with minimal user parameterization needs. Preliminary results show good mean within group error rates for the proposed algorithm in comparison to other competitors.*

Resumo. *Entre as ferramentas de mineração de dados disponíveis nos sistemas de apoio à decisão e ambientes de* Business Intelligence*, as árvores de classificação possuem as vantagens da simplicidade conceitual e da interpretabilidade. Todavia, são comuns problemas de classificação nos quais os dados são desbalanceados, e as classes minoritárias tendem a ser negligenciadas por algoritmos baseados em minimização de erro global. Neste trabalho propomos um novo algoritmo de indução de árvores de classificação para tratamento de dados desbalanceados, com baixa necessidade de parametrização pelo usuário. Resultados preliminares mostram boas taxas médias de erros intra-classes do método em relação a outros competidores.*

1. Introdução

As técnicas de mineração de dados, e mais especificamente de aprendizado de máquina, têm se popularizado enormemente nos últimos anos, passando a incorporar os Sistemas de Informação para Apoio à Decisão, Previsão de Eventos e Análise de Dados. Por exemplo, sistemas de apoio à decisão na área médica e ambientes de *Business Intelligence* fazem uso intensivo dessas técnicas, envolvendo particularmente árvores de decisão [Morais et al., 2012; Microsoft, 2006]. A mineração de informação e conhecimento a partir de grandes bases de dados tem sido reconhecida como tema chave de pesquisa em sistemas de banco de dados e aprendizado de máquina [Chen et al., 1996].

Concomitantemente a essa popularização, faz-se necessário o desenvolvimento de ferramentas de modelagem acessíveis, conceitualmente simples e com baixa necessidade de parametrização, que possam ser utilizadas (ao menos em análises mais simples) por profissionais que não sejam necessariamente especialistas nos métodos de modelagem subjacentes.

Algoritmos indutores de árvores de classificação, particularmente os algoritmos TDIDT (Top-Down Induction of Decision Trees), figuram entre as técnicas mais comuns de aprendizado supervisionado. A construção de uma árvore de decisão consiste em partições sucessivas do conjunto de treinamento original em subconjuntos menores. Uma das vantagens desses algoritmos em relação a outros é que, uma vez construída e validada, a árvore tende a ser interpretada com relativa facilidade, sem a necessidade de conhecimento prévio sobre o algoritmo de construção. Em um contexto de mineração de dados, mesmo que não sejam necessariamente utilizadas na classificação de novas instâncias, árvores de classificação podem ser construídas para fornecer descrições (na forma de regras de classificação) das características comuns aos membros de cada classe.

Todavia, são comuns problemas de classificação em que as frequências relativas das classes variam significativamente. Algoritmos baseados em minimização do erro global de classificação tendem a construir classificadores com baixos erros de classificação nas classes majoritárias e altos erros nas classes minoritárias [Batista et al, 2004; Qiao e Liu, 2009]. Esse fenômeno pode ser crítico quando as classes minoritárias representam eventos como a presença de uma doença grave (em um problema de diagnóstico médico) ou a inadimplência em um crédito concedido (em um problema de análise de crédito).

Diversos algoritmos TDIDT não possuem métodos adaptativos automáticos, demandando a calibração de parâmetros *ad-hoc* [Breiman et al, 1984] ou, na ausência de tais parâmetros, a adoção de métodos de balanceamento dos dados [Batista et al., 2004]. As duas abordagens não apenas introduzem uma maior complexidade no uso das ferramentas de mineração de dados para usuários menos experientes, como também nem sempre estão disponíveis.

Este trabalho apresenta uma descrição e os primeiros resultados empíricos de um algoritmo TDIDT em desenvolvimento, para construção de árvores na presença de dados desbalanceados. Esse algoritmo, denominado atualmente DDBT (Dynamic Discriminant Bounds Tree), utiliza um critério de partição de nós que, ao invés de se basear em frequências absolutas de classes, compara as proporções das classes nos nós com as proporções do conjunto de treinamento original, buscando formar subconjuntos com maior discriminação de classes em relação ao conjunto de dados original. Para a rotulação de nós terminais, o algoritmo atribui a classe com maior prevalência relativa no nó em relação à prevalência no conjunto original. Essas características fornecem ao algoritmo a flexibilidade para o tratamento de conjuntos de dados com desbalanceamento de classes, resultando em um maior equilíbrio entre as taxas de erro em classificação de objetos entre as classes.

Inicialmente, fixamos a notação apresentada neste trabalho, adaptada de Qiao e Liu (2009). Em um problema de classificação, é dado um conjunto de treinamento com N observações $\{\mathbf{x}_i, y_i\}$, $i = 1 \ldots N$. Cada entrada $\mathbf{x}_i$ é um vetor M-dimensional e y_i indica a qual classe $\mathbf{x}_i$ pertence. Suponha que haja K classes com $y_i \in \{1, 2, \ldots K\}$, e suponha que os exemplos $\{\mathbf{x}_i, y_i\}$'s sejam retirados ao acaso da população de forma independente, com a distribuição $M + 1$-dimensional de probabilidade conjunta subjacente $P(\mathbf{x}, y)$.

