Ensembles de Classificadores para Bases de Dados Desbalanceadas: Uma Abordagem baseada em Amostragem Evolucionária

E. R. Q. Fernandes[1], A. P. L. de Carvalho[1], A. L. V. Coelho[2]

[1] Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação (ICMC)
everlandio@usp.br, andre@icmc.usp.br
[2] Universidade de Fortaleza - Programa de Pós-Graduação em Informática Aplicada (PPGIA)
acoelho@unifor.br

Abstract. Em muitos problemas práticos de classificação, o conjunto de dados a ser utilizado para a indução do classificador é significativamente desbalanceado. Isso ocorre quando a quantidade de exemplos de determinada classe é muito inferior à(s) da(s) outra(s) classe(s). Conjuntos de dados desbalanceados podem comprometer o desempenho da maioria dos algoritmos clássicos de classificação, uma vez que estes assumem uma distribuição de exemplos equilibrada entre as classes. Por outro lado, em diferentes cenários de aplicação, a estratégia de combinar vários classificadores em estruturas conhecidas como *ensembles* tem se mostrado bastante eficaz, levando a uma acurácia preditiva estável e, muitas vezes, superior àquela obtida por um classificador isoladamente. Nesse contexto, este trabalho propõe uma nova abordagem para lidar com conjuntos de dados desbalanceados, a qual utiliza *ensembles* de classificadores induzidos a partir de amostras balanceadas do conjunto de dados original. Para tanto, utiliza-se algoritmo genético multiobjetivo, que evolui a combinação dos exemplos que compõe as amostras balanceadas, levando em consideração a diversidade e o valor da área sob a curva ROC (AUC) dos classificadores induzidos por estas amostras.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications

Keywords: Classificação, Bases de Dados Desbalanceadas, Algoritmos Genéticos Multiobjetivos, *Ensembles* de Classificadores.

1. INTRODUÇÃO

Uma característica importante de várias bases de dados utilizadas para treinar classificadores em problemas reais está no desbalanceamento das classes. A distribuição desequilibrada de exemplos em cada grupo acontece naturalmente em alguns cenários de aplicação, como, por exemplo, o financeiro, em que o número de exemplos da classe de inadimplentes geralmente é muito menor (classe minoritária) que o número de casos pertencentes à classe de adimplentes (classe majoritária) (Marqués et al., 2013). A questão fundamental é que bases de dados desbalanceadas podem comprometer o desempenho da maioria dos algoritmos clássicos de classificação. Esses algoritmos assumem que as bases têm uma distribuição de exemplos equilibrada entre os grupos e que o custo por uma classificação errada é o mesmo para todas as classes (He and Garcia, 2009).

Uma primeira estratégia para lidar com esse problema é selecionar uma porção balanceada da base de dados de treinamento, com exemplos da classe minoritária e da classe majoritária, de tal modo a gerar um modelo de classificação que não prejudique a classe minoritária. Essa estratégia, porém, não é de todo eficaz, uma vez que a geração do modelo final de conhecimento pode não levar em conta instâncias relevantes para a discriminação entre as classes que ficaram de fora da porção selecionada, levando a uma queda na acurácia do modelo.

Por outro lado, *ensemble* de classificadores é um paradigma de aprendizado de máquina em que vários classificadores são treinados para resolver o mesmo problema. Em um *ensemble*, um conjunto de hipóteses é induzido separadamente, sendo combinado através de algum método/operador de consenso (Zhou, 2009). A habilidade de generalização de um *ensemble* é, em geral, maior que a dos classificadores isolados que o compõem, usualmente chamados de classificadores-base. Tumer and Ghosh (1996) apresentaram uma prova formal disso. Dietterich (1997) demonstra que a condição necessária e suficiente para que um *ensemble* tenha melhor taxa de acerto que seus classificadores-base é que estes sejam diversos entre si e tenham uma taxa de acerto superior a 50%. Dois classificadores são diversos entre si se cometem erros em exemplos distintos de um conjunto de teste. Por essa razão, pesquisadores têm desenvolvido medidas de diversidade dos classificadores-base, como é o caso das medidas *negative correlation learning* (NCL) (Liu and Yao, 1997) e *pairwise failure crediting* (PFC) (Chandra and Yao, 2006).

