Impacto de Estratégias de Balanceamento no Problema de Classificação de Sítios de *Splice*

Cláudia G. Varassin[1], Alexandre Plastino[1], Bianca Zadrozny[2], Helena G. Leitão[1]

[1]Departamento de Ciência da Computação– Universidade Federal Fluminense (UFF)
Rua Passo da Pátria 156, Bloco E CEP: 24210-240 – Niterói, RJ– Brazil

[2]IBM Research - Rio de Janeiro, RJ. – Brazil
{cvarassin,plastino}@ic.uff.br, biancaz@br.ibm.com, hcgl@ic.uff.br

Abstract. *Splice sites are the boundaries between certain stretches in eukaryotic genes. The detection of such sites in the DNA is a highly imbalanced classification task. Aiming to increase the learning ability in this problem, two existing resampling techniques designed to deal with this kind of imbalance are used. The experimental results show that is possible to increase classification performance using training sets with an imbalance factor different from the naturally occurring one.*

Resumo. *Sítios de splice são os locais de junção entre certos segmentos dos genes de eucariotos. A detecção desses sítios no DNA é um problema de classificação altamente desbalanceado. Visando aumentar a capacidade de aprendizado nesse problema, duas técnicas de reamostragem de dados que lidam com classes desbalanceadas são empregadas. Os resultados experimentais mostram que é possível melhorar o desempenho adotando conjuntos de treinamento com fatores de desbalanceamento distintos do que ocorre nos dados originais.*

1. Introdução

O avanço da tecnologia de sequenciamento de genomas produziu, nas duas últimas décadas, uma grande quantidade dados biológicos. A identificação da estrutura dos genes, em organismos eucariotos, é um dos problemas relacionado à extração do conhecimento contido nestes dados. Uma abordagem possível para a identificação das partes que compõem um gene é fazer o reconhecimento de certos padrões, característicos de cada segmento, independente da etapa que prediz a estrutura como um todo [Schweikert et al. 2009]. Os sítios de *splice* constituem um dos mais importantes deles.

As moléculas de DNA são longas cadeias formadas por uma sucessão de nucleotídeos. Nos genes dos eucariotos, os éxons (trechos usados diretamente na produção de proteínas) são intercalados por outros segmentos, os íntrons. As regiões de fronteira entre éxons e íntrons são conhecidas como sítios de *splice.* No ponto de transição entre um éxon e um íntron (sítio doador) aparece, em 99% dos casos, o dímero GT e, entre o íntron e o éxon (sítio aceptor) aparece o dímero AG [Brent and Guigó 2004].

Um aspecto particular deste problema é que a quantidade de verdadeiros doadores e aceptores encontrados no DNA é muito menor do que a quantidade de falsos sítios que aparecem nas sequências (segmentos contendo AG ou GT). O fato de haver uma grande diferença no número de exemplos de um tipo em relação ao outro dificulta a detecção pois os classificadores são sensíveis ao desbalanceamento e tendem a valorizar a classe predominante, não conseguindo aprender bem o conceito das classes menos representadas [Kotsiantis et al. 2006].

Em classificação, o nível de desbalanceamento é medido pela razão entre o número de instâncias da classe majoritária e o número da classe minoritária. No reconhecimento de sítios de *splice* fatores acima de 100 são encontrados, caracterizando o como um problema altamente desbalanceado.

Com a finalidade de tratar esta questão algumas estratégias têm sido propostas em vários domínios [Batista et al. 2004], no entanto, as soluções apresentadas na literatura para detectar sítios de *splice* não fazem uso de nenhum tipo de pré-processamento de dados para lidar com o desbalanceamento.

Neste trabalho objetivamos encontrar os sítios doadores e aceptores, empregando classificadores tradicionalmente usados em aprendizado de máquina e aplicando técnicas propostas pela comunidade científica que tratam o problema de classificação em conjuntos desbalanceado. Dentre as soluções existentes, adotamos os métodos que visam tornar o conjunto de treinamento menos desbalanceado. Optamos pelas estratégias de subamostragem das instâncias da classe majoritária e de sobre amostragem da classe minoritária. Estes métodos foram escolhidos por serem simples e não terem sido ainda aplicados neste problema. Ambos são categorizados por tratamento em nível de dados. Como a razão entre o número de instâncias de uma classe em relação à outra é alta, fizemos uma avaliação com fatores progressivos de balanceamento.