O objetivo do problema de classificação é construir um modelo de decisão (classificador) $\delta(\mathbf{x}) : \mathbf{R}^M \rightarrow \{1, 2, \ldots K\}$, com base na informação do conjunto de exemplos $\{\mathbf{x}_i, y_i\}$'s, para prever o rótulo da classe para futuros vetores $\mathbf{x}$. Logo, $\delta(\mathbf{x})$ precisa não

apenas classificar bem os exemplos do conjunto de treinamento, como também possuir boa capacidade de generalização, de maneira a classificar bem toda a população.

2. O Altoritmo DDBT

A versão atual do DDBT é voltada para problemas de classificação binária, e versões multiclasses deverão ser apresentadas em trabalhos futuros. As árvores geradas são estritamente binárias, ou seja, cada nó da árvore possui zero ou dois filhos.

O algoritmo inicia com um único nó terminal contendo o conjunto de treinamento completo. Em cada nó terminal, realizam-se os seguintes passos:

1. Binarização de cada atributo numérico ou categórico, e sua avaliação segundo um critério de ganho de convicção;
2. Seleção do melhor atributo (com sua correspondente binarização ótima) e a divisão apropriada do nó;
3. Caso não haja mais atributos que resultem em ganho positivo de convicção, o procedimento pára a expansão do nó, rotulando-o com uma das classes, de acordo com o critério de classificação.

2.1. Rotulação dos Nós e Função de Convicção

Para cada nó terminal t da árvore, considere que n exemplos incidem sobre o nó, dentre os quais n_k são de classe k, para $k = 1, 2$. Assumindo que os exemplos incidentes sobre o nó sejam independentes, podemos considerar que n_k segue uma distribuição Binomial com parâmetro desconhecido $\pi_k \in [0, 1]$:

$$P(n_k|\pi_k, n) = \binom{n}{n_k} \pi_k^{n_k} \, (1 - \pi_k)^{n-n_k}.$$

O parâmetro π_k representa a probabilidade de um exemplo incidente sobre o nó t ser da classe k, e seu estimador usual é $\hat{\pi}_k = n_k/n$. Aqui, consideramos uma abordagem Bayesiana, através da qual, ao invés de considerar o estimador pontual $\hat{\pi}_k$, nosso interesse é considerar uma distribuição de probabilidade para π_k. Demonstra-se que, assumindo uma distribuição a priori uniforme para π_k, sua distribuição a posteriori após a observação de n, n_k segue uma distribuição Beta, com parâmetros $n_k + 1, n - n_k + 1$ [DeGroot, 1986, cap. 6]:

$$f(\pi_k|n_k + 1, n - n_k + 1) = \frac{\Gamma(n + 2)}{\Gamma(n_1 + 1)\Gamma(n - n_k + 1)} \pi_k^{n_k} \, (1 - \pi_k)^{n-n_k}.$$

O DDBT adota uma medida de convicção adaptada do Algoritmo REAL – Real-Valued Attribute Learning [Lauretto, 1996; Stern et al, 1998].

O REAL rotula cada nó com a classe mais frequente, e define uma função de convicção baseada em um limitante superior do erro de classificação. Supondo, sem perda de generalidade, que a classe majoritária seja a classe 1 (e portanto a classe 2 é a minoritária), n_2 é o número de exemplos classificados erroneamente e π_2 representa a probabilidade de erro de classificação no nó. A medida de convicção no REAL é definida como $\text{conv}(t) = 100 * (1 - cm)\%$, onde

$$cm = \min c \;\mid\; Pr(\pi_2 \leq c) \geq g(c)$$

e $g(\ )$ é uma bijeção monotonicamente decrescente do intervalo $[0, 1]$ sobre si mesmo, preferencialmente côncava. No REAL, $g(c) = 1 - c^r$, onde $r > 1$ é um parâmetro de concavidade. A medida cm busca representar, simultaneamente, o limitante superior do erro, $Pr(\pi_2 \leq c)$, e o nível de convicção dinâmico, $g(c)$.

Observando que $Pr(\pi_2 \leq c) = F(c|n_2 + 1, n - n_2 + 1)$, onde F denota a função acumulada da densidade f, a medida cm pode ser reescrita como a raiz da função abaixo:

$$\begin{aligned} cm(n, n_2, r) &= c \mid h(c) = 0 \\ h(c) &= 1 - c^r - F(c|n_2 + 1, n - n_2 + 1). \end{aligned}$$

Como a função cm apresentada pelo REAL busca minimizar o erro global de classificação e se baseia no critério de rotulação por maioria simples, não é adequada para conjuntos desbalanceados. Assim, propõem-se no algoritmo DDBT versões modificadas do critério de classificação e da medida cm.

Rotulação:

Em nosso algoritmo, a rotulação da classe é feita pelo seguinte critério: denotando por p_k a proporção observada da classe k no conjunto de treinamento, $k = 1, 2$, rotulamos o nó t com a classe k_m cuja frequência relativa em t seja maior do que p_K, ou seja, $k_m = max_k \hat{\pi}_k / p_k$.