Desse modo, diversidade e acurácia são os dois critérios-chave a serem levados em consideração para se gerar *ensembles*. Conforme demonstrado formalmente por Krogh and Vedelsby (1995), um *ensembles* ideal é aquele que consiste de preditores com altas taxas de acerto e que ao mesmo tempo discordam tanto quanto possível. Existindo assim, um dilema (trade-off) sobre qual deve ser a medida ideal entre diversidade e acurácia, uma vez que são critérios geralmente conflitantes (Chandra and Yao, 2004). Para lidar com essa situação, algoritmos evolucionários multiobjetivos (AEMs) parecem ser uma alternativa interessante, uma vez que tratam inerentemente objetivos conflitantes no processo de aprendizagem (Jin and Sendhoff, 2008). Tais algoritmos evoluem, simultaneamente, um conjunto (fronte) de soluções não-dominadas (ou seja, soluções de compromisso) ao longo de dois ou mais objetivos, sem que seja requerida a preferência por um objetivo a priori. No caso de *ensembles* de classificadores, tais objetivos se traduzem nos critérios de acurácia e diversidade (Chandra and Yao, 2004).

Nesse contexto, este trabalho propõe uma nova abordagem para lidar com o problema de bases de dados desbalanceadas, a qual utiliza *ensembles* de classificadores induzidos a partir de amostras balanceadas do conjunto de dados de treinamento original. Para isso, um AEM customizado evolui a combinação de exemplos em amostras balanceadas sendo guiado pelos níveis de acurácia e diversidade dos classificadores induzidos por essas amostras, almejando-se daí um melhor desempenho preditivo do *ensemble* final.

2. ABORDAGENS NA LITERATURA

De forma geral, as abordagens que têm sido propostas para a rotulação de instâncias em problemas com classes muito desbalanceadas seguem dois caminhos distintos (Deepa and Punithavalli, 2010). Um deles é o de atribuir custos diferenciados às classes durante a indução do modelo de classificação (Zadrozny et al., 2003). O outro caminho se baseia em reamostragem de dados (subamostragem ou sobreamostragem). Na subamostragem, dados da classe majoritária são removidos, enquanto na sobreamostragem, dados da classe minoritária são replicados ou são gerados dados sintéticos.

Embora simples, a subamostragem realizada de forma aleatória pode desprezar dados úteis. Para contornar esse problema, uma subamostragem direcionada almeja detectar e eliminar uma fração menos representativa dos dados. Este é o caso da técnica *one-sided selection* (OSS) Kubat and Matwin (1997) que elimina da classe majoritária os exemplos redundantes, ruidosos e os próximos à fronteira de separação entre as classes. Já na sobreamostragem, a replicação dos exemplos tende a aumentar o custo computacional do processo (Sun et al., 2009). Com relação à geração de dados sintéticos, a técnica de interpolação é comumente usada, como é o caso do SMOTE (Synthetic Minority Oversampling Technique) (Chawla et al., 2002). SMOTE encontra os k vizinhos mais próximos de cada exemplo da classe minoritária e, daí, exemplos sintéticos são gerados ao longo do segmento de reta que liga esses exemplos aos seus vizinhos.

