Pattern Recognition 44 (2011) 1801–1810

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A multi-objective optimisation approach for class imbalance learning

Paolo Soda *

Medical Informatics and Computer Science Laboratory, Integrated Research Centre, University Campus Bio-Medico of Rome, Via Alvaro del Portillo, 21, 00128 Roma, Italy

ARTICLE INFO

Article history:
Received 26 July 2010
Received in revised form
12 January 2011
Accepted 21 January 2011
Available online 3 February 2011

Keywords:
Pattern recognition
Machine learning
Class imbalance learning
Multi-objective optimisation

ABSTRACT

Class imbalance limits the performance of most learning algorithms since they cannot cope with large differences between the number of samples in each class, resulting in a low predictive accuracy over the minority class. In this respect, several papers proposed algorithms aiming at achieving more balanced performance. However, balancing the recognition accuracies for each class very often harms the global accuracy. Indeed, in these cases the accuracy over the minority class increases while the accuracy over the majority one decreases. This paper proposes an approach to overcome this limitation: for each classification act, it chooses between the output of a classifier trained on the original skewed distribution and the output of a classifier trained according to a learning method addressing the course of imbalanced data. This choice is driven by a parameter whose value maximizes, on a validation set, two objective functions, i.e. the global accuracy and the accuracies for each class. A series of experiments on ten public datasets with different proportions between the majority and minority classes show that the proposed approach provides more balanced recognition accuracies than classifiers trained according to traditional learning methods for imbalanced data as well as larger global accuracy than classifiers trained on the original skewed distribution.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Class imbalance is considered a crucial issue in machine learning and data mining since most learning systems cannot cope with an imbalanced (or skewed) training set (Training set (TS)), i.e. a set where one of the classes is largely under-represented in comparison to the others. It is worth noting that the study of learning with imbalanced TS is a relevant issue since class imbalance exists in a large number of real-world domains, including text classification, currency validation, medical diagnosis, fraud detection, etc.

Studies concerning the class imbalance issue typically consider binary problems, and make the assumption that positive and negative samples belong to minority and majority classes, respectively. Traditional algorithms are biased towards the majority class, resulting in poor predictive accuracy over the minority one. This happens because they are designed to minimize errors over training samples, ignoring classes composed of few instances.

Research efforts addressing the course of imbalanced TSs can be traced back to the following four categories:

(1) undersampling the majority class so as to match the size of the other class [1–4];
(2) oversampling the minority class so as to match the size of the other class [1,3–7];
(3) internally biasing the discriminating process so as to compensate for class imbalance [8–11];
(4) multi-experts system [12–16].

Despite several proposals for each of these categories, balancing the recognition accuracies for each class very often harms the global accuracy, as reported in several papers [4,5,10,14,15] as well as confirmed by the experiments reported in this paper. In such cases, the accuracy over the minority class increases while the accuracy over the majority one decreases.

In order to overcome present limitations, in this work we propose an approach choosing between the prediction on sample x of a classifier trained on the original skewed distribution and the prediction of a classifier trained according to a learning method addressing the course of imbalanced TS, e.g. undersampling, oversampling, etc. This choice is driven by a parameter whose value maximizes two objective functions on a validation set, i.e. the global accuracy and the accuracies for each class. A series of experiments on ten public datasets with different proportions between the majority and minority classes show that the proposed approach provides more balanced recognition accuracies than classifiers trained according to traditional learning methods for imbalanced TS as well as larger global accuracy than classifiers trained on the original skewed distribution. Furthermore, the paper shortly reviews the literature on learning methods handling

* Tel.: +39 6 225419620; fax: +39 6 225419609.
E-mail address: p.soda@unicampus.it

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.01.015

imbalanced TS, providing also an extensive experimental comparison among them.

The paper is organised as follows: next session presents performance metrics used in case of imbalanced problems and then reviews the existing methods. Section 3 presents our approach, Section 4 describes the used datasets together with experimental results and the discussion. Section 5 provides concluding remarks.

