Genetic Algorithms as a Pre Processing Strategy for Imbalanced Datasets

Marcelo Beckmann
beckmann.marcelo@gmail.com

Beatriz Souza L. P. de Lima
bia@coc.ufrj.br

Nelson F. F. Ebecken
nelson@ntt.ufrj.br

Federal University of Rio de Janeiro/COPPE, Technology Center, B-100,
Rio de Janeiro, RJ, Brazil

ABSTRACT

In data mining, the traditional classification algorithms tend to loose its predictive capacity when applied on a dataset which distribution between classes is imbalanced.

This work aims to present a new methodology using genetic algorithms, in order to create synthetic instances from the minority class. The experiments with the proposed methodology demonstrated a better classification performance in most of the problems, in comparison with other work in the specific literature.

Categories and Subject Descriptors

G.2.1 [Combinatory]: Combinatorial algorithms
I.5.2 [Design Methodology]: Classifier design and evaluation

General Terms

Genetic algorithms, Nature inspired algorithms, Pattern recognition, Classification, Imbalanced datasets.

Keywords

Classification, Genetic Algorithm, Imbalanced Datasets, Data Mining.

1. INTRODUCTION

Nowadays, the machine learning classification algorithms is largely used to understand the amount of data generated by automated processes. Nevertheless, most of the traditional classification algorithms ignore the imbalanced distribution among classes in the dataset and assume that the error cost in a class is the same as in the other classes.

This work presents a genetic algorithm (GA) for oversampling, that is, the creation of synthetic minority instances oriented by an evolutionary process. Basically the proposed GA aims to optimize the AUC measure obtained in the classification process, adjusting the positioning and size of regions inside the limits of the minority class (also known as positive class), so those regions are filled with synthetic positive instances to balance the dataset. The experiments demonstrated that the proposed GA provides a better classification performance, if compared with other results published in the specific literature.

2. THE IMBALANCE PROBLEM AND SOLUTIONS

In data mining [1], the classification algorithms on the presence of imbalanced datasets, in most of the cases, tend to classify the minority class instance as belonging to the majority class. Nevertheless, normally the class with lower number of samples is the most interesting and valuable to be identified, and this behavior can bring risks, financial and personal losses, if this prediction is incorrect. [7].

The scientific community agrees that this problem needs attention and solution [13]. In order to systematize the problem, [8] catalogued six causes associated with low performance in imbalanced datasets, which they concluded that not always the rarity of positive instances is the main cause of low performance**,** and according to the experiments in [5] and [10], the low performance rating in the data set is associated not only with imbalanced distribution of classes, but also with their overlap.

Over the past years several approaches have been adopted in the data mining community in order to tackle the causes of the problem [13][19], and to group such initiatives, [11] identified three main approaches: Algorithm level adjustment, cost sensitive learning and data level adjustment.

3. METHODOLOGY

This work proposes a data level adjustment that creates synthetic instances in the positive class, characterizing an oversampling strategy.

The Genetic Algorithm for Balancing (GAB) presented in this work, and previously in [20], adjusts regions position within the positive instances *P*. Those regions are randomly filled with synthetic instances. Thus, it is executed a distribution of instances in the training set *T*, compensating the imbalance between classes.

3.1 GA encoding

In the GAB, each individual is represented by a chromosome containing r regions. The chromosome *C* that encodes one solution in the GAB is composed of $t = r * n$ genes. For each region there's a set of n genes, where n is the number of attributes in the training set *T*. Each gene represents a minimum and maximum value of an attribute, and the r regions are randomly filled with synthetic instances. The problem shown in figure 1 contains two variables *(x, y)* of continuous numeric type. Table 1 presents the chromosome applied to this problem with $r=4$ regions, each one with $n=2$ genes, composing a total length $t = 8$.

Table 1 – Solution encoding: four regions and two attributes.

	Region 1		Region 2		Region 3		Region 4	
Attribute	**X**	**y**	**x**	**y**	**x**	**y**	**x**	**y**
Max	0.15	0.77	0.47	0.97	0.36	0.63	0.55	0.35
Min	0.04	0.45	0.11	0.87	0.22	0.14	0.45	0.17

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

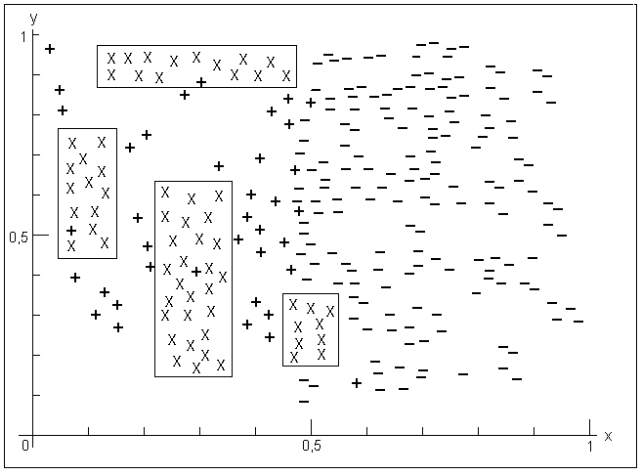

Figure 1 –Four regions filled with synthetic instances (x plots).