Este artigo está organizado da seguinte maneira. Na Seção 2, os principais trabalhos encontrados na literatura relacionados ao tema são apresentados. Na Seção 3, apresentamos os objetivos específicos do trabalho. Na Seção 4, as técnicas de classificação utilizadas são brevemente relatadas. Os experimentos conduzidos são descritos na Seção 5, e em seguida, na Seção 6, os resultados obtidos são apresentados e discutidos. Na Seção 7, concluímos o trabalho.

2. Abordagem na literatura

2.1 Reconhecimento de sítios de *splice*

O primeiro trabalho proposto visando detectar os sítios de *splice* no DNA foi publicado por Staden [Staden 1984]. O método descrito usa a distribuição composição dos nucleotídeos, por posição, existente em torno dos pontos de fronteira para construir um modelo. A distribuição captada através de verdadeiros sítios é armazenada em uma matriz, conhecida como matriz de pesos, que passa servir como padrão para avaliar novas sequências. Em um algoritmo chamado MDD (do inglês, *Maximal Dependence Decomposition*) [Burge and Karlin 1997], em vez de se usar apenas uma matriz de peso, uma seleção binária (do tipo se/senão) é aplicada no momento do treino de forma a dividir os dados em subconjuntos, por valor do atributo, sendo que a escolha do atributo,

para efetuar a partição, é baseada em cálculo de correlações entre as posições. O método equivale ao uso de árvores binárias onde, nas folhas, matrizes de peso são avaliadas.

Burge e Yeo [Burge and Yeo 2004] descrevem um algoritmo onde modelam o comportamento das instâncias captando a distribuição de probabilidade destas, fazendo uma busca em uma gama de possibilidades previamente estabelecida e usando o cálculo da entropia como critério para a escolha da distribuição mais próxima da real. Em 2005, os autores de *DGSplicer* [Chen et al. 2005] propuseram um algoritmo baseado em redes Bayesianas onde a topologia da rede é construída usando restrições no número de parentes que cada atributo pode ter e não limitando a dependência das posições a apenas posições adjacentes. Em ambos os algoritmos citados, a rotulação de uma nova instância é feita calculando-se a razão entre a probabilidade de uma sequência ser da classe positiva e a probabilidade de ser da classe negativa, ou seja, utilizando a razão de verossimilhança. Os modelos são construídos tomando a mesma relação entre quantidade de exemplos negativos e positivos existente no conjunto original.

Em *Splice Machine,* proposto em [Degroeve et al. 2005], o classificador SVM (máquina de suporte vetorial) linear é empregado para fazer a rotulação. Eles conduziram uma série de experimentos para avaliar a capacidade preditiva ao incluir atributos derivados e usaram sub amostragens do conjunto original (com menor desbalanceamento) tanto para efetuar o treino quanto para os testes. Os autores relataram que foi necessário tomar conjuntos com menos exemplos que o original devido ao custo computacional. No trabalho de [Sonnenburg et al. 2007] os autores também usam o classificador SVM mas adotam kernels projetados para detectar correlações locais nos atributos. Em um dos experimentos conduzidos apenas um quinto dos exemplos negativos é usado na fase de treinamento. A razão pela escolha deste fator de subamostragem não é explicitamente relatada. Os autores apenas sugerem que o ideal é sempre usar todos os dados disponíveis e, quando isto não for possível, devido à demanda computacional, então deve-se induzir o modelo retirando somente os exemplos da classe majoritária.

Até onde saibamos, não há trabalhos na literatura onde haja algum pré-processamento de dados visando lidar com o fato de que a identificação de sítios de *splice* seja um problema altamente desbalanceado.

2.2 O tratamento das classes desbalanceadas.

A rotulação de instâncias em problemas com classes muito desbalanceadas tem sido tratada pela comunidade de aprendizado de máquina via dois caminhos distintos [Deepa and Punithavalli 2010]. Um deles consiste em atribuir custos diferenciados às classes no momento de induzir o modelo [Zadrozny et al. 2003]. O outro é anterior à fase de treinamento e se baseia em reamostragem de dados.

Na abordagem de reamostragem, a ideia é usar um conjunto mais balanceado do que o conjunto original para construir o classificador. Os métodos de tratamento baseados em reamostragem se dividem em duas categorias, conhecidas como subamostragem (em inglês, *undersampling)* e sobre amostragem (em inglês, *oversampling)*.

