Um Estudo de Limpeza em Base de Dados Desbalanceada com Sobreposição de Classes

Emerson Lopes Machado e Marcelo Ladeira

Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Caixa Postal 4.466 – CEP 70.919-970 – Brasília – DF

emersoft@conectanet.com.br, mladeira@unb.br

Abstract. *This paper presents an undergoing study of the degeneration of classifiers resulting from skewed class distribution. Its central hypothesis is that the applying of regular sampling techniques may not improve classifiers' performance probably because of the class overlap natural to the data or caused by the sampling techniques. A new approach based on non supervised learning is present for this problem. The results so far obtained from the method developed with this approach, denoted C-clear, are not conclusive yet, though it seems to be a promise approach.*

Resumo. *Este artigo apresenta um estudo em desenvolvimento sobre degradação de classificadores no domínio de base de dados desbalanceadas. A hipótese central deste estudo é que a aplicação das técnicas convencionais de amostragem de dados neste domínio pode não resultar em melhora de desempenho, provavelmente devido ao uso delas aumentar a ocorrência de sobreposição de classes ou esta ser inerente aos dados. Para tanto, é apresentada uma nova abordagem para este problema baseada na aprendizagem não supervisionada. Os resultados até agora obtidos com o método desenvolvido sob esta nova abordagem, intitulado C-clear, ainda não são conclusivos, mas indicam que ela se mostra promissora.*

1. Introdução

No domínio de classificação, uma base de dados é dita desbalanceada quando existem muito menos casos de algumas classes do que de outras [Chawla *et al.*, 2004]. Classificadores desenvolvidos com métodos tradicionais são sensíveis a este tipo de desbalanceamento e tendem a valorizar classes predominantes e a ignorar classes de menor representação (também chamadas de classes raras) [Phua *et al.*, 2004]. Os classificadores gerados a partir de bases de treinamento desbalanceadas apresentam altas taxas de falsos negativos para as classes raras, o que é problemático quando a classe de interesse é classe rara. Para evitar esse viés, técnicas de pré-processamento dos dados são utilizadas para alterar as distribuições das classes na base de treinamento, visando reduzir desbalanceamento. Uma abordagem bastante comum é o uso de métodos de amostragem. Tais métodos consistem na eliminação de casos da classe majoritária (*undersampling*) e replicação (ou geração sintética) de casos da classe minoritária (*oversampling*) dos dados de treinamento visando obter classificadores melhores do que os obtidos a partir da distribuição original. Segundo Jo e Japkowicz (2004), não há garantia de que a distribuição original dos dados de treinamento seja a mais adequada para a construção de classificadores. Um método citado freqüentemente na literatura

para esse fim é o SMOTE[1] de Chawla *et al.* (2002) e suas diversas variações [Guo & Viktor, 2004] [Batista *et al.*, 2004] [Han *et al.*, 2005].

A simples replicação de casos positivos (instâncias de dados associados à classe de interesse) pode produzir classificadores muito específicos para os casos replicados e com baixo poder de generalização para outros casos positivos. A abordagem adotada no método SMOTE consiste em gerar casos sintéticos (artificiais) para a classe de interesse a partir dos casos já existentes. Os novos casos são gerados na vizinhança de cada caso da classe minoritária com o intuito de se crescer o espaço de decisão desta classe (região do $\Re^n$) e aumentar o poder de generalização dos classificadores obtidos. Visualmente, no espaço amostral do conjunto de dados, os novos casos sintéticos são interpolados aleatoriamente ao longo do segmento de reta que une cada caso da classe minoritária a um de seus *k* vizinhos mais próximos, escolhidos de forma aleatória. No entanto, o SMOTE pode apresentar o efeito indesejável de criação de casos positivos que invadem o espaço de decisão da classe negativa. Essa característica, denominada sobreposição de classes, tende a degradar o desempenho de classificadores obtidos a partir de tais dados [Batista *et al.*, 2004]. A sobreposição de classes também pode ser uma característica natural dos dados que é aguçada pela aplicação do SMOTE. Esse problema do relacionamento entre desbalanceamento e sobreposição de classes tem recebido atenção da comunidade [Prati *et al.*, 2004]. Batista *et al.* (2004) utilizam os métodos *Tomek links* [Tomek, 1976 apud Batista *et al.*, 2004] e ENN [Wilson, 1972 apud Wilson & Martinez, 2000] para a limpeza de dados após a aplicação do SMOTE. A aplicação de algoritmos de limpeza (exclusão de casos considerados inadequados) é desejada devido a erros de classificação ocorrerem com freqüência perto do limiar de decisão, onde geralmente ocorre sobreposição de classes [Chawla *et al.*, 2004].

