Evolutionary Cost-Sensitive Balancing: A Generic Method for Imbalanced Classification Problems

Camelia Lemnaru(✉) and Rodica Potolea

Computer Science Department, Technical University of Cluj-Napoca, Cluj-Napoca, Romania
{Camelia.Lemnaru,Rodica.Potolea}@cs.utcluj.ro

Abstract. Efficient classification under imbalanced class distributions is currently of interest in data mining research, considering that traditional learning methods often fail to achieve satisfying results in such domains. Also, the correct choice of the metric is essential for the recognition effort. This paper presents a new general methodology for improving the performance of classifiers in imbalanced problems. The method, *Evolutionary Cost-Sensitive Balancing* (*ECSB*), is a meta-approach, which can be employed with any error-reduction classifier. It utilizes genetic search and cost-sensitive mechanisms to boost the performance of the base classifier. We present evaluations on benchmark data, comparing the results obtained by ECSB with those of similar recent methods in the literature: *SMOTE* and *EUS*. We found that ECSB boosts the performance of traditional classifiers in imbalanced problems, achieving ∼45% relative improvement in true positive rate (TP_{rate}) and around 16% in F-measure (FM) on the average; also, it performs better than sampling strategies, with ∼35% relative improvement in TP_{rate} and ∼12% in FM over SMOTE (on the average), similar $textTP_{rate}$ and geometric mean (GM) values and slightly higher area under de curve (AUC) values than EUS (up to ∼9% relative improvement).

Keywords: Imbalanced classification · Meta-approach · Hybrid methodology · Genetic algorithms · Cost-sensitive

1 Introduction

One of the current important challenges in data mining research is classification under an imbalanced data distribution. This issue appears when a classifier has to identify a rare, but important case. Domains in which class imbalance is prevalent include fraud or intrusion detection, medical diagnosis, risk management, text classification and information retrieval [7], unexploded ordnance detection [1], or mine detection [33].

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation VI*, Advances in Intelligent Systems and Computing 674,
https://doi.org/10.1007/978-3-319-69710-9_14

A classification problem is imbalanced if, in the available data, a certain class is represented by a very small number of instances compared to the other classes [16]. In practice, the problem is addressed with 2-class problems; multi-class problems are translated to binary. As the minority instances are of greater interest, they are referred to as positive instances (positive class).

This paper presents a new general methodology for improving the performance of classifiers under imbalanced conditions. The method, Evolutionary Cost-Sensitive Balancing (ECSB), is a hybrid meta-approach which combines genetic search mechanisms with cost sensitive classification strategies. It involves the identification of the optimal cost matrix and parameter settings for the given problem, selected classifier (inducer) and evaluation metric. The method has been evaluated on benchmark data and compared to recently proposed methods for dealing with class imbalance, yielding significant performance improvements.

The rest of the paper is organized as follows: the next section reviews related work in this area. Section 3 details the proposed ECSB method, which is followed by its experimental validation in Sect. 4. Concluding remarks and future work are discussed in the last section.

2 Learning in Imbalanced Scenarios

Establishing how to assess performance is essential in imbalanced problems. The selection of an inappropriate measure may lead to unexpected predictions, which are not in agreement with the problem goals. This section presents the main evaluation metrics considered in imbalanced domains, a brief analysis of the limitations of traditional algorithms and an overview of existing techniques to tackle the imbalance.

2.1 Measuring Performance in Imbalanced Domains

The most employed evaluation measure for classification problems, the overall accuracy, is unfit in imbalanced domains [8,32], since the minority class contributes very little to its value. In highly imbalanced problems, a good recognition of the majority class translates into a high accuracy, regardless of how well the model identifies minority cases: for a data set with 99% examples for one class and 1% for the other, a model which classifies everything as belonging to the majority class yields 99% accuracy, while failing to identify any minority example.

For an imbalanced problem, the *true positive rate*, (TP_{rate}), also referred to as *recall* or *sensitivity*, is usually more important. However, there are other metrics, derived from the confusion matrix, which may also be relevant for assessing the performance in certain problems. A series of composite measures have been suggested by the scientific community for evaluating the performance in imbalanced problems: in [3,5,11] the *area under the ROC curve* (AUC) is employed; the *geometric mean* (GM) is proposed in [2] and employed in several other studies [11,13]; the *balanced accuracy* (BAcc) is another symmetric measure which

is more suited for imbalanced problems [4]; the *f-measure*, or *f-score* [8,13], and its generalization – the *f_β-measure* – provide a trade-off between the correct identification of the positive class and the cost of false alarms (in number of false positive errors). In [12] it is suggested that, in imbalanced problems, more attention should be given to sensitivity (TP_{rate}) than to specificity (TN_{rate}). In [8], the strategy to follow in imbalanced problems is to maximize the recall while keeping the *precision* under control. Both statements hold true in most imbalanced problems.