2. Background and motivations

This section discusses metrics used to assess the performance in class imbalance learning, and reviews the related literature. Finally, it introduces the notion of classification reliability since it is used to present our approach.

2.1. Performance measures

The confusion matrix is used to evaluate the performance of a classification system since its elements describe the behaviour of the system. With reference to Table 1, we denote as $n^- = FP + TN$ and $n^+ = TP + FN$ the number of samples in the negative and positive classes, respectively.

Classification accuracy, defined as $acc = (TP + TN)/(n^- + n^+)$, is a traditional performance measure of a pattern recognition system.

However, in case of imbalanced TS the recognition performance cannot be measured in terms of classification accuracy only. For instance, consider a dataset where 2% of the samples are positive: in this case if the system labels all test samples as negative it will achieve an accuracy of 98%, but it will fail on all positive cases. It is straightforward observing that such a situation is clearly meaningless. Indeed, when the prior class probabilities are very different, measuring only the accuracy may lead to misleading conclusions since it is strongly biased to favour the majority class. Such an observation can be easily explained observing that class distribution is the relationship between the first and the second column of the confusion matrix. Any performance measure based on values from both columns will be inherently sensitive to class skew, as accuracy is.

Another consideration against using only the accuracy as a performance figure consists in observing that it assumes that the error costs are equal. This assumption is unreal in imbalanced problems where, typically, the cost of an error on the minority class, which is supposed to be the class of interest, is larger than an error on the majority one.

Hence, it would be more interesting to use a performance measure dissociating the hits (or the errors) that occur in each class. From Table 1 we can compute four metrics that independently estimate the performance on the two classes:

- *True positive rate* or *recall*, which is defined as $TP_{rate} = acc^+ = TP/TP + FN$;
- *True negative rate*, which is defined as $TN_{rate} = acc^- = TN/TN + FP$;
- *False negative rate*, which is defined as $FN_{rate} = FN/TP + FN$;
- *False positive rate*, which is defined as $FP_{rate} = FP/TN + FP$;

Table 1
Confusion matrix of a 2-classes problem.

	Actual positive	Actual negative
Hypothesise positive	True positive (*TP*)	False positive (*FP*)
Hypothesise negative	False negative (*FN*)	True negative (*TN*)

It is straightforward observing that $FN_{rate} = 1 - acc^+$ and that $FP_{rate} = 1 - acc^-$. Hence, two independent pairs are sufficient to characterise the performance of the classifier. Moreover, they are independent of prior probabilities and, thus, they are robust when class distribution might be different in training and test sets or change over time.

The area under the ROC curve (AUC) and the geometric mean of accuracies are two performance measures used in the literature on skewed datasets.

AUC measures the area under the ROC curve and is used as a synthetic measure to compare classifier performance [17]. Its values can range between 0 and 1 but no realistic classifier should have an AUC less than 0.5, which corresponds to random guessing the output class.

The geometric mean of accuracies is defined as $g = \sqrt{acc^+ \cdot acc^-}$. It increases if the accuracies of each class increase, while they are still balanced. Furthermore, g is a non-linear measure since a change in one of the two parameters has a different effect on g depending on its magnitude; for instance, the smaller the acc^+ value, the larger the g variation is. It is worth noting that g closely relates with the distance to perfect classification in the ROC space [13].

In conclusion, *acc* in conjunction with AUC or g (or both) can be used as performance metrics to describe classifier performance. On the one hand, *acc* measures the global recognition rate and, on the other hand, AUC or g characterises the behaviour of a classifier with respect to each class because they measure how much the classifier provides balanced decisions.

2.2. Learning techniques for class imbalance datasets

This section first summarises the four approaches reported in the literature addressing the course of imbalanced TS that are listed in Section 1, and then presents the motivations of this work.

The first and second approach for class imbalance learning is named as undersampling and oversampling, respectively. Both approaches resize the TS making the class distribution more balanced, so as to match the size of the other class [1–5].

Denoting as P and N the minority and majority training sets, undersampling methods sample a subset N' from N, with $|N'| < |N|$. It is usually $|N'| = |P|$. On the contrary, oversampling approaches generate a set P', with $|P'| = |N|$. P' is composed of all samples in P and others generated by the method.