Neste artigo, a sobreposição de classes é tratada com uma nova abordagem, que intuitivamente consiste na limpeza e geração de dados sintéticos com SMOTE, guiados por aprendizagem não supervisionada. O método desenvolvido sob esta abordagem foi nomeado *C-clear*. Este trabalho se restringe à sobreposição de classes no domínio de base de dados binária, na qual são utilizados os conceitos classe positiva para a classe minoritária (classe de interesse) e classe negativa para a classe majoritária. Com o intuito de simplificar a nomenclatura utilizada, os casos da classe positiva e negativa são chamados de casos positivos e negativos, respectivamente.

Este artigo está organizado da seguinte maneira. Na Seção 2, o método *C-clear* e suas principais características são apresentadas. Na Seção 3, são apresentados os resultados experimentais obtidos com a aplicação do *C-clear*, e na Seção 4 são apresentadas conclusões e sugestões para trabalho futuro.

2. *C-clear*

A abordagem proposta nessa pesquisa visa guiar a aplicação do método SMOTE, de modo a diminuir a quantidade de ruído por ele gerado, e logo após fazer limpeza de dados. Intuitivamente, esta abordagem consiste em aplicar o método SMOTE aos casos positivos que residem em regiões onde a freqüência da classe positiva seja maior do que certo limiar e, logo após, remover os casos negativos (positivos) das regiões onde a freqüência da classe positiva (negativa) seja maior do que certo limiar. Tais regiões são

[1] do ingles, *Synthetic Minority Oversampling Technique*

obtidas com aprendizagem não supervisionada. O funcionamento dessa abordagem se baseia em duas premissas: primeiro, os dados tenderem a se agrupar em *clusters*, ao invés de se distribuir uniformemente no espaço amostral e segundo, a variável classe ser um atributo relevante a esse agrupamento.

O *C-clear* está baseado em três passos (Figura 1). No primeiro, os dados de entrada são agrupados (Figura 1a e Figura 1b). No segundo (Figura 1c) os *clusters* são rotulados. No terceiro (Figura 1d) é aplicado o método SMOTE aos casos positivos dos *clusters* positivos. Para a aplicação do SMOTE, o parâmetro limiarSmote indica qual a freqüência mínima de casos positivos para que um *cluster* seja considerado positivo. O limiarSmote varia no intervalo [0,1], sendo 0 para considerar todos os *clusters* como positivo e, portanto, aplicar SMOTE em todos os casos positivos, e 1 para não aplicar SMOTE. Após a aplicação do método SMOTE nos *clusters* positivos pode ser feita uma limpeza de dados com a remoção dos casos ruidosos (*i.e.*, com classes dissonantes do rótulo do *cluster* ao qual pertencem). A intensidade da limpeza depende do parâmetro limiarLimpezaPositivo (limiarLimpezaNegativo) o qual indica qual a freqüência positiva (negativa) mínima admitida para que um *cluster* seja considerado positivo (negativo). Os parâmetros utilizados no *C-clear* variam no intervalo real [0,1], embora na prática poucos valores sejam de interesse. De fato, o número máximo de valores de interesse é igual à quantidade de *clusters* que possuem freqüências de classes positivas distintas. O menor valor de interesse para os parâmetros limiarSmote e limiarLimpezaPositivo (limiarLimpezaNegativo), depois do 0, é a menor freqüência relativa de casos positivos (negativos) dos *clusters.* Por analogia, o maior valor de interesse, antes do 1, para os parâmetros limiarSmote e limiarLimpezaPositivo (limiarLimpezaNegativo) é a maior freqüência relativa de casos positivos (negativos) dos *clusters*.