We argue that metric selection in imbalanced problems is essential for both model quality assessment and guiding the learning process. The metric should reflect the goal of the specific classification process, not just focus on the imbalance. Thus, if we are additionally dealing with imbalance at the level of the error costs, then associating a cost parameter to account for such disproportions is appropriate. If, on the other hand, the focus is on identifying both classes correctly, then an equidistant metric provides a fair estimation.

2.2 Existing Approaches for Dealing with Imbalance

The existing approaches for dealing with imbalanced problems can be split into: data-centered, algorithm-centered and hybrid solutions.

1. *Data-centered techniques* focus on altering the distribution of the training data: either randomly, or by making an informed decision on which instances to eliminate or add (by multiplying existing examples, or artificially generating new cases). Under this category we find random over- and under-sampling, or more elaborated approaches, such as Synthetic Minority Over-sampling Technique (SMOTE) [5], Tomek links [27], the Condensed Nearest Neighbor Rule (CNN) [14], One-Sided Selection (OSS) [17], the Neighborhood Cleaning Rule (NCL) [18], or Evolutionary Under-Sampling (EUS) [11]. In order to maximize the classification performance in the mining step, one should carefully match the appropriate sampling technique to the learning algorithm employed at that stage. Also, some methods require the analyst to set the amount of re-sampling needed, and this is not always easy to establish. It is acknowledged that the naturally occurring distribution is not always the best for learning [31]. A balanced class distribution may yield satisfactory results, but is not always optimal either. The optimal class distribution is highly dependent on the particularities of the data at hand.
2. *Algorithm-centered techniques*, also known as internal approaches, refer to strategies which adapt the inductive bias of classifiers, or newly proposed methods for tackling the imbalance. For decision trees, such strategies include adjusting the decision threshold at leaf nodes [24], adapting the attribute selection criterion [22], or changing the pruning strategy [36]. For classification rule learners, using a strength multiplier or different algorithms for learning the rule set for the minority class is proposed in [12], while for association rule learners, multiple minimum supports are employed in rule generation [21]. In [23], confidence weights are associated to attribute values (given a class

label) in a kNN approach. For SVMs, class boundary alignment is proposed in [35] and the use of separate penalty coefficients for different classes is investigated in [20]. Newly proposed methods, which deal with the imbalance intrinsically, include the biased minimax probability machine (BMPM) [15], or the infinitely imbalanced logistic regression (IILR) [33].

3. *Hybrid approaches* combine data- and algorithm-centered strategies. A number of approaches in this category consist of ensembles built via boosting, which also employ replication on minority class instances to second the weight update mechanism. Also, the base classifiers may be modified to tackle imbalanced data. Such approaches include SMOTEBoost [6], DataBoost-IM [13], and a complex SVM ensemble [26]. Another hybrid strategy is the one employed in cost-sensitive problems, to bias the learning process according to the different costs of the errors involved [10,25,37]. The method we propose in this paper falls into this category.

2.3 Limitations of Traditional Techniques

It is widely acknowledged that the nature of imbalanced problems is manifold. The essential data characteristic in such areas is the *imbalance ratio* (IR), i.e. the ratio between the number of instances in the majority (m_{Maj}) and minority classes (m_{Min}) – Eq. (1). Other data meta-features which have been shown to influence the behavior of classifiers in such domains are the *size* and the *complexity* of the data [16] and the *instances per attribute ratio* (IAR), i.e. the ratio between the total number of instances (m) and the number of attributes recorder per instance (n), which combines size and complexity information [19] – Eq. (2):

$$IR = \frac{m_{Maj}}{m_{Min}} \tag{1}$$

$$IAR = \frac{m}{n} \tag{2}$$

Also, particularities related to the distribution of the minority samples, such as too many "special cases" in the minority class, may affect the classifiers' capability to recognize all cases of interest (*within-class rarity*, *small disjuncts problem* [32]).

Several studies [16,29] indicate that most traditional classifiers are affected by the class imbalance problem to some extent. This is mainly because the assumptions followed in the training process don't usually hold in imbalanced problems. First of all, classifiers attempt to maximize accuracy, which is not an appropriate measure in imbalanced domains. Moreover, they assume the same distribution in the training and test samples, meaning that the model is customized for a certain distribution which is not the actual occurring distribution. Such a situation appears, for example, when dealing with dynamic distributions (such as the distribution of flu cases, which changes according to the season). Also, the rare cases may be very costly to obtain (in terms of time required, economic costs and/or pain). Moreover, even if the actual distribution is known, it may not be optimal for learning [30].

