A Proposal of Evolutionary Prototype Selection for Class Imbalance Problems

Salvador García[1], José Ramón Cano[2], Alberto Fernández[1] and Francisco Herrera[1]

[1] University of Granada, Department of Computer Science and Artificial Intelligence, E.T.S.I. Informática, 18071 Granada, Spain
E-mail: salvagl@decsai.ugr.es, alfh@ugr.es, herrera@decsai.ugr.es
[2] University of Jaén, Department of Computer Science, 23700 Linares, Jaén, Spain
E-mail:jrcano@ujaen.es

Abstract. Unbalanced data in a classification problem appears when there are many more instances of some classes than others. Several solutions were proposed to solve this problem at data level by under-sampling. The aim of this work is to propose evolutionary prototype selection algorithms that tackle the problem of unbalanced data by using a new fitness function. The results obtained show that a balancing of data performed by evolutionary under-sampling outperforms previously proposed under-sampling methods in classification accuracy, obtaining reduced subsets and getting a good balance on data.

1 Introduction

The class unbalance problem emerged when machine learning started being applied to the technology, industry and scientific research. A set of examples that will be used as input of classification algorithms is said to be unbalanced when one of the classes is represented by a very small number of cases compared to the other classes. In such cases, standard classifiers tend to be flooded by the large classes and ignore the small ones.

A number of solutions have been proposed at the data and algorithmic levels [1]. At the data level, we found forms of re-sampling such as over-sampling, where replication of examples or generation of new instances is performed [2]; or under-sampling, where elimination of examples is performed. At the algorithmic level, an adjust of the operation of the algorithm is carried out to treat with unbalanced data, see [3] for an example.

Various approaches of under-sampling methods were proposed in the literature considering two-classes problems, see [4] for review. Most of them are modifications of Prototype Selection (PS) algorithms [5].

Evolutionary Algorithms (EAs) [6] are stochastic search methods that mimic the metaphor of natural biological evolution. All EAs rely on the concept of *population* of individuals (representing search points in the space of potential solutions to a given problem), which undergo probabilistic operators such as *mutation*, *selection* and *recombination*. EAs have been used to solve the PS

problem with promising results [7]. Its application is denoted by Evolutionary Prototype Selection (EPS).

In this work, we propose the use of EAs for under-sampling unbalanced data sets ,we call it Evolutionary Under-Sampling (EUS), in order to improve balanced classification accuracy and distribution of classes. The aim of this paper is to present our proposal model and compare it with others under-sampling methods studied in the literature. To address this, we have carried out experiments with unbalanced data sets with distinct degrees of distribution of classes.

The remainder of the paper is divided into four sections. Section 2 summarizes the main characteristics of EUS. Section 3 briefly describes the previous under-sampling methods. Section 4 presents the way to evaluate classification systems in domains with unbalanced data sets. Section 5 discusses the methodology used in the experiments, as well as the results achieved. Finally, Section 6 concludes the paper.

2 Evolutionary Under-Sampling

Let's assume that there is a training set TR which consists of pairs $(x_i, y_i), i = 1, ..., n$, where x_i defines input vector of attributes and y_i defines the corresponding class label. TR contains n instances, which have m input attributes each one and they should belong to positive or negative class. Let $S \subseteq TR$ be the subset of selected instances resulted for the execution of an algorithm.

PS problem can be considered as a search problem in which EAs can be applied. To accomplish this, we take into account two important issues: the specification of the representation of the solutions and the definition of the fitness function.

- *Representation:* Let us assume a data set TR with n instances. The search space associated is constituted by all the subsets of TR. This is accomplished by using a binary representation. A chromosome consists of n genes (one for each instance in TR) with two possible states: 0 and 1. If the gene is 1, the its associated instance is included in the subset of T represented by the chromosome. If it is 0, this does not occur.
- *Fitness Function:* Let S be a subset of instances of T to evaluate and be coded by a chromosome. Classically, we define a fitness function that combines two values: the classification rate ($clas\_rat$) associated with S and the percentage of reduction ($perc\_red$) of instances of S with regards to TR [7].

$$Fitness(S) = \alpha \cdot clas\_rat + (1 - \alpha) \cdot perc\_red. \quad (1)$$

The 1-NN classifier is used for measuring the classification rate, $clas\_rat$, associated with S. It denotes the percentage of correctly classified objects from T using only S to find the nearest neighbor. For each object y in S, the nearest neighbor is searched for amongst those in the set $S \setminus \{y\}$. Whereas, $perc_red$ is defined as

$$perc\_red = 100 \cdot \frac{|TR| - |S|}{|TR|}. \quad (2)$$

The objective of the EAs is to maximize the fitness function defined, i.e., maximize the classification rate and minimize the number of instances obtained. The EAs with this fitness function will be denoted with the extension PS in the name.