Em subamostragem, dados da classe majoritária são removidos, enquanto que na técnica de sobre amostragem dados da classe minoritária são incluídos. A forma de

remoção ou adição das instâncias varia. As mais simples fazem a extração ou replicação dos dados com escolha aleatória.

Weiss e Provost [Weiss and Provost 2003] fizeram uso da técnica de subamostragem com retirada aleatória, construindo conjuntos para o treinamento com diversos fatores de balanceamento. Mostraram que ganhos de desempenho podem ser obtidos.

Técnicas de reamostragem orientada também são encontradas na literatura. Uma delas, denominada *SMOTE* (do inglês, *Synthetic Minority Over-sampling Technique*) [Chawla et al. 2002], consiste em criar instâncias sintéticas da classe minoritária com o objetivo de aumentar o espaço de decisão desta classe. A geração dos exemplos sintéticos é feita tomando, dentre *k* (parâmetro do algoritmo) vizinhos mais próximos da classe minoritária, alguns aleatoriamente. Para cada um destes vizinhos, via interpolação, cria-se um novo exemplo. O número de vizinhos a tomar vai depender do aumento desejado.

3. Objetivo específico

O objetivo do nosso trabalho é verificar se o uso da estratégia de subamostragem aleatória assim como o emprego da técnica *SMOTE* afetam o desempenho da classificação de sítios de *splice.*

4. Técnicas de Classificação

Os algoritmos Naive Bayes, Alternating Decision Tree e Máquina de Suporte Vetorial (SVM) linear foram empregados nos experimentos. Foram escolhidos pelo fato de serem métodos de aprendizado com características bem distintas entre si e amplamente utilizados em classificação binária.

Naive Bayes é um classificador que se baseia na distribuição de probabilidade das instâncias, por classe, para fazer a rotulação. Assume que os atributos são independentes entre si [Han and Kamber 2006]. O algoritmo Alternating Decision Tree [Freund 1999] é uma generalização do indutor C4.5. A estratégia empregada para o aprendizado é a de *boosting*, ou seja, constroem-se iterativamente vários indutores simples e, através de uma combinação destes, gera-se o classificador final. SVM é um classificador tipicamente projetado para classificação binária (embora existam estratégias para resolver problemas com várias classes). A ideia é construir um separador linear entre as classes. A geração deste separador é feita impondo-se restrições de forma a obter a máxima margem de separação possível entre os dados das classes [Han and Kamber 2006].

As implementações existentes na ferramenta Weka [Witten and Frank 2005] foram utilizadas. Para SVM, o indutor SMO (Sequencial Minimal Optimization) foi adotado.

5. Experimentos

Para realizar os experimentos utilizamos os dados de [Chen et al. 2005]. A coleção é constituída por 2379 sítios doadores e 2379 sítios aceptores de genes humanos. O

número de exemplos negativos é de 283062 e de 400314, respectivamente para doadores e aceptores, correspondendo a fatores de desbalanceamento de 119 e de 168.

Os conjuntos de treino e teste são gerados a partir de cinco partições provenientes da coleção D dos dados originais. A construção é feita de forma que os conjuntos usados para medir o desempenho (os conjuntos de teste) sejam sempre os mesmos, variando apenas os dados para o aprendizado.

Nos experimentos com bases de treinamento contendo a mesma distribuição de classes do conjunto original, cada partição D_k é construída tomando um quinto das instâncias positivas de D mais um quinto das instâncias negativas. Desta forma, cada partição D_k, dada pela união dos exemplos positivos $D_{k,pos}$ com os negativos $D_{k,neg}$ possui desbalanceamento $r=|D_{k,neg}|/|D_{k,pos}|$ idêntico ao do conjunto original. Para as avaliações com conjuntos de treinamento com fatores de desbalanceamento r distintos do original, cada partição D^r_k é gerada tomando todas as instâncias de $D_{k,pos}$ mais $r*|D_{k,pos}|$ instâncias da partição $D_{k,neg}$, fazendo retirada aleatória. As bases para o aprendizado são obtidas, ou unindo as partições D_k (para treinamento com desbalanceamento original) ou as partições D^r_k, (para treinamento com fator de desbalanceamento r). Os conjuntos são dados por $T_k = D_1+... + D_{k-1}+ D_{k+1} +...+ D_5$ e $T^r_k= D^r_1+...+D^r_{k-1}+D^r_{k+1}+...+D^r_5$.