Função de Convicção:

A função de convicção modificada busca medir a discriminação das classes do nó t em relação ao conjunto de treinamento original. Isso é feito em três etapas:

- definir um nó hipotético t_0 com a mesma quantidade de exemplos de t (n), mas com proporções nas classes iguais ao do conjunto de treinamento original; o nó hipotético t_0 representa um nó "neutro", sem nenhum ganho de informação em relação aos dados originais;
- adotar uma distribuição de referência para as proporções das classes no nó t_0;
- comparar a distribuição das classes em t com a distribuição de referência em t_0.

Denotemos por ρ_k a probabilidade de um exemplo incidente em t_0 pertencer à classe k. Na ausência de uma distribuição a posteriori para ρ_0, adotamos uma distribuição de referência aproximada: a distribuição beta com parâmetros $(p_k n + 1, (1 - p_k)n + 1)$:

$$f(\rho_k | p_k n + 1, (1 - p_k)n + 1) = \frac{\Gamma(n + 2)}{\Gamma(p_k n + 1)\Gamma((1 - p_k)n + 1)} \rho_k^{p_k n} \ (1 - \rho_k)^{(1-p_k)n}.$$

Suponha, sem perda de generalidade, que o nó t seja rotulado com a classe 1. Note que, nesse caso, quanto menor for $\hat{\pi}_2$ em relação a p_2, maior será a evidência de discriminação entre as classes. Uma ideia é então testar a significância estatística da diferença entre $\hat{\pi}_2$ e p_2, usando as distribuições de π_2 e de ρ_2, respectivamente. A hipótese de interesse é $H_0 : \pi_2 = \rho_2$, contra a hipótese alternativa $H_1 : \pi_2 \neq \rho_2$.

Adotamos a abordagem proposta por Vêncio et al (2003), definindo o parâmetro $\tau = \pi_2/(\pi_2 + \rho_2)$. Observe que esse parâmetro está definido sobre o intervalo $[0, 1]$ e que, além disso, a hipótese $H_0 : \pi_2 = \rho_2$ equivale a $H_0 : \tau = 0, 5$.

Assim como na formulação do REAL são privilegiados nós com baixas probabilidades de erros de classificação, na formulação do DDBT busca-se privilegiar nós com baixos valores de τ. Assim, definimos a função de convicção modificada convm$(t) = 100 * (1 - cm)\%$, onde

$$cm = \min c \mid Pr(\tau \leq c) \geq 1 - c^r.$$

Embora Pham-Gia (2007) apresente uma aproximação analítica da função de densidade de probabilidade de τ, não conhecemos uma expressão para a probabilidade acumulada $Pr(\tau \leq c)$. A abordagem utilizada neste trabalho é aproximar τ por uma distribuição Beta, gerando amostras independentes de π_2 e ρ_2, computando os valores de τ correspondentes e realizando um ajuste pelo método de máxima verossimilhança.

O nome *Dynamic Discriminant Bounds Tree* (DDBT) é inspirado na função de convicção, pois esta busca estimar, dinamicamente, limitantes inferiores da discriminação das classes em t em relação à distribuição original no conjunto de treinamento.

Como mencionado acima, o parâmetro p_k é usualmente assumido como a proporção original da classe $k(k = 1, 2)$ no conjunto de treinamento. Todavia, esse parâmetro pode também ser calibrado pelo usuário, a fim de tornar o algoritmo mais sensível à classe 1 ou à classe 2 (vide seção 3).

2.2. Divisão dos Nós

Uma regra de binarização de um atributo j consiste em uma regra da forma $x_j \in R$, onde x_j denota o valor do atributo j, e R denota uma região dos valores do atributo j. Para atributos numéricos, $R = (-\infty, c]$, onde c é uma constante. Para atributos categóricos, R é um subconjunto não vazio das categorias do atributo j.

A partição binária do nó t pela regra "$x_j \in R$" consiste em dividir o conjunto de exemplos incidentes sobre o nó t em dois subconjuntos, de acordo com a validade da regra. Considere uma divisão candidata do nó t, com n exemplos, e denote por t_e e t_d os dois nós-filhos resultantes dessa divisão, cada qual com n_e e n_d exemplos. O ganho de convicção é obtido pela equação abaixo:

$$gain = n_e \text{ convm}(t_e) + n_d \text{ convm}(t_d) - n \text{ convm}(t).$$

Para a partição ótima do nó t, examinam-se todos os atributos e respectivas regiões candidatas, escolhendo-se aquela que resulta no maior ganho de convicção. Se não houver partições com ganho positivo, considera-se o nó como terminal, rotulando-o conforme a regra apresentada na subseção anterior.

3. Experimentos Numéricos e Resultados

Foram conduzidos testes numéricos para analisar o desempenho do DDBT frente a outros algoritmos de indução de árvores de classificação. Foram utilizados dezoito conjuntos de dados públicos, obtidos através do UCI Machine Learning Repository [Frank & Asuncion, 2010]. Esses *datasets* são caracterizados por classes binárias e ausência de *missing values*, uma vez que a versão atual do DDBT ainda não incorpora tratamento para os casos mais gerais envolvendo multi-classes ou *missing values*. Extensões para problemas com essas características estão em desenvolvimento.