In [19], the authors perform a systematic analysis of the effect of class imbalance on the performance of six different classifiers, using 32 binary (or binarized) real-life benchmark data sets. The performance of all the classifiers evaluated seemed to be affected by the imbalance. Another conclusion of the study refers to the factors affecting classifier performance. The reduction in performance becomes more severe as the IR increases. However, for the same IR, larger IAR values are associated with improved classifier performance. Therefore, techniques for increasing the value of IAR (i.e. larger data set size and/or smaller complexity) may lead to an improved behavior.

3 Evolutionary Cost-Sensitive Balancing (ECSB)

The objective of the ECSB method is to improve the performance of a classifier (inducer) in imbalanced domains. It is a meta-methodology, which can be employed with any error-reduction classifier. Two strategies are simultaneously followed by the method: (1) use a cost-sensitive meta-classifier to adapt to the imbalance and (2) tune the base classifier's parameters. The outcome of the method is a tuple <M, S> for the triple <p, i, m>, where M is a cost matrix and S is the set of resulting parameter settings for the given problem (p), selected inducer (i) and evaluation metric (m). M is employed in conjunction with the cost-sensitive classifier, in order to build a more efficient classification model, focused on better identifying the underrepresented/interest cases. The search for M and S is performed through evolutionary mechanisms. The cost-sensitive component employs a meta-classifier to make its base classifier cost-sensitive, taking into account the misclassification costs. The main mechanisms for wrapping cost-sensitivity around traditional classifiers usually focus on employing a larger penalty for the errors on classes with higher misclassification cost, or modifying the training data such that the costly cases are proportionally better represented than the others.

The general flow of the method is presented in Fig. 1. The inputs are: the problem (p), translated in terms of a set of labeled examples (i.e. the training set), the base inducer (i) and the metric (m) to use for assessing the performance of i. The result of the method is a <M, S> tuple, which is used by a (meta-) cost-sensitive classifier to build the final classification model.

3.1 The Cost-Sensitive Component

Cost-sensitive learning encompasses several algorithms which focus on minimizing the total expected cost instead of the classifier error. A taxonomy of the types of costs involved in inductive concept learning can be found in [28], the most important being the *misclassification* and the *test costs*. The first category includes the costs which are conventionally considered by most cost-sensitive classifiers, and attempts to quantify the different impact that distinct errors produce. Several solutions address the second category also, which models different types of costs involved in acquiring the data (time, physical pain, money, etc.).

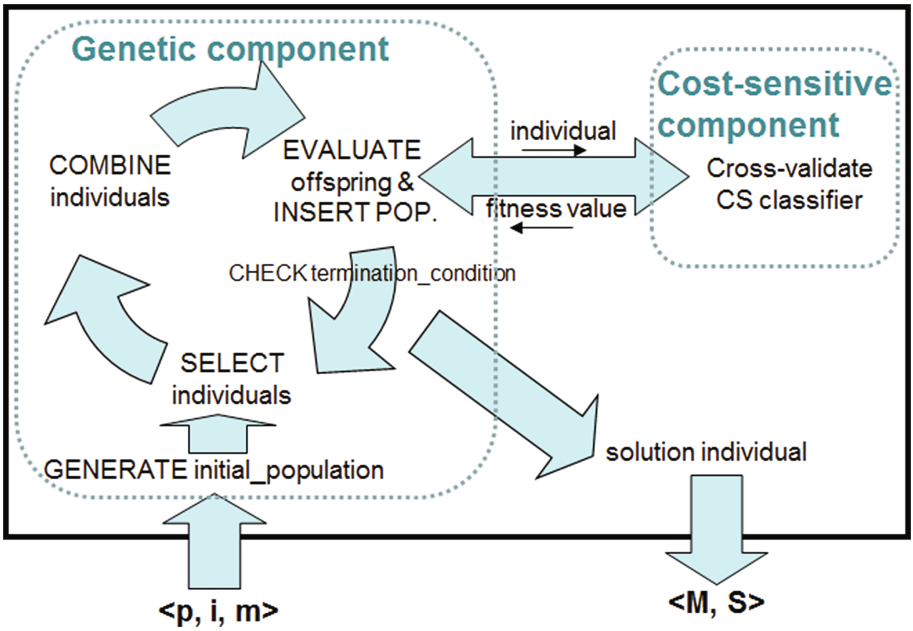


Fig. 1. General ECSB flow

We focus only on misclassification costs, since they can be employed to bias the learning process such as to provide a better identification for the minority class instances. The misclassification costs are represented via a cost matrix $(c_{ij})_{nxn}$, where c_{ij} represents the cost of misclassifying an instance of class j as being of class i. For imbalanced problems, we usually focus on binary classification, i.e. $n = 2$:

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \tag{3}$$

The main diagonal elements (c_{11} and c_{22}) represent the costs of correct identification and are normally smaller than or equal to 0 (i.e. reward or no penalty); c_{12} is the cost of a false negative (i.e. failing to identify a positive) and c_{21} captures the reverse situation. One of the most important difficulties when dealing with different error costs is to quantify misclassification costs. Even if it is relatively easy to determine which errors are more severe than others (e.g. in medical diagnosis $c_{12} > c_{21}$), it is difficult to quantify the gravity of an error exactly, since this may translate, indirectly, to more serious social/moral dilemmas, such a putting a price tag on human life.

In our approach, the cost matrix (M) for the given imbalanced problem is determined indirectly, following a genetic search. We can influence the result of the search by tuning the fitness function employed, which can be more easily translated, given a specific problem, than directly setting the cost matrix.

For example, it is more reasonable to state that the objective is to maximize both TP_{rate} and TN_{rate} in medical diagnosis, or to maximize precision in online advertising, than it is to set specific error costs.

The implementation of the cost-sensitive component has been carried out within the Waikato Environment for Knowledge Analysis (WEKA) framework [34]. Three cost-sensitive strategies have been considered:

(1) use an ensemble method to re-label the training instances according to the Bayes optimal prediction principle, which minimizes the conditional risk (*MC*) [10];
(2) reweight training instances according to the total cost assigned to each class (*CSr*) [34];
(3) predict the class with minimum expected misclassification cost, instead of the most likely class (*CS*) [34].

3.2 The Genetic Component

We have utilized the General Genetic Algorithm Tool for implementing the genetic component [9]. It provides the traditional genetic algorithms (GA) search organization, parent selection and recombination techniques. The specificity of our implementation is the problem representation and the fitness function(s) employed.

The search process starts with the initial population, i.e. a set of potential solutions, generated randomly (lines 1 and 2 in the pseudocode snippet below). By repeatedly applying recombination operators to some of the individuals in the population over a number of cycles, an element (or group of elements) is expected to emerge as a good quality approximate solution to the given problem (the loop between lines 3 and 9). Following a strategy similar to steady state evolution, in each cycle a number of new offspring is generated (additional pool). After evaluating their fitness (line 7), the fittest *p_size* individuals out of the old population and the additional pool (the newly generated offspring) will constitute the new population (line 8):

```
(1) population = generate_initial_population(p_size)
(2) evaluate_fitness (population)
(3) repeat
(4)    parents = select(population)
(5)    offspring = crossover(parents)
(6)    mutate(offspring)
(7)    evaluate_fitness (offspring)
(8)    insert (offspring, population)
(9) until (termination_condition)
(10) return best_individual
```

This strategy considers elitism implicitly. The search process stops when one of the following occurs: the optimal fitness value is reached, the difference between the fitness values of the best and the worst individuals in the current

population is 0, or a fixed (pre-determined) number of crossover cycles have been performed.

Each individual consists of four chromosomes: the first two representing each a misclassification cost (elements of M), and the last two representing parameters for the base classifier (elements of S). Although we have considered only two parameters for S – since most base classifiers used in the experiments have only two important learning parameters – the method can be extended to search for a larger number of parameters, depending on the tuned classifier. The first two chromosomes in the individual represent the meaningful coefficients of the 2×2 cost matrix. We assume the same reward (i.e. zero cost) for the correct classification of both minority and majority classes. Each chromosome consists of 7 genes, meaning that each cost is an integer between 0 and 127. We considered this to be sufficient to account even for large IRs. Gray coding is employed to ensure that similar genotypes produce close manifestations (phenotypes).

Fitness ranking is used to avoid premature convergence to a local optimum, which can occur if in the initial pool some individuals dominate, having a significantly better fitness than the others. Since establishing how to assess performance is essential in imbalanced problems and there is no universally best metric, which captures efficiently any problem's goals, we have implemented several different fitness functions, both balanced and (possibly) imbalanced. For consistency with the literature, we sometimes employ $\mathrm{TP}_{\mathrm{rate}}$ and sometimes recall for referring to the same measure:

$$\textbf{GM}(\text{geometric mean}) = \sqrt{TP_{rate} * TN_{rate}} \tag{4}$$

$$\textbf{BAcc}(\text{balanced accuracy}) = \frac{TP_{rate} + TN_{rate}}{2} \tag{5}$$

$$\textbf{FM}(f_{\beta}\text{-measure } = (1+\beta^2)\frac{prec * recall}{prec + recall} \tag{6}$$

$$\textbf{LIN}\,(\text{linear combination between}\mathrm{TP}_{\mathrm{rate}}, \mathrm{TN}_{\mathrm{rate}}) = \alpha * TP_{rate} + (1-\alpha) * TN_{rate} \tag{7}$$

$$\textbf{PLIN}\,(\text{linear combination between recall, prec.}) = \alpha * Recall + (1-\alpha) * Prec \tag{8}$$