Para se usar bases de treinamento pré-processadas com estratégia de sobre amostragem, o algoritmo SMOTE foi aplicado em cada conjunto T_k, gerando um novo conjunto T^S_k com mais instâncias da classe positiva do que a quantidade existente em T_k, sendo o aumento dependente da taxa de adição escolhida.

A base t_k usada para teste é idêntica à partição D_k possuindo o mesmo desnível de classes do conjunto original.

Como há uma aleatoriedade envolvida nas técnicas empregadas (na retirada e geração de instâncias) repetiu-se cada experimento três vezes. O desempenho final P é dado pela média das médias de cada procedimento de validação. É importante notar que ao fazer o cálculo do desempenho desta forma aparecem variações de natureza distintas. Há o desvio padrão observado nos resultados entre as partições (da validação cruzada) e o desvio padrão observado entre as três execuções.

Para avaliar o desempenho foi utilizada a métrica F. A métrica F foi escolhida pois permite quantificar, em um só valor, duas medidas que caminham em sentidos antagônicos. F representa a média harmônica entre precisão e sensibilidade, dada por $F= 2 /((1/Sensibilidade)+(1/Precisão))$.

Tanto na estratégia de subamostragem quanto na super amostragem, vários níveis de desbalanceamento foram utilizados. Os fatores r=1, r=10, r=20 e r=40 foram testados nos dois problemas de reconhecimento de sítios. Além destes, o fator correspondendo a uma proporção de negativos e positivos equivalente à metade do desbalanceamento dos dados originais (r=60 para doadores e r=84 para aceptores) foi usado. Para a sobre amostragem, o algoritmo SMOTE [Chawla et al. 2002] foi executado com taxas de 100%, 200% e 500% (e o número de vizinhos mais próximos foi sempre fixo igual a cinco). Os percentuais tomados implicam em fatores de desbalanceamentos de aproximadamente 60, 40 e 20 para o conjunto de sítios doadores e de 84, 56 e 28 nos sítios aceptores.

6. Resultados obtidos

A Tabela 1 mostra os valores da medida F obtidos com os três classificadores, usando o conjunto de treinamento com fatores crescentes de desbalanceamento, em sítios doadores. A Tabela 2 exibe os resultados com o uso da técnica SMOTE para as três taxas de aumento avaliadas. O desempenho com uso das técnicas de subamostragem e de super amostragem, em sítios aceptores, estão exibidos nas Tabelas 3 e 4.

O valor apresentado, para cada experimento, é a média dos valores médios das medidas F obtidos na validação cruzada. O desvio padrão mostrado é o maior desvio observado na validação cruzada. O desvio padrão (entre as repetições) do cálculo da média final foi sempre inferior ao desvio observado entre as partições.

Para verificar se as médias são distintas aplicou-se o teste-t pareado (com nível de significância de 0,05). Foi feita uma análise comparando o desempenho da classificação feita com conjunto de treinamento com balanceamento *r* com o desempenho da classificação que empregou um conjunto com fator de desbalanceamento imediatamente acima e também com treinamento com a distribuição original. Os resultados das análises estão colocados nas Tabelas 1, 2, 3 e 4. Quando os valores foram considerados distintos colocou-se a palavra "*dis"* e, em caso contrário, a palavra "*não"* foi usada. Os melhores valores atingidos estão em negrito (vários aparecem destacados quando não são estatisticamente diferentes).

Tabela 1. Desempenho (medida F) em sítios doadores, com subamostragem.

	r=1	r=10	r=20	r=40	r=60	Original (r=119)
N. Bayes	0,125 (0,003) 1 | 10: dis 1 | Or: dis	0,303 (0,008) 10 | 20: dis 10 | Or: dis	**0,368** (0,008) 20 | 40: não 20 | Or: dis	**0,382** (0,009) 40 | 60: dis 40 | Or: dis	0,361 (0,008) 60 | Or: dis	0,267 (0,018)
ADTree	0,165 (0,004) 1 | 10: dis 1 | Or: dis	0,369 (0,003) 10 | 20: dis 10 | Or: dis	**0,411** (0,011) 20 | 40: não 20 | Og: dis	**0,400** (0,008) 40 | 60: não 40 | Or: dis	**0,390** (0,016) 60 | Or: dis	0,298 (0,023)
SVM	0,163 (0,004) 1 | 10: dis 1 | Or: dis	0,353 (0,006) 10 | 20: dis 10 | Or: dis	**0,404** (0,012) 20 | 40: não 20 | Or: dis	**0,383** (0,020) 40 | 60: dis 40 | Or: dis	0,347 (0,021) 60 | Or: dis	0,179 (0,030)