A Tabela 1 apresenta um sumário dos *datasets*. A última coluna apresenta os percentuais das classes minoritárias e majoritárias em cada conjunto. As linhas da tabela são apresentadas em ordem crescente da participação da classe minoritária.

Tabela 1. Sumários dos *datasets* utilizados

Abreviação	Nome do Dataset	Tipos de Atributos	Qtde Atrib.	Qtde exemplos	Classes (minoritária, majoritária)	% nas Classes
PageBlock	Page Blocks Classification	Real, Inteiro	10	5242	Blocos de uma página: linha horizontal, texto	[6.3, 93.7]
Bank	Bank Marketing	Real, Categórico	16	4521	Contratação de produto bancário: sim, não	[11.5, 88.5]
Spect	SPECT Heart	Categórico	22	267	Avaliação cardíaca: Anormal, Normal	[20.6, 79.4]
SpectF	SPECTF Heart	Inteiro	44	267	Avaliação cardíaca: Anormal, Normal	[20.6, 79.4]
Blood	Blood Transfusion Service Center	Real	4	748	Doação de sangue feita em mar/07: sim, não	[23.8, 76.2]
Haberman	Haberman's Survival	Inteiro	3	306	Sobrevivência após cirurgia: óbito, sobrevivência	[26.5, 73.5]
Planning	Planning Relax	Real	12	182	Sinais de dois estágios mentais: Relaxado, Planejamento	[28.6, 71.4]
Statlog	Statlog (German Credit Data)	Inteiro, Categórico	20	1000	Risco de crédito: ruim, bom	[30, 70]
Statlog_n	Statlog (German Credit Data)	Inteiro	24	1000	Risco de crédito: ruim, bom	[30, 70]
Column_2C	Vertebral Column	Real	6	310	Diagnóstico de coluna vertebral: normal, anormal	[32.3, 67.7]
Monks2	MONK's Problems	Categórico	6	432	Classificação do exemplo: 0, 1	[32.9, 67.1]
Ionosphere	Ionosphere	Real, Inteiro	34	351	Qualidade do sinal recebido: ruim, bom	[35.9, 64.1]
Wdbc	Breast Cancer Wisconsin (Diagnostic)	Real	30	569	Tipo de tumor: maligno, benigno	[37.3, 62.7]
St_heart	Statlog (Heart)	Real, Categórico	13	270	Doença cardíaca: presente, ausente	[44.4, 55.6]
Sonar	Connectionist Bench (Sonar, Mines vs. Rocks)	Real	60	208	Identificação do objeto: rocha, mina	[46.6, 53.4]
Monks3	MONK's Problems	Categórico	6	432	Classificação do exemplo: 0, 1	[47.2, 52.8]
Chess	Chess (King-Rook vs. King-Pawn)	Inteiro, Categórico	36	3196	Pedra branca pode vencer: não vence, vence	[47.8, 52.2]
Monks1	MONK's Problems	Categórico	6	432	Classificação do exemplo: 0, 1	[50, 50]

O algoritmo proposto e as rotinas de testes foram implementados no ambiente R [R Core Team, 2012]. Além do DDBT, foram utilizados nos testes os seguintes algoritmos, também disponíveis para o ambiente R:

ctree (Conditional Inference Trees): utiliza a inferência condicional como método de partição em subconjuntos de forma binária e recursiva [Hothorn et al, 2006]. A distribuição do teste estatístico sob a hipótese nula é obtida calculando todos os possíveis valores do teste estatístico sobre os rearranjos dos dados para todos os

atributos. A implementação utilizada encontra-se no Pacote party [Hothorn et al, 2006], e não processa conjunto de dados com atributos do tipo categórico.

J48: implementa o C4.5 [Quinlan 1993] gerando árvores já podadas ou não. O critério para partição dos nós é o de maior ganho de informação (diferença na entropia de Shannon). Após a expansão da árvore, é realizado o procedimento de poda dos ramos com baixo ganho de acurárcia. A implementação utilizada encontra-se no Pacote RWeka [Kurt et al, 2009], e não processa conjuntos de dados com atributos do tipo categórico.

LMT (Logistic Model Trees): gera árvores de classificação com base em modelo de regressão logística para selecionar o atributo relevante no conjunto de dados, repetindo recursivamente o processo. LMT implementa funções de regressão simples e aplica o LogitBoost, que são modelos de generalização da regressão logística, na seleção de atributo para ramificação de um nó [Landwehr et al, 2005]. A implementação utilizada desse algoritmo também encontra-se no RWeka, e não processa conjuntos de dados com atributos do tipo categórico.

rpart: implementa a metodologia CART (Classification and Regression Trees) [Breiman, et al. 1984]. O critério de seleção de atributos para partição dos nós o índice de Gini. O crescimento da árvore é limitado a 31 (trinta e um) níveis de profundidade. A implementação utilizada encontra-se no Pacote rpart [Therneau et al, 2012], a qual permite a incorporação de custos distintos de erros de classificação nas classes, possibilitando assim aumentar a sensibilidade do classificador às classes minoritárias.