Com relação a abordagens de *ensembles* de classificadores, podemos destacar o *Bagging* (Breiman and Breiman, 1996) e *Boosting* (Schapire, 1990) (Freund and Schapire, 1997). *Bagging* treina um conjunto de classificadores-base com diferentes amostragens da base de treinamento. A amostragem é realizada com reposição e tem o mesmo tamanho da base de treinamento original. Após obter os resultados dos classificadores-base, *Bagging* combina-os por votação majoritária, e a classe mais votada é a predita. *AdaBoost*, o mais representativo algoritmo de *Boosting*, usa inteiramente a base de treinamento para criar classificadores em série, sendo que a cada iteração dá-se mais ênfase às instâncias que foram classificadas incorretamente na iteração anterior. Para isso, os pesos dos exemplos classificados incorretamente são aumentados e dos exemplos classificados corretamente diminuídos. Finalmente, quando uma nova instância é apresentada, cada classificador-base dá seu voto, ponderado por sua acurácia global, e o rótulo do exemplo é selecionado com base na maioria.

A seguir, serão discutidas algumas medidas de avaliação da qualidade de um classificador quando aplicado a bases de dados desbalanceadas para problemas de classificação binária.

Acurácia

Uma maneira eficaz de avaliar um classificador no contexto desbalanceado é utilizar as taxas de erros/acertos cometidos para cada classe (He and Garcia, 2009)(Sun et al., 2009)(López et al., 2013). Tais taxas podem ser obtidas a partir da matriz de confusão. Cada elemento dessa matriz fornece o número de exemplos cuja classe verdadeira é representada pela linha correspondente e cuja classe predita é representada pela coluna correspondente. Assim, os elementos ao longo da diagonal principal representam as decisões corretas, número de verdadeiros negativos (TN) e verdadeiros positivos (TP), enquanto os elementos fora da diagonal representam os erros cometidos, número de falsos positivos (FP) e falsos negativos (FN). A partir da Matriz de Confusão, é possível extrair duas medidas independentes, Taxa de Verdadeiros Positivos (TP_r = TP/(TP + FN)) e Taxa de Verdadeiros Negativos (TN_r = TN/(TN + FP)), que avaliam diretamente o desempenho sobre as classes positiva (minoritária) e negativa (majoritária) respectivamente.

Porém, a intenção é alcançar uma boa predição em ambas as classes, existe a necessidade de obter uma maneira de combinar essas medidas individuais, já que elas não são adequadas isoladamente. Conforme Lopez et al. (2013), uma maneira para unificar essas medidas e produzir um critério de avaliação é usar a curva *Receiver Operating Characteristic* (ROC) (Bradley, 1997). Essa curva permite visualizar a relação entre os benefícios (refletidos pela TPr) e os custos (refletidos pela FPr) da classificação, no que diz respeito à distribuição dos dados. Porém, para comparar vários modelos de classificação, utilizando as suas curvas ROC, só se pode afirmar que um modelo é melhor que o outro se a sua curva domina a outra. Então, faz-se necessário reduzir a curva ROC a um valor escalar, conhecido como a área sob a curva ROC (AUC) (Provost and Fawcett, 1997) (He and Garcia, 2009).

Diversidade

Como mencionado anteriormente, o sucesso de um *ensemble* depende fortemente da diversidade dos padrões de erros exibidos por seus classificadores-base. A diversidade de um *ensemble* pode ser induzida de diferentes modos; por exemplo, via uso de diferentes conjuntos de dados para treinamento dos classificadores-base. Elas podem ser divididas em dois grupos: as que consideram a diversidade de um par de classificadores por vez e daí obtém-se a diversidade média de todos os pares (*Pairwise Measures*); e as que consideram todos os classificadores juntos, calculando diretamente um valor de diversidade único para o ensemble (*Nonpairwise Measures*) (Kuncheva, 2004).

A medida *pairwise failure crediting* (PFC) (Chandra and Yao, 2006) foi utilizada nos experimentos com a nova abordagem proposta aqui por apresentar um bom desempenho quando do tratamento de bases de dados desbalanceadas (Bhowan et al., 2013). PFC mede os erros de cada classificador com relação a todos os outros de modo par a par. Em Bhowan et al. (2013), PFC é utilizada para calcular a diversidade do *ensemble* levando em consideração, separadamente, a classe minoritária e a classe majoritária.