4 Experimental Work

This section presents the experiments performed to validate the ECSB method and to compare it with recent proficient strategies. Subsect. 4.2 presents the general setup: it includes the evaluation methodology employed throughout the experiments, as well as the mechanisms and settings employed. Two different evaluation suites are then presented, with discussions of the results. A first set of tests evaluates comparatively the performance of different specializations of ECSB on large IR, small IAR data sets, since previous analyses [19] have shown that classifiers are most affected on such problems; the second presents a comparison between ECSB and a prominent under-sampling strategy for imbalanced data: Evolutionary Under-Sampling [11].

4.1 Experimental Setup

Experiments have been carried using 2-fold cross-validation. Generally, we have compared (1) the results of the base classifier with default settings (*Base*) with (2) the results obtained by the same classifier following data pre-processing with SMOTE [5] and default settings (*Base+SMOTE*), (3) the results obtained by the classifier following a parameter tuning stage, performed with the genetic component of ECSB (*ECSBT*) and (4) the results obtained by a classifier wrapped in our ECSB method (ECSB).

The specific mechanisms and setting values employed for the genetic component are presented in Table 1. Several fitness functions have been considered. No tuning has been performed on settings of the component so far. Five classifiers have been included in the experimental study, belonging to different categories: lazy methods (k-nearest neighbor, kNN), Bayesian methods (Naive Bayes, NB), decision trees (C4.5), support vector machines (SVM) and ensemble methods (AdaBoost.M1, AB). Table 2 describes the parameters considered for the base classifiers (in ECSB and ECSBT).

4.2 General Validation on Large IR, Small IAR Data Sets

We have performed a first analysis on benchmark data sets having large IR and small IAR values, as considered in [19] – Table 3. This combination of imbalance-related factors has a strong negative influence on the performance of classifiers. All three cost-sensitive strategies were considered (MC, CS and CSr), and five different fitness functions (GM, BAcc, FM with $\beta = 1$, LIN and PLIN, the last two having $\alpha = 0.7$).

This results in 15 combinations for the ECSB method, compared with the results obtained by the classifier alone (*Base*), the classifier with SMOTE (Base+*SMOTE*) and the classifier with tuned parameter values (*ECSBT*).

Table 1. Specific genetic mechanisms employed

Setting	Value
Population type	Single, similar to steady state
Initial population	Random
Population size	20
Additional pool	10
Crossover cycles	200
Parent selection	Roulette wheel
Recombination operators	Crossover: random crossover, 4 points
	Mutation: single bit uniform mutation, 0.2 rate
Fitness functions	GM; BAcc; FM; LIN; PLIN
Other	Fitness ranking
	Elitism, implicit with use of single population

Table 2. Base classifiers parameter ranges

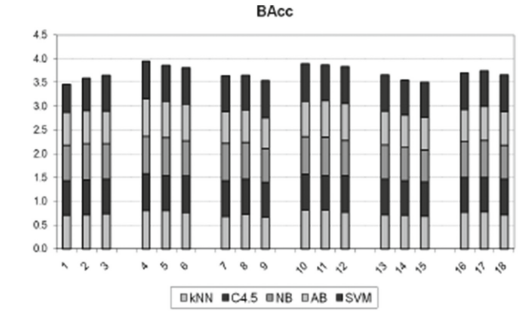
Classifier	Parameters	Type and range
kNN	K – number of neighbors	Integer between 1 and 10
C4.5	C – confidence ratio	Real, between 0 and 0.4
	M – min. number of instances per leaf	Integer, between 1 and 5
NB	n.a	n.a.
AB	P – weight threshold for weight pruning	Integer, between 1 and 127
	I – number of iterations	Integer, between 1 and 30
SVM	C – complexity	Real, between 1 and 100
	E – exponent	Integer, between 1 and 11