A análise de desempenho foi realizada através da Validação Cruzada [Mitchell, 1997]. Este método particiona o *dataset* em k sub-conjuntos aproximadamente de mesmo tamanho; em seguida, separa um dos k subconjuntos como conjunto de teste e utiliza os demais para construir a árvore. O processo é repetido de modo a gerar k árvores, todas testadas com conjuntos não utilizados em sua construção. Em nossos testes, utilizamos $k = \min\{20, \lfloor N/30 \rfloor\}$, onde N denota a quantidade de exemplos do conjunto. Dessa forma, garantiu-se que cada subconjunto k tivesse no mínimo 30 exemplos. Tomou-se também o cuidado de realizar a partição aleatória de cada *dataset* uma única vez, de modo a garantir que os treinamentos e testes dos diferentes algoritmos fossem realizados exatamente sobre os mesmos exemplos. Dessa forma, buscou-se evitar que as diferenças de desempenho entre algoritmos pudessem ser atribuídas às flutuações decorrentes da amostragem [Mitchell, 1997].

As medidas de desempenho apresentadas são baseadas nos indicadores tradicionais de *Sensibilidade* e *Especificidade* [Batista et al, 2004; Qiao & Liu, 2009]. Como ocorre usualmente em problemas de diagnósticos médicos, assumimos as classes minoritárias como Positivas, e as majoritárias como Negativas.

Após construída uma árvore, sua aplicação sobre o conjunto de testes resulta em uma matriz de confusão, que consiste nas contagens de exemplos de teste em cada um dos casos possíveis:

- VP (Verdadeiros Positivos): quantidade de exemplos positivos classificados corretamente;
- FN (Falsos Negativos): quantidade de exemplos positivos classificados erroneamente como negativos;

- FP (Falsos Positivos): quantidade de exemplos negativos classificados erroneamente como positivos;
- VP (Verdadeiros Negativos): quantidade de exemplos negativos classificados corretamente.

A partir dessas contagens, são obtidos os três indicadores de erros abaixo:

Taxa de falsos negativos: $FN_r = FN/(VP + FN)$ é o percentual de casos positivos (classe minoritária) classificados como negativos;

Taxa de falsos positivos: $FP_r = FP/(FP + VN)$ é o percentual de casos negativos (classe majoritária) classificados como positivos;

Taxa de erro médio intra-grupo: $EIG = (FN_r + FP_r)/2$. A taxa de erro médio intra-grupo (Mean Within Group Error Rate) é apresentada por Qiao & Liu (2009), em uma formulação geral incluindo multiclasses. Para problemas com classes raras, esse indicador tende a ser mais apropriado do que as medidas usuais de erro global de classificação.

A Tabela 2 apresenta as médias dos indicadores FN_r, FP_r e EIG obtidas nas validações cruzadas sobre os 18 *datasets*. As linhas da tabela estão em ordem crescente do percentual de participação da classe minoritária (coluna 2), e portanto as primeiras linhas correspondem aos *datasets* com maiores desbalanceamentos entre classes. As células sombreadas em verde escuro e em verde claro identificam, respectivamente, os algoritmo que obtiveram o menor e o segundo menor EIG para cada *dataset*. As células sombreadas na cor azul indicam os algoritmos que obtiveram a menor FN_r .

Tabela 2. Médias das taxas de taxas de falsos negativos (FN_r), falsos positivos (FP_r) e erros médios intra-grupos (EIG) obtidos a partir das iterações das validações cruzadas; as linhas estão ordenadas pelo percentual da classe minoritária em cada *dataset* (2^a coluna).