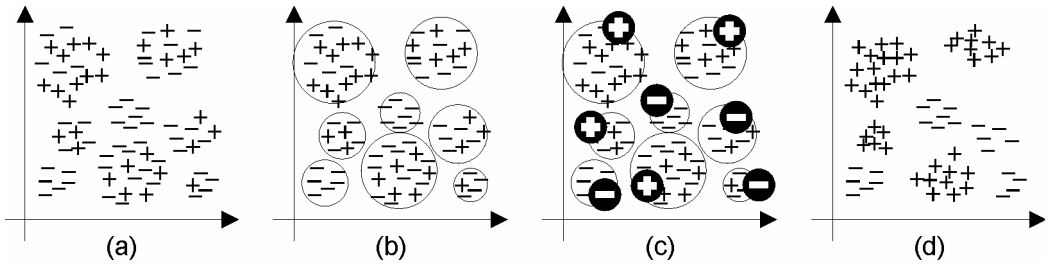

Figura 1. Funcionamento do método *C-clear*.

Como abordagens relacionadas, podemos citar os algoritmos propostos em Nickerson *et al.* (2001) e em Batista *et al.* (2004). O primeiro, *Cluster-Based Oversampling*, utiliza a aprendizagem não supervisionada para encontrar os *clusters* de cada classe e replicar os casos de cada *cluster* até que todos tenham a mesma quantidade de casos, tendo por objetivo melhorar a aprendizagem para casos raros[2]. Essa abordagem de *oversampling* pode produzir classificadores muito específicos para os casos replicados e com baixo poder de generalização para outros casos positivos. O algoritmo proposto em Batista *et al.* (2004) utiliza as técnicas de limpeza *Tomek links* e

[2] um caso é considerado raro quando a distribuição dos valores dos atributos é muito assimétrica, ocorrendo, assim, desbalanceamento intraclasse.

ENN para remover casos ruidosos localizados no lado errado da zona de decisão. A remoção de casos ruidosos pode ajudar a formar classes melhor definidas, e resultar em melhora na qualidade preditiva de classificadores. Essas técnicas de limpeza se baseiam em decisões locais (vizinhança local) para decidirem quais casos devem ser eliminados. Por isso, casos válidos podem ser eliminados junto com casos ruidosos. A motivação para o desenvolvimento do *C-clear* foi propor um algoritmo de limpeza que lidasse de forma adequada com os dois problemas citados acima, quais sejam: não produzir classificadores muito específicos e minimizar a eliminação de casos legítimos. Note que é provável ocorrer eliminação de casos legítimos se o domínio apresentar sobreposição natural de classes.

Algoritmo 1: ***C-clear***

Descrição das variáveis:
S: conjunto de dados de entrada;
$C = \{C_1, C_2, ..., C_n\}$: *clusters* retornados pelo algoritmo de clusterização;
$P_i \subseteq C_i$: casos da classe positiva do *cluster* C_i;
$N_i \subseteq C_i$: casos da classe negativa do *cluster* C_i;
buildClusters(.): chama a função de clusterização desejada;
fazerLimpeza: indica se deve ser feita limpeza de dados;
limiarSmote: freqüência positiva mínima de um *cluster* para oversampling;
limiarLimpezaPositivo: freqüência positiva mínima de um *cluster* para limpeza da classe positiva;
limiarLimpezaNegativo: freqüência negativa mínima de um *cluster* para limpeza da classe negativa;

Entrada: S
Saída: S limpo e balanceado

1. $C \leftarrow$ buildClusters(S);
2. Para cada $C_i \in C$ faça {
3. f = $|P_i| / |C_i|$;
4. Se f $\geq$ limiarSmote {
5. SMOTE(C_i, x, k);
6. }
7. }
8. $C \leftarrow$ buildClusters(S);
9. Se fazerLimpeza faça {
10. Para cada $C_i \in C$ faça {
11. f = $|P_i| / |C_i|$;
12. Se f $\geq$ limiarLimpezaPositivo {
13. Remova casos negativos de C_i.
14. }
15. f = $|N_i| / |C_i|$;
16. Se f > limiarLimpezaNegativo {
17. Remova casos positivos de C_i.
18. }
19. }
20. }
21. Retorna S