In order to approach the unbalance data problem, EPS algorithms can be adjusted making use of a new fitness function defined as follows:

$$Fitness_{Bal}(S) = g - |1 - \frac{n_+}{n_-}| \cdot P, \tag{3}$$

where g is geometric mean of balanced accuracy defined in Section 4, n_+ is the number of positive instances selected (minority class), n_- is the number of negative instances selected (majority class), and P is a penalization factor.

This fitness function try to find subsets of instances making a trade-off between the classification balanced accuracy and an equal number of examples selected of each class. This second objective is obtained through the penalization applied to g in fitness value.

In this paper, we have applied this fitness function in two models of EAs. The first one, heterogeneous recombinations and cataclysmic mutation (CHC), is a classical model that introduces different features to obtain a tradeoff between exploration and exploitation [8], and the second one, PBIL [9], is a specific EA approach designed for binary spaces. We denote them as CHC-US and PBIL-US respectively.

3 Under-Sampling and Prototype Selection Methods

In this section, we describe the under-sampling methods and PS algorithms used in this study.

3.1 Under-Sampling Methods for Balance of Class Distribution

In this work, we evaluate six different methods of under-sampling to balance the class distribution on training data:

Random under-sampling: It is a non-heuristic method that aims to balance class distribution through the random elimination of majority class examples to get a balanced instance set.

Tomek Links [10]: It can be defined as follows: given two examples $E_i = (x_i, y_i)$ and $E_j = (x_j, y_j)$ where $y_i \neq y_j$ and $d(E_i, E_j)$ being the distance between E_i and E_j. A pair (E_i, E_j) is called Tomek link if there is not an example E_l, such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$. Tomek links can be used as an under-sampling method eliminating only examples belonging to the majority class in each Tomek link found.

Condensed Nearest Neighbor Rule (CNN-US) [11]: First, randomly draw one majority class example and all examples from the minority class and put these examples in S. Afterwards, use a 1-NN over the examples in S to classify the examples in TR. Every misclassified example from TR is moved to S.

One-sided Selection (OSS) [12]: It is an under-sampling method resulting from the application of Tomek links followed by the application of CNN-US.

CNN-US + Tomek Links [4]: It is similar to OSS, but the method CNN-US is applied before the Tomek links.

Neighborhood Cleaning Rule (NCL) [13]: Uses the *Wilsons Edited Nearest Neighbor Rule (ENN)* [14] to remove majority class examples. For each example $E_i = (x_i, y_i)$ in the training set, its three nearest neighbors are found. If E_i belongs to the majority class and the classification given by its three nearest neighbors contradicts the original class of E_i, then E_i is removed. If E_i belongs to the minority class and its three nearest neighbors misclassify E_i, the the nearest neighbors that belongs to the majority class are removed.

3.2 Prototype Selection Methods

Two classical models for PS are used in this study: DROP3 [5] and IB3 [15]. Furthermore, same two EAs used as EUS are employed as EPS with classical objective and we denote them as CHC-PS and PBIL-PS [7].

4 Evaluation on Unbalanced Data Classification

The most correct way of evaluating the performance of classifiers is based on the analysis of the confusion matrix. In Table 1, a confusion matrix is illustrated for a problem of two classes, with the values for the positive and negative classes. From this matrix it is possible to extract a number of widely used metric to measure the performance of learning systems, such as *Error Rate*, defined as $Err = \frac{FP+FN}{TP+FN+FP+TN}$ and *Accuracy*, defined as $Acc = \frac{TP+TN}{TP+FN+FP+TN} = 1 - Err$.

Table 1. Confusion matrix for a two-class problem

	Positive Prediction	*Negative Prediction*
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

Face to the use of error (or accuracy) rate, another type of metric in the domain of the unbalanced problems is considered more correct. Concretely, from

Table 1 it is possible to obtain four metrics of performance that measure the classification performance for the positive and negative classes independently:

- **False negative rate** $FN_{rate} = \frac{FN}{TP+FN}$ is the percentage of positive cases misclassified.
- **False positive rate** $FP_{rate} = \frac{FN}{FP+TN}$ is the percentage of negative cases misclassified.
- **True negative rate** $TN_{rate} = \frac{TN}{FP+TN}$ is the percentage of negative cases correctly classified.
- **True positive rate** $TP_{rate} = \frac{TP}{TP+FN}$ is the percentage of positive cases correctly classified.

These four performance measures have the advantage of being independent of the costs for class and prior probabilities. The goal of a classifier is to minimize the false positive and false negative rates or, in a similar way, to maximize the true positive and true negative rates.