Dataset	% classe minor.	ctree			DDBTtree			J48			LMT			rpart		
		FN_r (%)	FPr (%)	EIG (%)	FN_r (%)	FPr (%)	EIG (%)	FN_r (%)	FPr (%)	EIG (%)	FN_r (%)	FPr (%)	EIG (%)	FN_r (%)	FPr (%)	EIG (%)
PageBlock	6.3	8.6	1.0	4.8	5.5	1.5	3.5	9.4	0.7	5.1	8.6	0.7	4.6	11.1	0.6	5.9
Bank	11.5	57.4	3.9	30.6	15.8	29.6	22.7	61.6	3.7	32.7	68.9	2.3	35.6	64.8	3.0	33.9
Spect	20.6	76.0	11.4	43.7	23.2	19.9	21.5	44.3	12.7	28.5	54.2	6.7	30.4	49.8	6.0	27.9
SpectF	20.6	73.6	10.4	42.0	53.7	14.7	34.2	56.1	16.9	36.5	66.2	9.8	38.0	56.1	17.4	36.7
Blood	23.8	59.4	9.9	34.7	36.9	31.5	34.2	62.2	9.0	35.6	63.1	6.4	34.7	66.7	6.4	36.5
Haberman	26.5	66.8	12.8	39.8	54.4	20.5	37.5	62.3	12.5	37.4	75.4	4.7	40.0	65.5	9.8	37.7
Planning	28.6	100.0	0.0	50.0	60.2	29.6	44.9	100.0	0.0	50.0	100.0	0.0	50.0	77.4	29.0	53.2
Statlog	30.0	54.1	13.9	34.0	32.3	30.1	31.2	61.9	15.5	38.7	55.3	12.5	33.9	61.7	12.1	36.9
Statlog_n	30.0	58.3	11.4	34.9	32.2	29.1	30.6	55.6	14.2	34.9	53.5	10.4	31.9	52.0	13.4	32.7
Column_2C	32.3	37.1	9.6	23.4	15.4	21.1	18.2	35.8	10.1	22.9	21.6	13.2	17.4	31.3	13.9	22.6
Monks2	32.9	100.0	0.0	50.0	45.5	50.6	48.1	21.9	2.6	12.3	29.4	6.0	17.7	47.2	8.0	27.6
Ionosphere	35.9	23.1	4.0	13.6	11.2	8.2	9.7	20.3	6.4	13.4	19.5	1.3	10.4	20.8	5.5	13.2
Wdbc	37.3	8.6	5.4	7.0	12.2	3.5	7.8	8.4	4.1	6.2	4.6	0.8	2.7	12.4	6.6	9.5
St_heart	44.4	37.4	14.7	26.0	28.5	23.3	25.9	23.7	18.5	21.1	22.5	12.9	17.7	24.9	16.1	20.5
Sonar	46.6	25.9	28.8	27.4	44.6	21.5	33.1	32.1	17.8	25.0	27.1	22.5	24.8	31.5	25.7	28.6
Monks3	47.2	0.0	5.2	2.6	0.0	5.2	2.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Chess	47.8	2.1	0.9	1.5	9.0	3.0	6.0	0.5	0.5	0.5	0.4	0.2	0.3	3.1	3.1	3.1
Monks1	50.0	0.0	49.5	24.7	0.0	49.5	24.7	0.0	49.5	24.7	0.0	49.5	24.7	10.2	30.5	20.3

Pode-se observar que, na maioria dos *datasets*, o DDBT obteve as menores taxas de falsos negativos dentre os cinco métodos, especialmente sobre os datasets mais desbalanceados. O LMT apresentou menores taxas de falsos positivos entre os *datasets* mais balanceados.

Um aspecto importante é que a menor taxa de erros na classe minoritária não implicou em incremento expressivo das taxas de erros na classe majoritária, pois o DDBT obteve o menor erro médio intra-grupo (EIG) em nove *datasets*, e o segundo menor EIG em três *datasets*. Outro aspecto relevante é que os melhores desempenhos do DDBT (menores valores de EIG) são observados nos *datasets* mais desbalanceados (com percentual da classe minoritária inferior a 30%). O LMT também teve um bom desempenho na comparação, apresentando o menor EIG em seis *datasets*, todos com percentual da classe minoritária superior a 30%.

Esses resultados preliminares sugerem que o DDBT tem boa adaptabilidade sobre *datasets* desbalanceados, pois fornece modelos de classificação com bom equilíbrio entre os erros nas classes.

Foi também realizada uma análise de sensibilidade e de especificidade para os métodos DDBT e rpart. A sensibilidade corresponde à taxa de verdadeiros positivos, ou seja, $1 - FN_r$; a especificidade corresponde à taxa de verdadeiros negativos, ou seja, $1 - FP_r$. A curva ROC (*Receiver Operating Characteristic*) é uma ferramenta gráfica bastante útil na análise de desempenho de algoritmos, por permitir estudar a relação entre essas duas medidas. O gráfico na curva ROC confronta a sensibilidade (no eixo vertical) com o complemento da especificidade (no eixo horizontal). O ponto ideal no gráfico é o ponto $(0, 1)$ (taxa de falsos positivos igual a 0 e de verdadeiros positivos igual a 100%). Assim, a área sob a Curva (AUC) é usualmente adotada como uma medida de qualidade [Batista et al, 2004]. Essa curva pode ser obtida quando o algoritmo dispõe de um parâmetro que permita controlar a sensibilidade e/ou a especificidade.

Neste trabalho, a curva ROC do DDBT foi construída testando-se as árvores para diferentes valores das proporções de referência (p_1, p_2), e a curva ROC do rpart foi construída variando-se os valores dos custos de erros de classificação. Para os demais algoritmos, não foram encontrados parâmetros de calibração, não sendo assim possível gerar as curvas ROC correspondentes.

A Figura 1 apresenta as curvas ROC do DDBT ($\times$) e rpart ($+$) sobre os dezoito *datasets*. Os gráficos são apresentados na mesma ordem da Tabela 2, ou seja, em ordem crescente da partição da classe minoritária. Na maioria dos conjuntos analisados, as curvas dos dois métodos são bastante similares, especialmente nos conjuntos mais desbalanceados (gráficos superiores). Por outro lado, nos *datasets* artificiais Monks1, Monks2 e Monks3 [Thrun et al, 1991], a curva ROC do DDBT ficou abaixo da curva do rpart. Esse resultado sugere a necessidade de uma investigação mais detalhada das possíveis causas para o desempenho inferior nesses conjuntos.