Nevertheless, both undersampling and oversampling have relevant drawbacks. The former may remove potentially useful data, while the latter may increase the likelihood of overfitting due to samples random replication [2,5].

Besides such basic sampling methods, there are other approaches that work in more elaborate ways.

One-sided selection is an undersampling method that removes majority class samples, while it leaves untouched all cases of the minority class [5,7]. To this aim, majority class samples are divided into four groups: (i) samples suffering from class-label noise, (ii) borderline examples, which are close to the boundary between negative and positive regions, (iii) redundant samples, and (iv) safe samples that are worth being kept for classification. Borderline and noisy cases are detected by Tomek links [18], whereas redundant cases are defined as those not being in a consistent subset[1] of the training set. One-sided selection creates a TS composed by safe cases of majority class and by all cases of the minority one.

[1] A subset C of the training set S is said to be a consistent subset when the nearest neighbour rule using C correctly classifies samples in S.

Synthetic minority oversampling technique is an oversampling approach creating synthetic samples in the feature space along the line segments joining any/all of the k minority class nearest neighbours [2]. Depending on the amount of required oversampling, neighbours from the k nearest neighbours are randomly chosen. For instance, if we need the 200% of oversampling, we choose two of the k neighbours (with $k \geq 2$). Synthetic samples are generated following these steps: (i) compute the difference between the feature vector under consideration and its nearest neighbour, (ii) multiply the difference by a random number in [0,1], (iii) add this quantity to the feature vector under consideration. Such steps permit to select a random point along the line segment between two specific features.

Besides these two methods, the others reported in the literature apply different sampling strategies to achieve further improvement [1,6].

The third approach internally biases the discrimination-based process in order to compensate class imbalance [8–11]. In [8] the authors propose a weighted distance function to be used in the classification phase that compensates the TS imbalance without altering the class distribution since the weights are assigned to the respective classes and not to the individual examples. Ezawa et al. [11] biased the classifier in favour of certain attribute relationship, whereas Eavis et al. [9] presented a modified auto-encoder that allows for the incorporation of a recognition component into the conventional multi-layer Perceptrons mechanism. In [10] the authors adjust classifier sensitivity and specificity introducing different loss functions for positively and negatively labelled points.

The fourth learning approach for skewed TS is based on multi-experts system (MES), where each composing classifier C_i is trained on a subset of the majority class and on the whole minority class [12–16]. After sampling several subsets $N_1, N_2, \ldots, N_R$ from N, C_i is trained on $N_i \cup P$. Then, the decisions taken by all C_i on the test sample x are combined to set the final output. The rationale lies in observing that a MES generally produces better results than those obtained by individual composing experts [19,20]. Furthermore, base classifiers C_i are now trained on balanced sub-problems whose samples contain information on different aspects of the original set N. Note also that such an approach avoids the drawbacks of both under and oversampling.

In [13] the authors generated as many training subsets as required to get balanced subsets from the given TS. The number of subsets was determined as the ratio between the number of samples from the majority and minority classes. They employed nearest neighbour (NN) classifiers to build the MES, combining the outputs via the majority voting (MV) rule.

Molinara et al. in [14] presented a MES adopting Gentle AdaBoost as base classifiers. They used dynamic selection, mean or MV to aggregate individual classifier decisions. They tested two ways to divide the original TS: one is based on clustering and the other randomly selects majority class samples. The paper studied also how the behaviour of the MES varies when the number of base classifiers ranges from one up to the ratio between the cardinalities of the majority and minority classes. Note that a MES composed of one base classifier is equal to any classifier trained on the original imbalanced TS, which we refer to as IC in the rest of the paper. Best performance was achieved when the number of base classifiers is equal to $\lfloor |N|/|P| \rfloor$.

In [15] the authors proposed a MES composed by a fixed number of base classifiers that is independent of sample distribution in the TS. Base classifiers are 5-NN, C4.5 decision tree and Naïve Bayes whose outputs are combined via the MV rule.