2.1 Implementação

O método *C-clear* foi implementado em Java 5 no módulo de pré-processamento do UnBMiner [Ladeira *et al.*, 2005]. O UnBMiner é uma plataforma e API abertas desenvolvidas para facilitar a implementação, avaliação de técnicas de mineração de

dados e construção de modelos. Esse software suporta parcialmente o modelo de referência CRISP-DM [SPSS, 1999] (do inglês, *CRoss Industry Standard Process for Data Mining*), abrangendo parte da fase de preparação dos dados (módulo pré-processamento do UnBMiner), fase de modelagem (criação de modelos baseados em *naïve Bayes*, CNM, árvore de decisão – algoritmos ID3 e C4.5 – e rede MLP *backpropagation*) e fase de avaliação (módulo de avaliação do UnBMiner). Para a obtenção dos *clusters* de bases com atributos contínuos, foi utilizado o algoritmo *k-means* [MacQueen, 1967]. Optou-se pelo *k-means* pela simplicidade de seu funcionamento e implementação. Para atributos categóricos, optou-se pelo *Squeezer* [He *et al.*, 2002] por possuir uma heurística para seu único parâmetro de entrada. Como freqüentemente bases de dados reais possuem tanto atributos contínuos quanto categóricos, foi implementado o *framework* de meta clusterização proposto por He *et al.* (2005), que utiliza o próprio *Squeezer* como meta clusterizador para bases de dados com atributos mistos. Por opção de projeto, o *C-clear* funciona com qualquer método de clusterização desejado, pois ele utiliza um vetor de inteiros como parâmetro de entrada que indica a qual *cluster* cada caso pertence, ao invés de chamar um método de clusterização como indicado no Algoritmo 1.

3. Experimento

O experimento consistiu em avaliar o ganho obtido (em relação ao classificador gerado com a base de dados original) com a utilização do *C-clear* aplicado à base de dados *Pima*, disponível no repositório da UCI[3] [Merz & Murphy, 1998] e freqüentemente utilizada na literatura. A base de dados *Pima Indians Diabetes* foi escolhida por ter sido uma das bases de dados em que a aplicação do método SMOTE não surtiu bons resultados [Chawla *et al.*, 2002]. Ela contém 768 casos, 9 atributos contínuos e uma classe binária. Esta base descreve os casos de diabetes na população indígena *Pima*, situada no Arizona, EUA. O algoritmo C4.5 [Quinlan, 1993], implementado no UnBMiner, foi escolhido para a geração dos classificadores por estar se tornando um padrão *de facto* para avaliação de algoritmos no âmbito de desbalanceamento de classes [Prati *et al.*, 2004] [Guo & Viktor, 2004].

O algoritmo C4.5 foi utilizado com a técnica de correção de Laplace [Ferri *et al.*, 2002]. Esta correção é uma técnica para melhorar a precisão na estimação de probabilidades com base na freqüência observada dos casos positivos sobre o número total de casos. No estudo em questão, as probabilidades a serem utilizadas para a construção da curva ROC, foram inferidas a partir do número de casos reportados pelo C4.5 em cada classe (*tem diabetes* ou *não tem diabetes*) representada nas folhas da árvore. Cada folha representa um tipo de caso da base de dados. Por exemplo, se uma folha apresenta apenas 1 caso positivo e 0 casos negativos e se outra folha apresenta 50 casos positivos e 0 casos negativos, a probabilidade de diabetes associada a ambas as folhas é de 1. No entanto, a probabilidade estimada a partir da folha que apresenta maior freqüência é mais precisa. Com a correção, o cálculo para essas probabilidades passa a ser, respectivamente, 2/3 e 51/52. Neste experimento, o C4.5 foi utilizado sem poda. De acordo com Batista *et al.* (2004), a aplicação de poda raramente resulta em melhora na

[3] University of California, Irvine

medida AUC[4], a qual foi a medida de qualidade adotada para se avaliar os modelos gerados nesse experimento. Um classificador que porventura tenha o maior AUC [Fawcett, 2004] não necessariamente é o melhor de todos, pois pode acontecer de ele ter pior desempenho em alguma região específica do espaço da ROC. Para tanto, a análise das curvas ROC proporciona escolher o melhor modelo independente da distribuição de classe, o que é desejado no domínio de desbalanceamento de classe [Fawcett, 2004]. Os classificadores gerados nesse experimento foram também analisados quanto às suas curvas ROC.