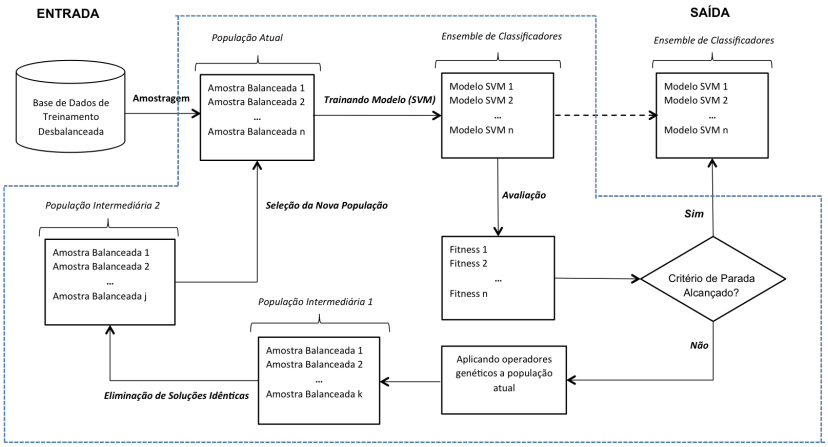

Fig. 1. Multiobjective Genetic Sampling (MOGASamp)

3. MÉTODO PROPOSTO

A figura 1 descreve o método proposto e nas subseções seguintes são detalhados cada passo.

Amostragem, Treinamento dos Modelos e Cálculo do Valor de Aptidão (fitness)

No primeiro passo, n amostras balanceadas são obtidas a partir da base de treinamento. Essa amostragem é realizada sem reposição dos exemplos. A escolha do tamanho das amostras é realizado com base na quantidade de exemplos da classe minoritária, de forma que reste pelo menos 10% dos exemplos dessa classe para realizar a validação das soluções durante o processo evolutivo (próximo passo). Cada amostragem representa um indivíduo na população do AEM, sendo representado (codificado) por um vetor indicando os exemplos da base de treinamento que fazem parte da amostra. Os indivíduos também contêm um modelo de classificação gerado pelo algoritmo SVM (*Support Vector Machines*) (Vapnik, 1995) treinado sobre tal amostra. Esse modelo SVM é utilizado para verificar a aptidão dessa amostra em gerar um modelo de classificação com os objetivos almejados (acurácia elevada e diversidade).

O modelo SVM de cada indivíduo é validado utilizando os exemplos da base de treinamento que não fazem parte da amostra. Assim, a métrica AUC é calculada com base no desempenho desse modelo sobre os dados de validação. Além disso, o PFC é calculado para cada indivíduo, comparando par a par as saídas do seu classificador sobre a base de treinamento com as saídas produzidas pelos classificadores dos demais indivíduos da população. Dessa forma, os dois objetivos almejados pelo método proposto são representados pelas métricas AUC e PFC. Sendo as melhores soluções aquelas que apresentam maiores valores nas duas métricas. Daí, essas métricas são utilizadas para compor um *rank* de não-dominância das soluções. Dizemos que uma solução x_1 domina uma solução x_2 quando x_1 não apresenta piores valores em nenhum dos objetivos avaliados, e x_1 apresenta melhor resultado em pelo menos um dos objetivos (Deb, 2001). Dessa forma, as soluções que não são dominadas por nenhuma outra recebe maior grau de não-dominância.