Table 3. Large IR, small IAR data sets

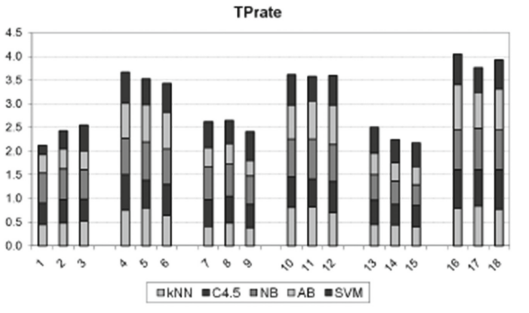
Dataset	#Examples	#Attributes	IR	IAR
Chess_IR5	2002	37	5	54
Ecoli_om_remainder_binary	336	8	15.8	42
Ecoli_imu_remainder_binary	336	8	8.6	42
Glass_VWFP_binary	214	10	11.59	21
German_IR10	769	21	10.14	37

The results are presented in Fig. 2. For viewing purposes, we have numbered the different methods from 1 to 18; please refer to the legend for identification. Each bar in the diagrams represents the overall average score (under the specific metric) obtained by all five classifiers, using the corresponding method. For example – in diagram (b), the first bar represents the overall average TP_{rate} obtained by all five classifiers on all data sets, under imbalance conditions, while the fourth bar represents the overall average TP_{rate} obtained by all five classifiers on all data sets obtained by ECSB using BAcc as fitness measure and CS as cost-sensitive strategy.

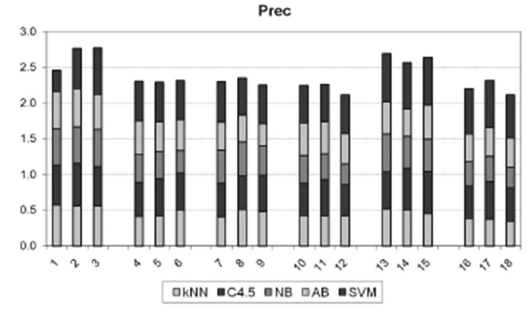
Several remarks can be made regarding these results: (1) using balanced metrics as fitness measures, such as GM or BAcc, produces significant improvements in the TP_{rate} (second and fourth groups in Fig. 2(b); (2) FM is not effective as fitness measure (third group in all diagrams); (3) the linear combination between TP_{rate} and TN_{rate} ($\alpha = 0.7$) as fitness function does not improve TP_{rate} significantly (fifth group in Fig. 2(b)), but instead it improves Prec (fifth group in Fig. 2(c)); (4) the linear combination between recall and precision ($\alpha = 0.7$) as fitness score yields the most important improvement in TP_{rate} (last group in Fig. 2(b)), but it degrades precision (Fig. 2(c)) – since $\alpha = 0.7$, more importance is given to improving recall than to precision; (5) for the SVM, both the TP_{rate} and the precision are significantly improved through the ECSB method (Fig. 2(b) and (c), the top portion of the bars); (6) out of the three cost-sensitive strategies



(a)



(b)



(c)

1 – Base
2 – Base+SMOTE
3 – ECSBT
4 – ECSB(CS, BAcc)
5 – ECSB(CSr, BAcc)
6 – ECSB(MC, BAcc)
7 – ECSB(CS, FM)
8 – ECSB(CSr, FM)
9 – ECSB(MC, FM)
10 – ECSB(CS, GM)
11 – ECSB(CSr, GM)
12 – ECSB(MC, GM)
13 – ECSB(CS, LIN)
14 – ECSB(CSr, LIN)
15 – ECSB(MC, LIN)
16 – ECSB(CS, PLIN)
17 – ECSB(CSr, PLIN)
18 – ECSB(MC, PLIN)

Fig. 2. Balanced accuracy, TP_{rate} and Precision obtained by the various methods on the large IR, small IAR data

evaluated, the most successful is CS (the first bar in each group from the second to the last), i.e. predict the class with minimum expected misclassification cost.

Therefore, balanced metrics (except FM) are generally appropriate as fitness measures for ECSB in imbalanced problems; when the recall is of utmost importance (e.g. medical diagnosis), using the linear combination between recall and precision, with a high value for α, is appropriate; this is also suitable when both precision and recall (TP_{rate}) are important (e.g. credit risk assessment), but with a lower value for α. Cost-sensitive prediction is the most appropriate strategy to employ.

4.3 Comparative Analysis with Evolutionary Under-Sampling

A second analysis was performed on a set of 28 imbalanced benchmark problems from [11], in order to compare our results with the performance of the Evolutionary Under-Sampling (EUS) strategy presented there. EUS has been shown to produce superior results when compared to state-of-the-art under-sampling methods, making it a good candidate for imbalanced data sets, especially with a high imbalance ratio among the classes. In this set of experiments, we have employed CS as cost-sensitive strategy and GM as fitness function – because it is the function employed in the most successful EUS model. We have also considered in the comparison the classifier with default settings (*Base*), the classifier with SMOTE and default settings (Base+*SMOTE*) and the classifier with tuned parameter values (*ECSBT*).