3.1 Detalhes do Experimento

Como o algoritmo *k-means* requer o informe do número *k* de clusters e a priori não se conhece esse valor, foi feita uma analise de sensibilidade para *k* variando de 3 a 15, de 3 em 3. Os resultados de AUC obtidos não foram significativamente diferentes ao nível de significância de 0,05. Os resultados a serem mostrados foram obtidos com $k = 15$, limiarSmote = freqüência positiva (0,35), limiarLimpezaPositivo = 0,7 e limiarLimpezaNegativo = 0,3. Um estudo sobre o impacto dos valores desses parâmetros e uma forma de aprendê-los a partir dos dados foram deixados como trabalho futuro. A intuição quanto ao elevado número de *clusters* utilizados foi amenizar a chance de se excluir *clusters* com alta representatividade e, desta forma, prevenir uma possível degradação do classificador resultante. A mesma intuição foi utilizada para os parâmetros limiarLimpezaPositivo e limiarLimpezaNegativo. Um valor demasiadamente baixo para limiarLimpezaPositivo removeria muitos casos da classe negativa, enquanto que um valor muito alto para o parâmetro limiarLimpezaNegativo removeria muitos casos positivos considerados como ruído. O *C-clear* foi avaliado de duas formas: com e sem limpeza. Desta seção em diante, o método *C-clear* grafado com o símbolo '-' (como em *C-clear*) refere-se à limpeza de dados *habilitada* e quando grafado com o símbolo '+' (como em *C+clear*) refere-se à limpeza de dados *desabilitada*.

A hipótese nula neste experimento – a sobreposição de classes degrada o desempenho de classificadores – foi testada com a avaliação de cinco modelos gerados a partir de cinco diferentes amostras da base Pima. Estas amostras foram geradas da seguinte maneira: a primeira consiste no conjunto original e as quatro demais foram formadas com o conjunto original pré-processado com os métodos oversampling aleatório, SMOTE, *C+clear* e *C-clear*. A partir dessas cinco amostras foram gerados cinco modelos com o C4.5, os quais foram avaliados com a validação cruzada de dez dobras estratificada (*stratified ten-fold cross-validation*), indicada por Kohavi (1995) como a forma mais eficiente para se selecionar modelos. Para se obter resultados mais realistas foram realizadas 40 rodadas de dez dobras e, a cada dobra, todas as cinco amostras foram geradas e avaliadas em conjunto. Ao final das 40 rodadas, foram computadas a média e o desvio padrão das 400 medidas AUC e dos pontos das 400 curvas ROC [Fawcett, 2004] geradas para cada um dos cinco classificadores. Esta forma de avaliação permite que todos os classificadores sejam gerados a partir de uma mesma dobra e, portanto, ameniza a ocorrência de distorções nos resultados. Os cinco classificadores gerados foram avaliados com respeito à medida AUC e à análise de curvas ROC. A Tabela 1, a seguir, apresenta as medidas AUC (média e desvio padrão

[4] do inglês, *Area Under the ROC* (*Receiver Operating Characteristic) Curve*.

expressos em percentagens) de cada um dos cinco modelos, nomeados conforme o método de amostragem utilizado.

Tabela 1. Resultados para cada amostra base do conjunto de dados Pima.