Seleção, Reprodução e Eliminação de Soluções Idênticas

O *rank* de não-dominância é usado para selecionar os indivíduos que irão gerar novos indivíduos através da aplicação dos operadores genéticos (reprodução e mutação). Essa seleção é executada usando torneio de tamanho 3. É selecionada uma quantidade de pais igual à quantidade de indivíduos da população. Para cada par de pais selecionados, dois novos indivíduos são gerados, juntando os exemplos da classe minoritária de um pai com os exemplos da classe majoritária do outro pai, e vice-versa. O operador de mutação gera uma pequena reestruturação em 5% dos filhos gerados. Onde

um trecho do vetor é selecionado aleatoriamente e os exemplos contidos nesse trecho são trocados por outros exemplos da base de treinamento de forma que o balanceamento da amostra se mantenha.

Após aplicar os operadores genéticos, indivíduos idênticos podem ocorrer, especialmente quando a taxa de desbalanceamento não é alta (menor que 1:6). Soluções idênticas com alto fitness têm maior probabilidade de serem selecionadas para reprodução e para as próximas gerações, gerando cada vez mais soluções idênticas. Como um dos pontos idealizados no *ensemble* de classificadores é que ele seja o mais diverso possível após a etapa de reprodução, as soluções idênticas são eliminadas.

Nova Geração e Critério de Parada

A seleção dos indivíduos que irão compor a nova geração é baseada no *rank* de não-dominância. Primeiro, os indivíduos com maior grau de não-dominância são selecionados, em seguida, os que são dominados apenas pelos primeiros, e assim por diante, até que o tamanho predefinido da população seja alcançado. O processo se repete por um número definido de gerações ou até que o melhor valor de AUC seja alcançado. Os modelos de classificação de todos os indivíduos da população final compõem o *ensemble* de classificadores. Quando um novo exemplo é apresentado ao *ensemble* final, este verifica a saída de todos os seus membros e a classe predita é decidida por voto majoritário.

4. EXPERIMENTOS

Para realizar os experimentos, foram utilizadas seis bases de dados de classificação binária com diferentes níveis de desbalanceamento. Essas bases foram obtidas do repositório UCI (*UCI Machine Learning Repository*) e estão sumarizadas na Tabela I. Para cada base de dados, metade dos exemplos de cada classe foram aleatoriamente escolhidos para formar o conjunto de treinamento e a outra metade para formar o conjunto de teste. Isso assegura que tanto o conjunto de treinamento quanto o de teste mantêm a mesma taxa de desbalanceamento entre as classes da base original.

Nome	YeastME1	YeastMit	YeastME3	Spect	Ion	German
Taxa de Desbalanc.	1:32,72	1:5,08	1:8,14	1:3,78	1:1,79	1:2,33
Total de Exemplos	1484	1484	1484	267	351	1000

Table I. Bases de Dados Desbalanceadas Usadas nos Experimentos

A avaliação do método proposto (MOGASamp) foi realizada comparando seu desempenho com aquele exibido por técnicas de reamostragem e classificação. As técnicas de reamostragem utilizadas foram: SMOTE; Subamostragem clássica (aleatória) e direcionada (OSS); e Sobreamostragem clássica. Já as técnicas de classificação utilizadas são *Bagging* e *Adaboost*. Para os algoritmos de reamostragem, após o processo de rebalanceamento das classes, foi utilizado o algoritmo SVM para gerar o modelo de classificação. As implementações do SMOTE, Subamostragem clássica, OSS e Sobreamostragem clássica existentes no pacote Unbalanced da linguagem de programação R (Pozzolo et al., 2014) foram utilizadas nos experimentos. Para executar *Bagging* e *AdaBoost*, foi utilizado o pacote Adabag (Esteban et al., 2013), ao passo que, para SVM, foi utilizado o pacote e1071 (Meyer et al., 2014).

O MOGASamp foi executado com uma população de 40 indivíduos e quantidade máxima de 20 gerações. As configurações do SMOTE foram: percentual de *oversampling* e *undersampling* igual a 200, quantidade dos k vizinhos igual a 5. Essas são as configurações padrões indicadas no pacote utilizado. A Subamostragem clássica e Sobreamostragem clássica foram executadas de forma que resultassem em bases balanceadas. Para o OSS não é necessário definir parâmetros, utilizando o método conforme descrito no pacote. Por fim, para o *Bagging* e o *AdaBoost* foram utilizadas o número de iterações igual 100, também como a configuração padrão do pacote.