In [16] the author reported how different MES combination rules perform when samples distribution is skewed, comparing ten fusion and selection criteria. Fusion methods attempt to determine the most likely class on the basis of the responses of C_i, whereas selection rules combine different C_i each of which is defined over a local region of the input space. These paradigms were tested applying two methods to divide N, namely random selection and clustering. The results showed that, on the one side, dynamic classifier selection with local accuracy [21] outperforms other selection and fusion combination methods and, on the other side, better performance was achieved when N is divided by random selection.

In [12] the authors proposed a method, named as BalanceCascade, that explores the majority class in a supervised way. It uses a cascade of Q base classifiers that are sequentially trained. For each step, the method randomly samples a subset N_i from N, with $|N_i| = |P|$, and train C_i with $N_i \cup P$. Then it removes majority class examples which are correctly classified by C_i. The cascade classifier is the conjunction of all $\{C_i\}_{i=1,\ldots,Q}$. A test sample is labelled as positive if and only if all C_i predict positive.

The results of the methods reported so far show that their performance, measured both in terms of g [5,7,8,12–16] and AUC [1,2,4,11,12], is more balanced than performance provided by an IC. Some papers reported also the values of acc or the pairs (acc^+, acc^-), which give us insight into the results and permit a more effective evaluation of the results. In these cases, we observe that (i) acc values provided by learning methods for imbalanced TS are lower than acc value provided by IC [4,5], (ii) acc^+ values measured for skewed TS learning methods are larger than acc^+ value provided by IC [10,14,15], (iii) acc^- values provided by learning methods for skewed TS are lower than the corresponding value yielded by IC [10,14,15]. Furthermore, our experiments reported in Section 4 confirm that balancing the accuracies for each class has the side effect of decreasing the global recognition rate.

In this respect, Section 3 presents a method to overcome this limitation: it aims at achieving balanced accuracies over the two classes without harming the global accuracy. This method can be applied with different learning approaches handling imbalanced datasets.

2.3. Reliability estimation

It is well known that information derived from classifier output permits to properly estimate the reliability of each classification act [19,20]. Reliability takes into account the many issues influencing the achievement of a correct classification, such as the noise affecting the samples domain or the difference between the objects to be recognized and those used to train the classifier. Without loss of generality, we can assume that the reliability of a sample x, denoted as $\phi(x)$ in the following, varies in [0,1]. A low value of $\phi(x)$ suggests that the decision on sample x is not safe since, for example, it can be a borderline instance or it can be affected by noise in the feature space. A large value of $\phi(x)$ suggests that the recognition system is more likely to provide a correct classification [19,20].

Note that, in general, the use of classification reliability does not limit the choice of classifier architecture since it is always possible to obtain a measurement, which can be used to compute $\phi(x)$, for each classification act of any kind of classifier [22].

3. Methods

This section first introduces our method and then shows that it provides an optimum solution according to multi-objective optimisation theory.

3.1. Reliability-based balancing method

The method is based on a classifier trained on the original skewed distribution and on a classifier built according to a class imbalance learning method. Let us introduce the following notation:

- IC is any classifier trained on the original skewed distribution that does not apply any learning methods for skewed TS, as reported in Section 2.2;
- BC is any classifier trained according to a traditional learning method addressing the course of imbalanced TS, e.g. undersampling, oversampling, etc.;
- x is a sample;
- $O_{IC}(x)$ is the label assigned by IC to sample x;
- $O_{BC}(x)$ is the label assigned by BC to sample x;
- $O(x)$ is the final label assigned by the proposed method to sample x;
- $\phi(x)$ is the reliability assigned by IC to sample x;
- $\bar{t}$ is a real number in [0,1].

On this basis, the final label is given by:

$$O(x)=\begin{cases} O_{IC}(x) & \text{if } \phi(x)\geq \bar{t} \\ O_{BC}(x) & \text{otherwise} \end{cases} \quad (1)$$

When the reliability provided by IC is larger than the threshold $\bar{t}$, the final label corresponds to the label returned by IC because it is reasonable to assume that IC is likely to provide a correct classification. When $\phi(x)$ is below $\bar{t}$, $O(x)$ is equal to the label assigned by a classification system trained according to a method specifically tailored for imbalanced TS. Indeed, in this case the value of the reliability suggests that the decision returned by IC should be not safe.