Modelos	AUC
Original	**76,74 (5,09)**
Oversampling	**75,89 (5,46)**
SMOTE	**75,71 (5,33)**
C+clear	**77,06 (5,36)**
C-clear	**78,37 (4,84)**

No trabalho de Batista *et al.* (2004), os métodos SMOTE e *oversampling* aleatório apresentaram ganho em relação aos dados originais. Provavelmente, a razão para tanto se deve ao fato deles terem utilizado o C4.5 com a modificação proposta por Ferri *et al.* (2002) para a construção dos pontos da curva ROC e da medida AUC. Desta forma, não há como comparar os resultados por eles obtidos com os obtidos com o *C-clear*. O valor AUC do modelo *C-clear* é significativamente melhor que o AUC dos demais modelos. O teste utilizado para comparar os AUC foi o de análise de variância (ANOVA) com nível de significância $\alpha = 0,05$. Na Figura 2, são apresentadas as curvas ROC dos modelos da Tabela 1 e da casca convexa[5] (envoltória dos pontos das curvas destes modelos). A curva ROC apresenta a relação entre as taxas de falso positivo (FP) e de verdadeiro positivo (TP) obtidas com a avaliação de um classificador. Um classificador é ótimo se sua curva ROC tende para uma reta vertical próxima à origem (menor taxa FP e maior taxa TP possíveis). Conforme Fawcett (2004), um classificador é potencialmente ótimo se, e somente se, sua curva ROC se sobrepõe à casca convexa. Na Figura 2, percebe-se que grande parte da curva ROC do classificador *C-clear* está sobreposta à casca convexa, enquanto que as curvas dos outros classificadores se situam abaixo dela. A curva ROC do classificador SMOTE se situa abaixo de todas as outras até o ponto onde a taxa FP é aproximadamente 0,3. Após este valor, ela se cruza com as demais curvas abaixo da curva do classificador *C-clear*.

4. Conclusões

Os resultados apresentados na Seção 4 corroboram a hipótese de que a aplicação do método SMOTE à base *Pima* produz o efeito indesejado de aumentar a sobreposição de classes. Além disto, o ganho na medida AUC promovida pela limpeza realizada com a aplicação do *C-clear* aos dados originais é evidência de que existe uma sobreposição natural de classes inerente à base *Pima*. Outro ponto observado foi o baixo desempenho obtido com a aplicação do SMOTE (pior curva ROC para taxas FP menores do que 0,5). Obviamente, com base de dados desbalanceada, onde a classe positiva é a minoritária, se deseja obter classificadores que apresentem altas taxas de verdadeiro positivo associadas a taxas de falso positivo mais baixas possíveis, pois estes últimos representam lixo de classificação. Deste ponto de vista, o SMOTE apresentou o pior resultado. A aplicação de SMOTE restrita aos *clusters* rotulados como positivos melhora o desempenho desse algoritmo e reforça ainda mais a hipótese de que há um aumento artificial da sobreposição de classes, o que afeta negativamente o desempenho do SMOTE.