4. Conclusões

Neste trabalho apresentamos uma proposta de algoritmo indutor de árvores de classificação para dados desbalanceados. A abordagem proposta é inovadora, pois define uma função de convicção que estabelece limitantes dinâmicos da discriminação entre as frequências de classes em cada nó e distribuições de referência.

Na análise de taxas de falsos negativos, falsos positivos e erros médios intra-grupos, o DDBT apresentou menores erros de classificação nas classes minoritárias e menores taxas de erros intra-grupos, especialmente em conjuntos de dados desbalanceados. Essa característica é especialmente importante, nos casos em que a classe minoritária

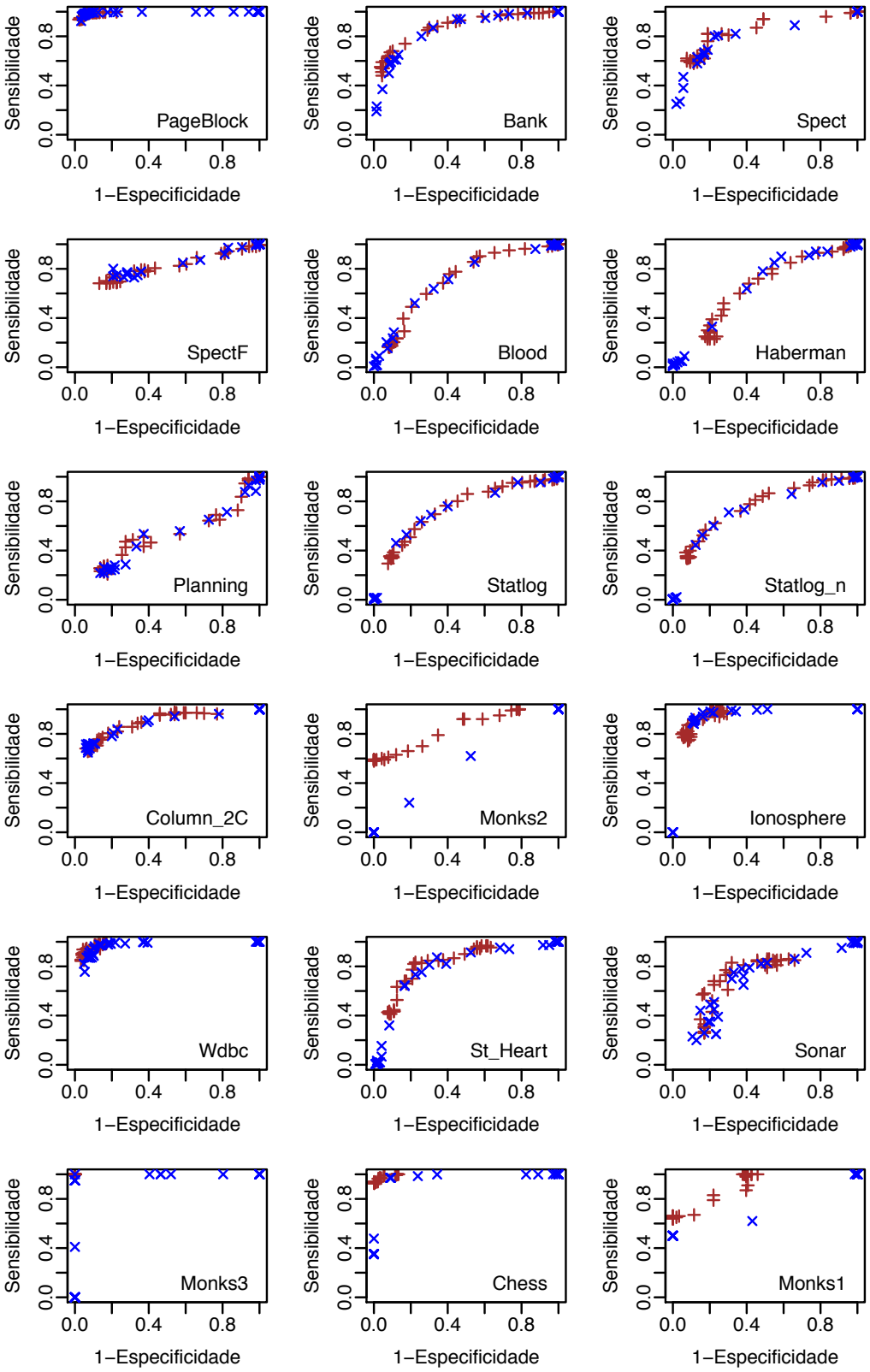

Figura 1. Curvas ROC obtidas a partir dos algoritmos DDBT ($\times$) e rpart ($+$).

representa eventos raros ou possui elevados custos de erro de classificação. Outro aspecto importante é que, nesses experimentos, o DDBT não precisou passar por calibração de nenhum parâmetro – uma característica desejável em ferramentas de mineração de dados incorporadas em sistemas de *Business Intelligence*.