The results of this second analysis are shown in Tables 4 and 5. It can be observed that ECSB significantly boosts the performance of classifiers when compared to their behavior on the original problem (except for the AUC for AdaBoost.M1 – Table 5); on the average, there is ∼25% relative improvement on the GM and ∼5% on the AUC; the most significant improvements have been obtained for the SVM classifier (∼86% relative improvement on GM and 16% on AUC). Also, it yields significant improvements over SMOTE and ECSBT (∼17% and ∼14%, respectively, relative improvement on GM and ∼5% and ∼2%, respectively, on AUC). Slight improvements over the best EUS method

Table 4. Average GM (with standard deviations) obtained by the various methods

Geometric Mean (GM)										
	Best EUS		Base		Base+SMOTE		ECSBT		ECSB	
	mean	*stddev*	*mean*	*stddev*	*mean*	*stddev*	*mean*	*stddev*	*mean*	*stddev*
kNN	.797	.169	.731	.225	.744	.218	.762	.230	.817	.173
C4.5			.660	.317	.716	.254	.635	.307	.796	.179
NB			.754	.202	.771	.164	.754	.202	.814	.129
AB			.640	.314	.658	.306	.619	.323	.798	.188
SVM			.431	.401	.558	.358	.750	.213	.803	.184

Table 5. AUC (with standard deviations) obtained by the various methods

Area Under tin-Curve (AUC)										
	Best EUS		Base		Base+ SMOTE		ECSBT		ECSB	
	mean	*stddev*	*mean*	*stddev*	*mean*	*stddev*	*mean*	*stddev*	*mean*	*stddev*
kNN	.809	.170	.803	.144	.803	.144	.848	.140	.867	.128
C4.5			.797	.147	.797	.147	.786	.157	.830	.125
NB			.873	.110	.873	.110	.874	.111	.874	.111
AB			.892	.15	892	.105	.891	.098	.878	.121
SVM			.714	.175	.714	.175	.790	.143	.830	.132

have also been observed (i.e. the specialization of EUS which achieved the best performance in the above cited work): up to 9% relative improvement in AUC.

5 Conclusions and Future Work

Classification under imbalanced conditions is one of the current challenges in data mining research, triggered by the needs of specific application domains. All traditional algorithms are affected to some extent by the class imbalance problem. Also, the correct choice of the metric (or combination of metrics) to assess – and ultimately improve, is essential for the success of a data mining effort in such areas, since most of the time improving one metric degrades others.

A series of methods which deal with the class imbalance have been proposed in the literature over the last years. Sampling strategies are important because they can be used as pre-processing strategies. However, some approaches are difficult to employ by a less experienced user – e.g. some require to set the amount of sampling. Most importantly, to maximize their effect, they need to be matched to the specific classifier employed. Modifications to basic algorithms have also been proposed in the literature, with good performance improvements, but each is restricted to a specific class of techniques.

In this paper we propose a general hybrid strategy for improving the performance of classifiers in imbalanced problems. The method, Evolutionary Cost-Sensitive Balancing (ECSB), is a meta-approach, which can be employed with any error-reduction classifier. Two strategies are followed by the method simultaneously: tune the base classifier's parameters and use a cost-sensitive meta-classifier to adapt to the imbalance. A great advantage of the method, besides its generality, is that it needs little knowledge of the base classifier; instead, it requires specific knowledge of the domain to select the appropriate fitness measure.

We have performed several evaluations on benchmark data, to compare ECSB with current state of the art strategies for imbalanced classification. The results have demonstrated the following:

- ECSB significantly improves the performance of the base classifiers, achieving superior results to sampling with SMOTE or adapting the algorithm to the imbalance via evolutionary parameter selection;
- ECSB achieves superior results to current prominent approaches in literature: SMOTE and Evolutionary Under-Sampling;
- the most successful cost-sensitive strategy is predicting the class with minimum expected misclassification cost, instead of the most likely class (CS);
- balanced metrics are generally appropriate as fitness functions (except for the F-measure).

Our current focus is on adding an extra layer to the genetic search component, which will focus on finding the most suitable GA parameters for the given problem.