[5] do inglês, *Convex Hull.*

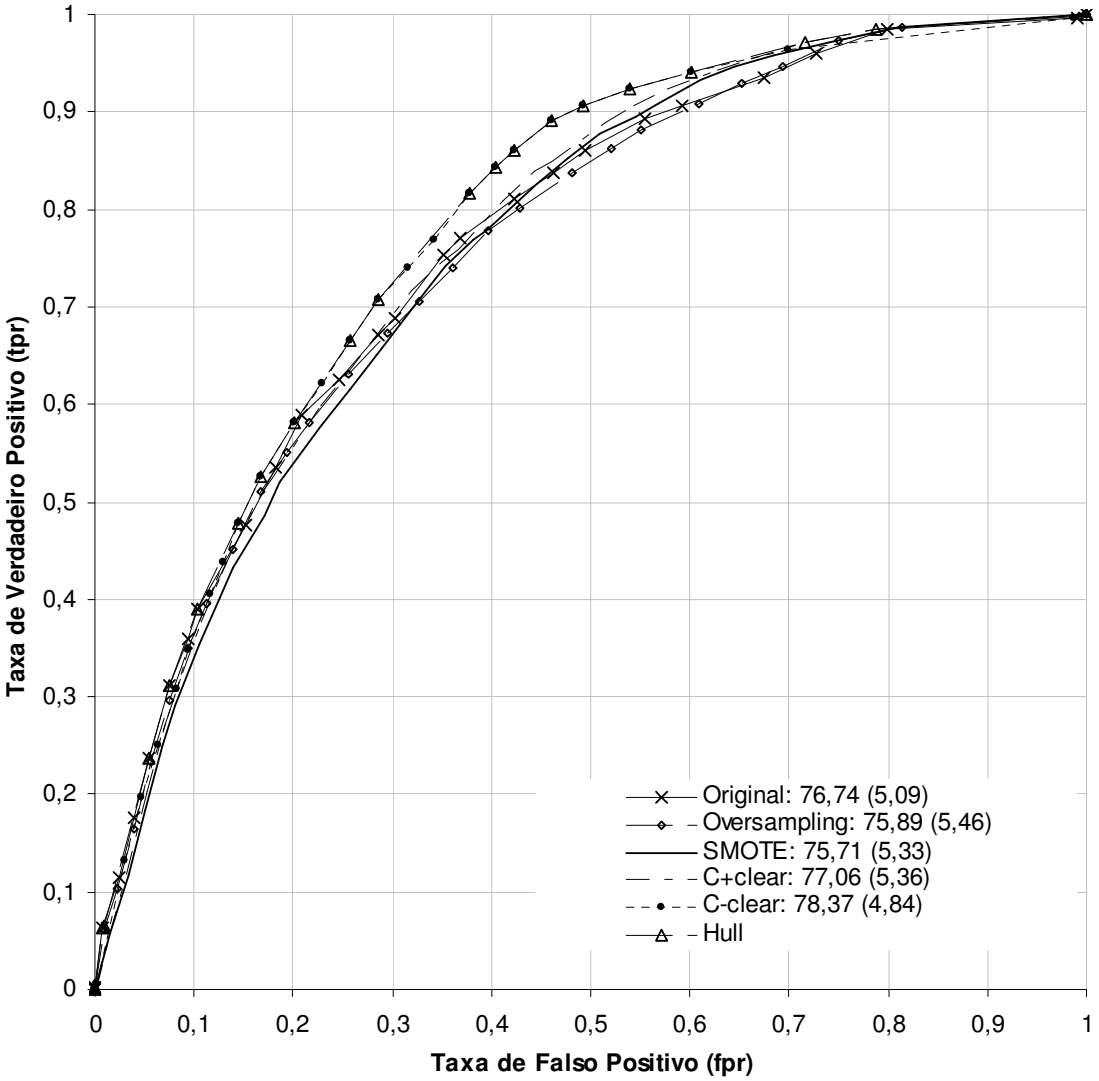

Figura 2. Curvas ROC dos classificadores selecionados.

Este artigo apresentou evidências de que a aplicação do SMOTE, algoritmo de amostragem bastante difundido na literatura, pode implicar no aumento artificial da sobreposição de classes em base de dados desbalanceadas. A principal contribuição foi a proposição do algoritmo *C-clear* que apresentou desempenho superior ao SMOTE na base de dados *Pima*, base esta freqüentemente utilizada para avaliação de algoritmos de mineração de dados. Mesmo assim, o experimento aqui realizado ainda não é conclusivo, pois a aplicação a apenas uma base de dados não é suficiente para conclusões mais definitivas.

4.1 Trabalho Futuro

A vantagem do *C-clear* é a sua independência de funcionamento em relação ao método de clusterização utilizado. Em contrapartida, ele é fortemente influenciado pela qualidade dos *clusters* por ele utilizados, como mencionado na Seção 2. Seria interessante a utilização de outros algoritmos de clusterização além do *k-means*.

O método *C-clear* ainda está em desenvolvimento e há muitos fatores para serem explorados ainda, como encontrar uma heurística para otimização do seu parâmetro de entrada ou fazer *undersampling* dos *clusters* ao invés da remoção total dos casos.

Agradecimentos

Esta pesquisa foi realizada com apoio parcial da CAPES (bolsa de mestrado e programas PROCAD e Cooperação Internacional CAPES/GRICES).

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, v.6 p.20-29.

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

Chawla, N.V.; Japkowicz, N.; Kotcz, A. (2004) Editorial: Special Issue on Learning from Imbalanced Data Sets. ACM SIGKDD Explorations. v.6. p.1-6.

Fawcett, T. (2004) "ROC Graphs - Notes and Practical Considerations", Machine Learning.