Tabela 2. Desempenho (medida F) em sítios doadores, com a técnica SMOTE.

	Smote 500 (r~20)	Smote 200 (r~40)	Smote 100 (r~60)	Original (r=119)
N. Bayes	**0,351** (0,012) 500 | 200: não 500 | Or : dis	**0,360** (0,007) 200 | 100: não 200 | Or : dis	**0,351** (0,015) 100 | Or : dis	0,267 (0,018)
ADTree	**0,402** (0,010) 500 | 200: não 500 | Or : dis	**0,388** (0,019) 200 | 100: não 200 | Or : dis	**0,398** (0,019) 100 | Or : dis	0,298 (0,023)
SVM	**0,380** (0,008) 500 | 200: não 500 | Or : dis	**0,389** (0,020) 200 | 100: não 200 | Or : dis	**0,408** (0,027) 100 | Or : dis	0,179 (0,030)

Tabela 3. Desempenho (medida F) em sítios aceptores, com subamostragem.

	r=1	r=10	r=20	r=40	r=84	Original (r=168)
N. Bayes	0,082 (0,003) 1 | 10: dis 1 | Or: dis	0,169 (0,007) 10 | 20: dis 10 | Or: dis	0,202 (0,005) 20 | 40: dis 20 | Or: dis	**0,260** (0,006) 40 | 84: não 40 | Or: não	**0,261** (0,006) 84 | Or: não	**0,268** (0,001)
ADTree	0,095 (0.001) 1 | 10: dis 1 | Or: dis	0,247 (0.006) 10 | 20: dis 10 | Or: dis	**0,299** (0,005) 20 | 40: não 20 | Or: dis	**0,310** (0,014) 40 | 84: dis 40 | Or: dis	0,291(0,014) 84 | Or: dis	0,191(0,033)
SVM	0,097(0,003) 1 | 10: dis 1 | Or: dis	0,248 (0,005) 10 | 20: dis 10 | Or: dis	**0,305** (0,002) 20 | 40: não 20 | Or: dis	**0,300** (0,005) 40 | 84: dis 40 | Or: dis	0,158 (0,006) 84 | Or: dis	0,117(0,021)

Tabela 4. Desempenho (medida F) em sítios aceptores, com a técnica SMOTE.

	Smote 500 (r~28)	Smote 200 (r~56)	Smote 100 (r~84)	Original (r=168)
N. Bayes	0,203 (0,004) 500 | 200: dis 500 | Or : dis	**0,249** (0,004) 200 | 100: não 200 | Or : dis	**0,261** (0,007) 100 | Or : não	**0,268** (0,001)
ADTree	**0,274** (0,015) 500 | 200: não 500 | Or : dis	**0,292** (0,028) 200 | 100: não 200 | Or : dis	**0,277** (0,023) 100 | Or : dis	0,191 (0,033)
SVM	**0,302** (0,007) 500 | 200: não 500 | Or : dis	**0,299** (0,005) 200 | 100: dis 200 | Or : dis	0,226 (0,002) 100 | Or : dis	0,117(0,021)

Nas tabelas, vê-se que os valores da medida F variam significativamente com os diversos níveis de desbalanceamento avaliados.

Em sítios doadores, verifica-se um ganho ao usar conjuntos de treinamento com fatores de desbalanceamento de 20 e 40 para os classificadores Naive Bayes e SVM. Para ADTree o desempenho foi melhor com os fatores 20, 40 e 60. Além disso, nota-se que, para os três indutores, o emprego da técnica de super amostragem também resultou em valores de F superiores àqueles atingidos com a distribuição original.