In [16] it was proposed another metric called *Geometric Mean (GM)*, defined as $g = \sqrt{a^+ \cdot a^-}$, where a^+ denote accuracy on positive examples (TP_{rate}), and a^- is accuracy on negative examples (TN_{rate}). This measure try to maximize accuracy in order to balance both classes at the same time. It is an evaluation measure that joins two objectives.

5 Experiments and Results

Performance of the under-sampling and PS methods, described in Section 2 and 3 respectively, is analyzed using 7 data sets taken from the UCI Machine Learning Database Repository [17]. These data sets are transformed to obtain two-class non-balanced problems. The main characteristics of these data sets are summarized in Table 2. For each data set, it shows the number of examples (#Examples), number of attributes (#Attributes), name of the class (minority and majority) together with class distribution.

Table 2. Relevant Information about each data set used in this study

Data Set	#Examples	#Attributes	%Class (min., maj.)	%Class (min., maj.)
Ecoli	336	7	(iMU, Remainder)	(10.42,89.58)
German	1000	20	(Bad, Good)	(30.00,70.00)
Glass	214	9	(Ve-win-float-proc, Remainder)	(7.94,92.06)
Haberman	306	3	(Die, Survive)	(26.47,73.53)
New-thyroid	215	5	(hypo, Remainder)	(16.28,83.72)
Pima	768	8	(1,0)	(34.77,66.23)
Vehicle	846	18	(van, Remainder)	(23.52,76.48)

The data sets considered are partitioned using the *ten fold cross-validation (10-fcv)* procedure. The parameters of algorithms used are presented in Table 3.

Table 3. Parameters considered for the algorithms

Algorithm	Parameters
CHC-PS	$Pop = 50, Eval = 10000, \alpha = 0.5$
IB3	$Acept.Level = 0.9, DropLevel = 0.7$
PBIL-PS	$LR = 0.1, Mut_{shift} = 0.05, p_m = 0.02, Pop = 50$ $Negative_{LR} = 0.075, Eval = 10000$
CHC-US	$Pop = 50, Eval = 10000, P = 20$
PBIL-US	$LR = 0.1, Mut_{shift} = 0.05, p_m = 0.02, Pop = 50$ $Negative_{LR} = 0.075, Eval = 10000, P = 20$

Table 4. Class distribution after balancing

Balancing Method	**% Minority Class (Positive)**	**% Majority Class (Negative)**
CHC-PS	34.74	65.26
PBIL-PS	33.94	66.06
DROP3	45.10	54.90
IB3	33.95	66.05
CNN-US + TomekLinks	87.14	12.86
CNN-US	58.25	41.75
NCL	31.52	68.48
OSS	38.76	61.24
RandomUnderSampling	50.00	50.00
TomekLinks	29.29	70.71
CHC-US	*50.00*	*50.00*
PBIL-US	*49.99*	*50.01*

Table 4 shows class distribution after balancing with each method. Table 5 shows us the average of the results offered by each algorithm. Each column shows:

- The balancing method employed. *None* indicates that no balancing method is employed (original data set is used to classification with 1-NN).
- Percentage of reduction with respect to the original data set size.
- Accuracy percentage for each class by using a 1-NN classifier (a^+ and a^-), where subindex tra refers to training data and subindex tst refers to test data. GM value also is showed to training and test data.

Tables 4 and 5 are divided in three parts by separator lines: PS methods, Under-Sampling methods and proposed methods.

The following analysis of results can be made for these tables:

- CHC-US and PBIL-US present the best trade-off accuracy between both classes, they have the higher value of average GM in training and test (Table 5).
- The new fitness function used in EUS allows us to obtain a well-balanced class distribution (Table 4).
- There are algorithms that discriminates the negative class to a great extent, such as CNN-US+Tomek Links. PS algorithms discriminate positive class

because they only take into account the global performance of classification, which is highly conditioned for negative (majority) class examples.

Table 5. Average results

Balancing Method	% Red	$\%a^-_{tra}$	$\%a^+_{tra}$	GM_{tra}	$\%a^-_{tst}$	$\%a^+_{tst}$	GM_{tst}
None (1-NN)	-	89.34	57.69	70.46	88.71	56.19	69.38
DROP3	87.75	84.51	67.31	74.90	81.42	56.88	67.08
IB3	71.61	84.96	48.84	62.47	85.35	51.88	65.18
CHC-PS	**98.67**	94.88	57.07	67.04	**93.16**	50.11	61.73
PBIL-PS	95.48	**95.58**	61.19	69.78	91.59	53.81	63.49
CNN-US	61.70	79.08	63.00	69.41	81.26	63.53	70.76
CNN-US + TomekLinks	74.77	54.05	**91.66**	68.28	54.29	**89.16**	67.43
NCL	17.70	81.72	82.30	81.38	80.17	73.47	75.96
OSS	75.39	81.23	69.01	74.23	81.72	67.33	73.43
RandomUnderSampling	57.32	75.25	76.85	75.99	74.70	75.88	75.21
TomekLinks	11.88	85.64	76.50	80.15	83.77	68.58	75.19
CHC-US	95.45	82.46	88.78	85.54	79.17	75.91	77.48
PBIL-US	77.66	86.98	90.82	**88.87**	81.08	74.32	**77.56**