Ferri, C.; Flach, P.; Hernández-Orallo, J. H. (2002) Learning Decision Trees using the Area under the ROC curve. In C. S. A. Hoffman, editor, Nineteenth International Conference on Machine Learning (ICML). Morgan Kaufmann Publishers. p.139–146.

Guo, H.; Viktor, H.L. (2004) Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach. SIGKDD Explorations, v6 p.30-39

He, Z; Xu, X.; Deng, S.; (2002) Squeezer: an Efficient Algorithm for Clustering Categorical Data. *Jornal of Computer Science and Technology*. v.17, n.5, p.611-625

He, Z; Xu, X.; Deng, S.; (2005) Clustering Mixed Numeric and Categorical Data: a Cluster Ensemble Approach. *ArXiv Computer Science e-prints*. (Acesso em 12/12/2006. Disponível em: http://arxiv.org/ftp/cs/papers/0509/0509011.pdf)

Han, H.; Wang, W.Y.; Mao, B.H. (2005) Borderline-SMOTE: a New Over-Sampling Method in Imbalanced Data Sets Learning. Advances in Intelligent Computing. International Conference on Intelligent Computing (ICIC). Lecture Notes in Computer Science. V.3644, Springer-Verlag, Hefei (China) p.878-887

Japkowicz, N. (2002) Supervised Learning with Unsupervised Output Separation. In Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing (ASC). p.321-325.

Japkowicz, N. (2003) Class imbalances: Are we Focusing on the Right Issue? In Proceedings of the ICML Workshop on Learning from Imbalanced Data Sets (II).

Jo, T.; Japkowicz, N. (2004). Class Imbalances versus Small Disjuncts. *SIGKDD Explorations*, v.6, p.40-49.

Kohavi, R. (1995) A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Artificial Intelligence (IJCAI). p.1137-1145

Ladeira, M; Vieira, M.H.P; Prado, H.A; Noivo, R.M; Castanheira, D.B.S (2005). UnBMiner - Ferramenta Aberta Para Mineração de Dados. *Revista Tecnologia da Informação*, Brasília-DF, v.5, n.1, p.45-63.

MacQueen, J.B. (1967) Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press, v.1, p.281-297.

Merz, C.J.; Murphy, P.M. (1998) UCI Repository of Machine Learning Datasets. http://www.ics.uci.edu/~mlearn/MLRepository.html. (Acesso em 20/01/2007).

Nickerson, A.; Japkowicz, N.; Milios, E. (2001) Using Unsupervised Learning to Guide Re-Sampling in Imbalanced Data Sets. Proceedings of the Eighth International Workshop on AI and Statistics. p. 261-265.

Phua, C.; Alahakoon, D.; Lee, V. (2004). Minority Report in Fraud Detection: Classification of Skewed Data. *ACM SIGKDD Explorations*. v.6, p.50-59.

Prati, R.C.; Batista, G.E.A.P.A.; Monard, M.C (2004). Class Imbalances versus Class Overlapping: an Analysis of a Learning System Behavior. In MICAI, p. 312-321.

Quinlan, J.R. (1993). C4.5 Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

Sanches, M. K.; Monard, M. C. (2004). Proposta de um Algoritmo de Clustering Semi-supervisionado para Rotular Exemplos a Partir de Poucos Exemplos Rotulados. In: Workshop in Artificial Intelligence, Arica-Chile. Jornadas Chilenas de Computación. Chile : Sociedad Chilena de Ciencias de la Computación, v.1, p.1-9.

SPSS Inc., NCR Systems Engineering Copenhagen & DaimlerChrysler AG (1999). *CRISP-DM 1.0 – Step-by-step Data Mining Guide*. SPSS & CRISP-DM Consortium. (Acesso em 05/03/2005. Disponível em www.crisp-dm.org/CRISPWP-0800.pdf).

Wilson, D.R.; Martinez, T.R. (2000). Reduction Techniques for Exemplar-Based Learning Algorithms. Machine Learning. v.38, n.3, p 257-286.

Weiss, G. (2004) Mining with Rarity: A Unifying Framework. ACM SIGKDD Explorations v.6. p.7-19.