5. RESULTADOS OBTIDOS

A Tabela II mostra os valores de AUC, taxa de Verdadeiros Positivos (classe Minoritária) e taxa de Verdadeiros Negativos (classe Majoritária) obtidos pelas técnicas utilizadas em cada base de dados trabalhada. Como as técnicas trabalhadas são métodos estocásticos, os valores apresentados são a média e o desvio-padrão dos valores obtidos em 30 execuções de cada algoritmo. Para cada problema, destaca-se o maior valor médio de cada medida utilizada.

		AUC	AccMin	AccMaj
YeastME1	MOGASamp	**0.960 [0.001]**	**1.000 [0.000]**	0.918 [0.003]
	SMOTE	**0.960 [0.008]**	0.995 [0.013]	0.929 [0.021]
	Sobreamostragem	0.831 [0.011]	0.675 [0.022]	0.988 [0.001]
	Subamostragem	0.951 [0.009]	**1.000 [0.000]**	0.902 [0.019]
	Bagging	0.772 [0.042]	0.547 [0.086]	**0.997 [0.001]**
	AdaBoost	0.835 [0.025]	0.675 [0.051]	0.995 [0.001]
	OSS	0.838 [0.000]	0.681 [0.000]	0.994 [0.000]
YeastMit	MOGASamp	**0.764 [0.007]**	0.627 [0.015]	0.900 [0.007]
	SMOTE	**0.764 [0.007]**	0.618 [0.019]	0.910 [0.008]
	Sobreamostragem	0.741 [0.005]	0.584 [0.010]	0.898 [0.010]
	Subamostragem	0.753 [0.008]	**0.648 [0.029]**	0.859 [0.022]
	Bagging	0.682 [0.009]	0.385 [0.019]	**0.979 [0.001]**
	AdaBoost	0.694 [0.008]	0.435 [0.016]	0.953 [0.003]
	OSS	0.736 [0.000]	0.508 [0.000]	0.964 [0.000]
YeastME3	MOGASamp	**0.910 [0.004]**	**0.913 [0.010]**	0.905 [0.008]
	SMOTE	0.885 [0.011]	0.822 [0.022]	0.949 [0.007]
	Sobreamostragem	0.862 [0.005]	0.785 [0.008]	0.938 [0.003]
	Subamostragem	0.896 [0.008]	0.894 [0.030]	0.898 [0.025]
	Bagging	0.885 [0.008]	0.802 [0.018]	0.968 [0.002]
	AdaBoost	0.858 [0.009]	0.751 [0.018]	0.966 [0.002]
	OSS	0.852 [0.000]	0.731 [0.000]	**0.974 [0.000]**
Spect	MOGASamp	0.680 [0.000]	0.408 [0.000]	0.953 [0.000]
	SMOTE	0.672 [0.006]	0.383 [0.018]	0.961 [0.006]
	Sobreamostragem	0.678 [0.005]	0.403 [0.011]	0.953 [0.002]
	Subamostragem	0.679 [0.005]	0.401 [0.013]	0.957 [0.005]
	Bagging	0.684 [0.010]	**0.512 [0.045]**	0.856 [0.037]
	AdaBoost	**0.699 [0.011]**	0.501 [0.022]	0.896 [0.019]
	OSS	0.666 [0.000]	0.370 [0.000]	**0.962 [0.000]**
Ion	MOGASamp	**0.966 [0.003]**	0.985 [0.003]	0.944 [0.004]
	SMOTE	**0.967 [0.003]**	**0.996 [0.007]**	0.938 [0.005]
	Sobreamostragem	0.959 [0.002]	0.950 [0.005]	0.968 [0.004]
	Subamostragem	0.949 [0.012]	0.986 [0.007]	0.912 [0.024]
	Bagging	0.906 [0.005]	0.872 [0.003]	0.941 [0.012]
	AdaBoost	0.924 [0.005]	0.865 [0.010]	**0.984 [0.006]**
	OSS	0.943 [0.000]	0.968 [0.000]	0.919 [0.000]
German	MOGASamp	0.960 [0.003]	**1.000 [0.000]**	0.917 [0.007]
	SMOTE	0.994 [0.001]	**1.000 [0.000]**	0.989 [0.002]
	Sobreamostragem	0.954 [0.005]	0.908 [0.011]	**1.000 [0.000]**
	Subamostragem	0.918 [0.048]	**1.000 [0.000]**	0.836 [0.090]
	Bagging	0.805 [0.005]	0.647 [0.008]	0.963 [0.004]
	AdaBoost	**1.000 [0.000]**	**1.000 [0.000]**	**1.000 [0.000]**
	OSS	0.994 [0.000]	0.995 [0.000]	0.993 [0.000]