We have included a second type of table accomplishing a statistical comparison of methods over multiple data sets. Demšar [18] recommends a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers. One of them is Wilcoxon Signed-Ranks Test [19][20]. Table 7 collects results of applying Wilcoxon test between our proposed methods and the rest of Under-Sampling algorithm studied in this paper over the 7 data sets considered. This table is divided in two parts: In the first part, the measure of performance used is the accuracy classification in test set through geometric mean. In the second part, we accomplish Wilcoxon test by using as performance measure only the reduction of the training set. Each part of this table contains two rows, representing our proposed methods, and N columns where N is the number of algorithms considered in this study. Algorithms order is given at Table 6. In each one of the cells can appear three symbols: $+$, $=$ or $-$. They represent that the algorithm situated in that row outperforms ($+$), is similar ($=$) or is worse ($-$) in performance than the algorithm which appear in the column (Table 7).

Table 6. Algorithms order

Algorithm	Number	Algorithm	Number
DROP3	1	NCL	7
IB3	2	OSS	8
CHC-PS	3	RandomUnderSampling	9
PBIL-PS	4	Tomek Links	10
CNN-US	5	CHC-US	11
CNN-US+Tomek Links	6	PBIL-US	12

Table 7. Wilcoxon test

GM_{tst} *Accuracy Performance*												
	1	2	3	4	5	6	7	8	9	10	11	12
CHC-US (11)	+	+	+	+	=	+	=	=	=	=		=
PBIL-US (12)	+	+	=	=	=	+	=	=	=	=	=	
Reduction Performance												
	1	2	3	4	5	6	7	8	9	10	11	12
CHC-US (11)	+	+	-	=	+	+	+	+	+	+		+
PBIL-US (12)	-	=	-	-	+	=	+	=	+	+	-	

We make a brief analysis of results summarized in Table 7:

- Wilcoxon test shows us that our proposed algorithms are statistically equal to other Under-Sampling methods and outperforms PS methods, in terms of accuracy in test.
- However, in reduction performance, EUS obtain better reduction than non-evolutionary under-sampling methods. It points out that EUS provides reduced subsets without loss of balanced accuracy classification performance with respect to the rest of algorithms.
- Note that Wilcoxon test is performed using a low number of data sets (the minimum possible number of them to carry out the test). This implies that the results obtained may need more depth study (see for example the difference between CHC-PS and PBIL-US in GM test with similar statistical behavior). It is necessary to use more data sets in a future study.

6 Conclusions

The purpose of this paper is to present a proposal of Evolutionary Prototype Selection Algorithm with balance of data through under-sampling for imbalanced data sets. The results shows that our proposal is better analyzing the mean, equal statistically and better in reduction versus the remainder of under-sampling methods. Furthermore, a good balance of distribution of classes is achieved.

The paper also points out that standard Prototype Selection must not be employed to manage non-balanced problems.

Acknowledgement

This work was supported by TIN2005-08386-C05-01 and TIN2005-08386-C05-03.

References

1. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations **6** (2004) 1–6

2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence and Research **16** (2002) 321–357
3. Tan, S.: Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Systems with Applications **28** (2005) 667–671
4. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6** (2004) 20–29
5. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning **38** (2000) 257–286
6. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. SpringerVerlag (2003)
7. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in kdd: An experimental study. IEEE Transactions on Evolutionary Computation **7** (2003) 561–575
8. Eshelman, L.J.: The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: FOGA. (1990) 265–283
9. Baluja, S.: Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Pittsburgh, PA, USA (1994)
10. Tomek, I.: Two modifications of cnn. IEEE Transactions on Systems, Man, and Communications **6** (1976) 769–772
11. Hart, P.E.: The condensed nearest neighbour rule. IEEE Transactions on Information Theory **18** (1968) 515–516
12. Kubat, M., Matwin, S.: Addressing the course of imbalanced training sets: Onesided selection. In: ICML. (1997) 179–186
13. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: AIME '01: Proceedings of the 8th Conference on AI in Medicine in Europe, London, UK, Springer-Verlag (2001) 63–66
14. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics **2** (1972) 408–421
15. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6** (1991) 37–66
16. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition **36** (2003) 849–851
17. D.J. Newman, S. Hettich, C.B., Merz, C.: UCI repository of machine learning databases (1998)
18. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7** (2006) 1–30
19. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics **1** (1945) 80–83
20. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press (1997)