Por outro lado, para conjuntos de dados com razoável balanceamento (proporção da classe inferior superior a 30%), os algoritmos tradicionais da literatura, em particular o LMT, apresentam desempenho superior ao DDBT.

As curvas ROC para análise de sensibilidade e especificidade entre os algoritmos DDBT e rpart demostraram desempenhos equilibrados entre o DDBT e o rpart.

Algumas extensões a esse trabalho são: o desenvolvimento de classificadores multi-classe; incorporação de tratamento de *missing values*; investigação do desempenho do algoritmo sob *ensembles* dos tipos *bagging* e *boosting*; análises teóricas das funções de convicção; análises comparativas envolvendo números maiores de *datasets* e algoritmos.

Os autores são gratos pelo apoio e financiamento recebidos da EACH-USP, do Programa de Pós-Graduação em Sistemas de Informação da EACH-USP e da Fundação de Apoio à Pesquisa do Estado de São Paulo (FAPESP).

Referências

Batista G.E.A.P.A., Prati R.C., Monard M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. *Sigkdd Explorations* 6(1), 20-29.

Breiman L., Friedman J., Stone C. J. e Olshen R. A. (1984). Classification and Regression Trees, Chapman and Hall.

Chen M., Han J. e Yu P. S. (1996). Data Mining: An Overview from Database Perspective. *IEEE Xplore Digital Library* 15(6), 866-883

Frank A., Asuncion A. (2010). "UCI Machine Learning Repository". University of California, Irvine, School of Information and Computer Sciences. `http://archive.ics.uci.edu/ml`

Gama J. (2004). Functional Trees. *Machine Learning* 55, 219-250.

Hornik K., Buchta C., Zeileis A. (2009) Open-Source Machine Learning: R Meets Weka. *Computational Statistics* 24(2), 225-232.

DeGroot M.H. (1986). Probability and Statistics, 2nd Ed. Menlo Park, CA: Addison-Wesley

Hothorn T., Hornik K. and Zeileis A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics* 15(3), 651-674

Landwehr N., Hall M. e Frank E. (2005). Logistic Model Trees. *Machine Learning* 59, 161-205.

Lauretto, M.S. (1996). Árvores de Classificação para Escolha de Estratégias de Operação em Mercados de Capitais. Dissertação de Mestrado, Instituto de Matemática e Estatística, Universidade de São Paulo.

Microsoft (2006). "SQL Server 2005 Analysis Services Tutorial". http://msdn.microsoft.com/en-US/library/ms170208$\%$28v=sql.90$\%$29.aspx

Mingers, J. (1989). An Empirical Comparison of Selection Measures for Decision-Tree Induction, *Machine Learning* 3, 319-342.

Mitchell, T.M. (1997). Machine Learning. McGraw-Hill.

Morais D.C.S., Morais B.C.S., Menezes Junior J.V. e Gusmão C.M.G. (2012). Sistema Móvel de Apoio a Decisão Médica Aplicado ao Diagnóstico de Asma - InteliMED. In *VIII Simpósio Brasileiro de Sistemas de Informação*, São Paulo, 2012.

Paulino C.D., Turkman M.A.A., Murteira B. (2003). Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa.

Pham-Gia T. (2007). Distributions of the ratios of independent beta variables and applications. *Communications in Statistics - Theory and Methods* 29(12), 2693–2715.

Quinlan J. R. (1993). C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Qiao X., Liu Y. (2009). Adaptive weighted learning for unbalanced multicategory classification. *Biometrics 65* 159-168.

Stern, J. M., Nakano, F., Lauretto, M. S. & Ribeiro, C. O. (1998). Algoritmo de Aprendizagem para Atributos Reais e Estratégias de Operação em Mercado. In: *Sixth Iberoamerican Conference on Artificial Intelligence - IBERAMIA'98*, Lisboa.

Therneau T., Atkinson B. and Ripley B. (2012). rpart: Recursive Partitioning. R package version 3.1-55. http://CRAN.R-project.org/package=rpart

Thrun, S.B.; Bala, J.; Bloedorn, E.; Bratko, I.; Cestnik, B.; Cheng, J.; De Jong, K.; Dzeroski, S.; Fahlman, S.E.; Fisher, D.; Hamann, R.; Kaufman, K.; Keller, S.; Kononenko, I.; Kreuziger, J.; Michalski, R.S.; Mitchell, T.; Pachowicz, P.; Reich, Y.; Vafaie, H.; Van de Welde, W.; Wenzel, W.; Wnek, J.; Zhang, J. (1991) "The MONK's Problems - A Performance Comparison of Different Learning algorithms". Technical Report CS-CMU-91-197, Carnegie Mellon University.

Vêncio R.Z.N., Brentani H., Pereira C.A.B. (2003). Using credibility intervals instead of hypothesis tests in SAGE analysis. *Bioinformatics* 19(18), 2461–2464.