Acknowledgement. The work of the authors is supported by European Social Fund, via Programme POSDRU, DMI 1.5, ID 137516 – PARTING

References

1. Aliamiri A: Statistical Methods for Unexploded Ordnance Discrimination. PhD Thesis. Department of Electrical and Computer Engineering. Northeastern University. Boston, MA (2006)
2. Barandela, R., Sanchez, J.S., Garcia, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recogn. **36**(3), 849–85 (2003)
3. 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. ACM SIGKDD Explor. Newslett. **6**(1), 20—29 (2004). doi:10.1145/1007730.1007735
4. Brodersen, K.H., Ong, C.S., Stephen, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: Proceedings of the 20th International Conference on Pattern Recognition, pp. 3121–3124 (2010)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
6. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEboost: improving prediction of the minority class in boosting. In: Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 107—119 (2003)
7. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor. **6**(1), 1–6 (2004)
8. Chawla, N.: Data Mining from Imbalanced Data Sets: An Overview. Data Mining and Knowledge Discovery Handbook. Springer, US (2006)
9. Derderian, K.: General Genetic Algorithm Tool (2002), http://www.karnig.co.uk/ga/ggat.html
10. Domingos, P.: MetaCost: a general method for making classifiers cost-sensitive. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, pp. 155–164. ACM Press (1999)
11. Garcia, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol. Comput. **17**(3), 275–306 (2009)

12. Grzymala-Busse, J.W., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data. J. Intell. Manuf. **16**, 565–573 (2005)
13. Guo, H., Viktor, H.L.: Learning from imbalanced data sets with boosting and data generation: the databoost-IM approach. Sigkdd Explor. **6**, 30–39 (2004)
14. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. Inf. Theory **IT-14**, 515—516 (1968)
15. Huang, K., Yang, H., King, I., Lyu, M.R.: Imbalanced learning with a biased minimax probability machine. IEEE Trans. Syst. Man Cybern. B Cybern. **36**(4), 913–923 (2006)
16. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intell. Data Anal. J. **6**(5), 429–449 (2002)
17. Kubat, M., Matwin, S.: Addressing the course of imbalanced training sets: one-sided selection. In: ICML, pp. 179—186 (1997)
18. Laurikkala, J.: Improving Identification of Difficult Small Classes by Balancing Class Distribution. Technical Report A-2001-2. University of Tampere (2001)
19. Lemnaru, C., Potolea, R.: Imbalanced Classification Problems: Systematic Study. Issues and Best Practices. LNBIP, vol. 102, pp. 35–50 (2012)
20. Lin, Y., Lee, Y., Wahba, G.: Support vector machines for classification in nonstandard situations. Mach. Learn. **46**, 191–202 (2002)
21. Liu, B., Ma, Y., Wong, C.K.: Improving an association rule based classifier. In: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 504–509 (2000)
22. Liu, W., Chawlam, S., Cieslak, D., Chawla, N.: A robust decision tree algorithms for imbalanced data sets. In: Proceedings of the Tenth SIAM International Conference on Data Mining, pp. 766–777 (2010)
23. Liu, W., Chawla, S.: Class Confidence Weighted kNN Algorithms for Imbalanced Data Sets. Advances in Knowledge Discovery and Data Mining. LNCS, vol. 6635, pp. 345–356 (2011)
24. Quinlan, J.R.: Improved estimates for the accuracy of small disjuncts. Mach. Learn. **6**, 93–98 (1991)
25. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**(12), 3358–3378 (2007)
26. Tian, J., Gu, H., Liu, W.: Imbalanced classification using support vector machine ensemble. Neural Comput. Appl. **20**(2), 203–209 (2011)
27. Tomek, I.: Two modifications of CNN. IEEE Trans. Syst. Man Commun. **SMC-6**, 769—772 (1976)
28. Turney, P.: Types of cost in inductive concept learning. In: Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning. Stanford University, California (2000)
29. Visa, S., Ralescu, A.: Issues in mining imbalanced data sets-a review paper. In: Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, pp. 67–73 (2005)
30. Weiss, G.M., Provost, F.: The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML-TR-44. Department of Computer Science, Rutgers University (2001)
31. Weiss, G.M., Provost, F.: Learning when training data are costly: the effect of class distribution on tree induction. J. Artif. Intell. Res. **19**, 315–354 (2003)
32. Weiss, G.: Mining with rarity: a unifying framework. SIGKDD Explor. **6**(1), 7—19 (2004)
33. Williams, D., Myers, V., Silvious, M.: Mine classification with imbalanced data. IEEE Geosci. Remote Sens. Lett. **6**(3), 528–532 (2009)

34. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
35. Wu, G., Chang, E.Y.: Class-boundary alignment for imbalanced dataset learning. In: Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets (2003)
36. Zadrozny, B., Elkan, C.: Learning and making decisions when costs and probabilities are both unknown. In: Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining, pp. 204–213 (2001)
37. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. Knowl. Data Eng. **18**(1), 63–77 (2006)