Table II. AUC, Taxa de Verdadeiros Positivos, Taxa de Verdadeiros Negativos Utilizando Diferentes Técnicas de Reamostragem e Classificação

A área sob a curva ROC (AUC) foi utilizada como medida para avaliar o desempenho de cada abordagem em relação a ambas as classes. Pode-se observar que MOGASamp obteve os melhores resultados em quatro das bases de dados utilizadas. Outro importante aspecto é que o método

proposto não apresentou o pior valor de AUC para nenhum dos problemas. Isso é um indicativo de que MOGASamp pode ser aplicado a diferentes bases de dados, contendo diferentes taxas de desbalanceamento, mesmo quando não existir conhecimento a priori dos dados.

Com relação ao fator de erros/acertos cometidos para cada classe, pode-se observar na Tabela II que o método proposto apresenta grande proporção entre as taxas de Verdadeiros Positivos e Verdadeiros Negativos. Já sob o ponto de vista da taxa de Verdadeiros Positivos o método proposto apresenta excelente resultados. E que, esses resultados foram obtidos sem prejudicar a acurácia alcançada na classe majoritária. Como é o caso da técnica UnderSampling que quando alcança bons resultados na taxa de Verdadeiros Positivos apresenta baixo resultados na taxa de Verdadeiros Negativos.

Analisando sob o ponto de vista da taxa de Verdadeiros Negativos, observa-se que o método *Bagging* obteve resultados significativos. Porém, observando os resultados obtidos pelo *Bagging* com relação a taxa de Verdadeiros Positivos, verifica-se que o método apresenta baixo desempenho, o que indica que esse método não apresenta bons resultados em conjuntos de dados desbalanceados. Motivo para isso é que *Bagging*, assim como os algoritmos clássicos de classificação, assume que a base é balanceada. Situação semelhante, em menor grau, pode ser observada para os métodos *AdaBoost* e OSS.

6. CONCLUSÃO

O objetivo desse artigo foi apresentar uma nova abordagem evolucionária multiobjectivo, chamada *Multiobjective Genetic Sampling* (MOGASamp), para a classificação de dados em bases desbalanceadas. A MOGASamp usa a AUC e a medida de diversidade PFC como objetivos para evoluir um conjunto de amostras balanceadas da base de treinamento, de forma que essas amostras gerem classificadores com alta acurácia e diversidade. Por fim os classificadores resultantes do processo compõe um *ensemble* de classificadores para predição de novos exemplos através de voto majoritário.

Foram realizados experimentos com seis bases de dados desbalanceadas e os resultados obtidos pelo MOGASamp foram comparados com os obtidos por outras seis técnicas de reamostragem e classificação. Os resultados alcançados nos experimentos mostraram que o *ensemble* resultante do MOGASamp apresentou alta acurácia preditiva. Além disso, o MOGASamp também demonstrou elevada estabilidade para predizer dados de ambas classes. Ou seja, sem grandes diferenças na taxa de Verdadeiros Positivos e Verdadeiros Negativos, o que é comum acontecer quando se trata de algoritmos clássicos de classificação para bases de dados desbalanceadas.