For categorical attributes, the gene must be adapted to contain a subset of existing values for this kind of positive attribute in the class, considering the same cannot be represented by minimum and maximum, because there is no previous or subsequent value. For example, there is no ascending ordering in the set that contains the MERCOSUL countries (Brazil, Argentina, Uruguay, Paraguay). Thus, if we add the attribute country in the chromosome of table 1, the new gene would be a new column containing a subset of the MERCOSUL set.

To avoid overfitting and region overlapping, it is not possible to create repeated instances given a minimum distance *dm*, which is the averaged euclidean distance between instances in the subset of positive instances *P*.

3.2 Crossover and mutation operators

The selection of individuals for crossover is done by roulette wheel [14][15][16]. To perform the crossover operator, a multiple crossover with two cut points randomly selected by region was used, generating two new individuals per operation. The simple mutation strategy [2] was applied, where part of the gene is selected to be inserted some random variety. For the mutation rate, the preliminary experiments did not obtain satisfactory results with a low rate of mutation, so, this rate was increased to 25% [23].

3.3 Fitness function

In GAB, the fitness function is the AUC [3][4][6], which is a known classification measure used in several benchmarks that represent a ROC curve as a single scalar.

4. EXPERIMENTS AND RESULTS

In order to prove the algorithm applicability, this section demonstrates and compares the GAB results in terms of AUC, obtained with the C4.5 algorithm [9] after balancing the datasets. The results are compared with the experiments published in [5], which is a study about imbalanced data sets that covers several methods for oversampling, and is considered a tough benchmark. The best results in [5] were obtained with SMOTE oversampling method [12], which also creates synthetic positive instances, combined with an undersampling method [21][22], which remove negative instances.

To make an equivalent benchmark comparison, the experiment steps described in [5] were also adopted in this work. The GAB experiments were performed over 15 datasets from UCI machine learning repository [17] used in [5]. All the experiments were done with two class problems. For multi-classes datasets, one class was selected, and the remainder ones were clustered into one class. The oversampling percent should be enough for a 50% class distribution. The use of four regions in the experiments showed better classification performance and stability.

The 15 datasets were balanced with GAB, and then scored with the C4.5 algorithm. A cross-validation with 10 folds [18] was performed for each execution, and the AUC average obtained as the measure of each folding. To evaluate the algorithm stability, 10 independent executions were performed, and the mean and standard deviation (in parentheses) for AUC obtained are displayed in table 2.

Table 2 – Comparison between GAB and the results published in [5].

	Batista & Monard, 2004		**GAB**
Dataset	**Best algorithm**	**AUC**	**AUC**
Pima	SMOTE	**85.49(5,17)**	82.72(0.48)
German	SMOTE+TOMEK	**81.75(4.78)**	76.99(0.82)
P-Operative	SMOTE+ENN	59.83(33.91)	**69.68(1.69)**
Habermann	SMOTE+ENN	**76.38(5.51)**	73.73(0.74)
Splice-ie	SMOTE	98.46(0.87)	**98,84(0.20)**
Splice-ei	SMOTE	**98.92(0.44)**	96.65(0.22)
Vehicle	SMOTE	98.96(0.98)	**99.01(0.32)**
Letter-vowel	SMOTE+ENN	**98.94(0.22)**	95.38(0.22)
New-thyroid	SMOTE+ENN	99.22(1.72)	**99.81(0.23)**
E.Coli	SMOTE+TOMEK	95.98(4.21)	**97.19(0.24)**
Satimage	SMOTE+ENN	**95.67(1.18)**	84.14(0.52)
Flag	SMOTE+ENN	79.32(28.83)	**86.03(1.86)**
Glass	SMOTE+ENN	92,90(7,30)	**96.81(1.62)**
Letter-a	SMOTE	99.91(0.12)	**99.95(0.09)**
Nursery	SMOTE+TOMEK	99.27(0.36)	**99,38(0.3)**

As can be seen in table 2, the GAB performed better in 9 of 15 problems, and the low values of standard deviations shows the robustness of the proposed algorithm.

5. CONCLUSION

This work presents an oversampling strategy using a genetic algorithm for balancing (GAB), which performs the dataset adjustment by the creation of synthetic positive instances targeted by an evolutionary process. The 15 datasets from UCI machine learning repository were balanced with the GAB and classified with the C4.5 algorithm. The results were compared with [5], and in most of the cases, the GAB has presented the best classification performance.

In future works, we intend to use the GAB in conjunction with undersampling strategies and other distance metrics applied to nominal attributes and multiclass problems.