No reconhecimento de sítios aceptores, um ganho no desempenho pode ser observado ao se usar fatores 20 e 40, para ADTree e SVM. Para Naive Bayes, os resultados atingidos com fatores 40, 84 e com distribuição original se mostraram estatisticamente idênticos. A super amostragem beneficiou a taxa de reconhecimento destes sítios para os classificadores ADTree e SVM.

Verifica-se ainda que, para ambas as estratégias, o desempenho aumenta com certos valores de fator de balanceamento, não importando qual a técnica específica tenha sido empregada.

Além disso, percebe-se que a melhoria alcançada é similar em uma larga faixa de valores, não dependendo de uma escolha exata para o fator de desbalanceamento.

7. Conclusões

Os resultados apresentados mostram que métodos de balanceamento afetam o desempenho da classificação e podem aumentar o taxa de reconhecimento de sítios de *splice.* Aprender com conjuntos com níveis intermediários de desbalanceamento, em relação ao original, se mostrou quase sempre mais vantajoso do que com conjuntos pouco desbalanceados ou do que com conjuntos com a distribuição original.

Tanto a aplicação de subamostragem quanto a aplicação de super amostragem podem melhorar o desempenho da detecção de sítios doadores e aceptores. Como as duas estratégias proporcionaram ganhos equivalentes a subamostragem deve ser preferida já que não envolve a aplicação de um algoritmo específico, e sim, apenas a aplicação de um procedimento de retirada aleatória. Além desta simplicidade, a subamostragem tem a vantagem de gerar conjuntos finais com tamanhos menores do que o original, implicando tanto em uma diminuição no tempo de indução dos classificadores como também em uma redução de espaço para armazenar os dados, fator que pode ser importante ao se lidar com uma grande quantidade de organismos.

Referências

Batista, G. E., Prati, R. C. and Monard, M. C. (2004) A study of the behavior of several methods for balancing machine learning training data. In *SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets*, v.5, n.1, p.20-29.

Brent, M. R. and Guigó, R. (2004) Recent Advances in gene Structure Prediction. In *Current Opinion in Structural Biology*, v.14, p.264-272.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002) SMOTE: Synthetic Minority Over-sampling Technique JAIR, v.16, p.321–357.

Chen, T. M. et al (2005) Prediction of splice sites with dependency graphs and their expanded Bayesian networks. In *Bioinformatics,* v.21, p.471-482.

Deepa, T. and Punithavalli, M. (2010) An Analysis for Mining Imbalanced Datasets. In *International Journal of Computer Science and Information Security*, v.8, p.132-137.

Degroeve, S., Saeys, Y., Baets, B. D., Rouzé, P. and de Peer, Y. V. (2005) SpliceMachine: predicting splice sites from high-dimensional local context representations". In *Bioinformatics,* v.21, p.1332-1338.

Freund Y. (1999) The alternating decision tree learning algorithm, In *Machine Learning: Proceedings of the Sixteenth International Conference*, p.124-133.

Han, J. and Kamber, M. (2006) Data Mining, Concepts and techniques. Morgan Kaufmann. 2nd edition

Kotsiantis, S., Kanellopoulos, D. and Pintelas, P. (2006) Handling imbalanced datasets: A review. In *GESTS International Transactions on Computer Science and Engineering.*

Schweikert, G, et al. (2009) mGene: accurate SVM-based gene finding with an application to nematode genomes. *In Genome Research,* v.19, p.1233-2143.

Sonnenburg, S., Philips, P., Schweikert, G. and Rätsch, G. (2007) Accurate splice site prediction using support vector machines. In *BMC Bioinformatics*, v.8.

Yeo, G. and Burge, C. (2004) Maximum entropy modeling of short sequences motifs with applications to RNA splicing signals. In *Journal of Computational Biology* v.11, p.377-94.

Staden, R. (1984) Computer methods to locate signals in nucleic acid sequences. *In Nucleic Acids Research.,*v.12, p.505–519.

Weiss, G. M. and Provost, F. (2003) Learning when training data are costly: the effect of class distribution in tree induction. *In Journal of Artificial Intelligence Research* v.19, p. 315-354.

Witten, I. H. and Frank, E. (2005) Practical Machine Learning Tools and Techniques. Morgan Kaufmann. 2nd edition.

Zadrozny, B., Langford J. and N. Abe (2003) Cost-Sensitive Learning by Cost-Proportionate Example Weighting. In *Proceedings of the 2003 IEEE International Conference on Data Mining*.