Agradecimentos

Agradecemos o apoio financeiro da FAPESP.

Referências

Bhowan, U., Johnston, M., Zhang, M., and Yao, X. Evolving Diverse Ensembles using Genetic Programming for Classification with Unbalanced Data. *IEEE Transactions on Evolutionary Computation* 17 (3): 368–386, 2013.

Bradley, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition* vol. 30, pp. 1145–1159, 1997.

Breiman, L. and Breiman, L. Bagging predictors. In *Machine Learning*. pp. 123–140, 1996.

Chandra, A. and Yao, X. DIVACE: diverse and accurate ensemble learning algorithm. In *Intelligent Data Engineering and Automated Learning - IDEAL 2004, 5th International Conference, Exeter, UK, August 25-27, 2004, Proceedings*. pp. 619–625, 2004.

Chandra, A. and Yao, X. Ensemble learning using multi-objective evolutionary algorithms. *J. Math. Model. Algorithms* 5 (4): 417–445, 2006.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* vol. 16, pp. 321–357, 2002.

Deb, K. *Unbalanced: The package implements different data-driven method for unbalanced datasets.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

Deepa, T. and Punithavalli, M. An analysis for mining imbalanced datasets. *International Journal of Computer Science and Information Security* vol. 8, pp. 132–137, 2010.

Dietterich, T. G. Machine-learning research – four current directions. *AI MAGAZINE* vol. 18, pp. 97–136, 1997.

Esteban, A., Gamez, M., and Garcia, N. Applies multiclass AdaBoost.M1, AdaBoost-SAMME and Bagging, 2013.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55 (1): 119–139, Aug., 1997.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21 (9): 1263–1284, 2009.

Jin, Y. and Sendhoff, B. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (3): 397–415, 2008.

Krogh, A. and Vedelsby, J. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*. MIT Press, pp. 231–238, 1995.

Kubat, M. and Matwin, S. Addressing the curse of imbalanced training sets: One-sided selection. In *In Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann, pp. 179–186, 1997.

Kuncheva, L. I. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

Liu, Y. and Yao, X. Negatively correlated neural networks can produce best ensembles. *Australian Journal of Intelligent Information Processing Systems* 4 (3/4): 176–185, 1997.

López, V., Fernández, A., García, S., Palade, V., and Herrera, F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* vol. 250, pp. 113–141, 2013.

Marqués, A. I., García, V., and Sánchez, J. S. On the suitability of resampling techniques for the class imbalance problem in credit scoring. *JORS* 64 (7): 1060–1070, 2013.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., and Lin, C.-C. Misc Functions of the Department of Statistics (e1071), 2014.

Pozzolo, A., Olivier, C., and Bontempi, G. Unbalanced: The package implements different data-driven method for unbalanced datasets. http://mlg.ulb.ac.be, 2014.

Provost, F. J. and Fawcett, T. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *KDD*, D. Heckerman, H. Mannila, and D. Pregibon (Eds.). pp. 43–48, 1997.

Schapire, R. E. The strength of weak learnability. In *Machine Learning*. pp. 197–227, 1990.

Sun, Y., Wong, A. K. C., and Kamel, M. S. Classification of imbalanced data: a review. *IJPRAI* 23 (4): 687–719, 2009.

Tumer, K. and Ghosh, J. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition* vol. 29, pp. 341–348, 1996.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

Zadrozny, B., Langford, J., and Abe, N. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the Third IEEE International Conference on Data Mining*. ICDM '03. IEEE Computer Society, pp. 435–, 2003.

Zhou, Z.-H. Ensemble learning. In *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain (Eds.). Springer US, pp. 270–273, 2009.