Hereafter, we will refer to this method as *Reliability-based Balancing* (RbB) since using either IC or BC depends on the reliability of sample classification. Fig. 1 schematically represents the RbB method, where the RbB block selects one of its inputs comparing $\phi(x)$ with the threshold $\bar{t}$.

The value of the threshold $\bar{t}$ is set so that it maximizes both *acc* and g on a validation set (Algorithm 1), thus providing the best global performance as well as the most balanced accuracies on this set. Samples are first divided into training, validation and test sets. Both IC and BC are trained on the training set; then they classify samples belonging to the validation set to determine the value $\bar{t}$ to be used with the test set. To this aim, we apply Eq. (1) and measure both g and *acc* for each value of a threshold t ranging in [0,1]. Indeed, g measures how much the accuracies over two classes are balanced, whereas *acc* estimates the global performance of the classification system. Representing g and *acc* on the X- and Y-axis, respectively, varying t generates a set of points that can be used to plot a curve, which is referred to as "g vs. *acc*" curve in the following. The curve extrema at $t=0$ and 1 correspond to IC and BC performance, respectively. It is straightforward noting that the ideal point in this plot is (1,1); hence, intuitively, the nearer the curve to this point, the better the performance obtained.

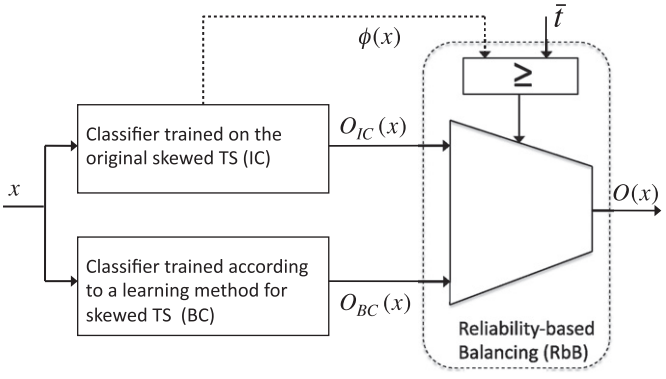

Fig. 1. Schematic representation of the proposed method.

Algorithm 1. *Reliability-based Balancing* algorithm

(1) Divide the labelled dataset $\mathbf{Z}$ into training, validation and test sets, denoted by $\mathbf{Z}_{Tr}$, $\mathbf{Z}_{Va}$, $\mathbf{Z}_{Te}$.
(2) Using $\mathbf{Z}_{Tr}$, train a classifier IC on the skewed distribution and train a classifier BC according to a traditional learning method addressing the course of imbalanced TS.
(3) Let T be a set of equally spaced values in [0,1].
(4) **for** all $t\in T$ **do**
  for all $x\in\mathbf{Z}_{Va}$ **do**
    Get $O_{IC}(x)$, $O_{BC}(x)$, $\phi(x)$
    if $\phi(x)\geq t$ **then**
      $O(x)\leftarrow O_{IC}(x)$
    else
      $O(x)\leftarrow O_{BC}(x)$
    end if
  end for
  From all $O(x)$ compute *acc* and g
  $\mathbf{p}(t)=[g,acc]$
 end for
(5) $\bar{t}=\text{argmin}_t(\|\mathbf{p}(t)-\mathbf{C}\|)$
  where $\mathbf{C}=[1,1]$
(6) **for** any $x\in\mathbf{Z}_{Te}$ **do**
  Get $O_{IC}(x)$, $O_{BC}(x)$, $\phi(x)$
  if $\phi(x)\geq\bar{t}$ **then**
    $O(x)\leftarrow O_{IC}(x)$
  else
    $O(x)\leftarrow O_{BC}(x)$
  end if
 end for

Formally, let $\mathbf{p}(t)$ be the pair of $g(t)$ and $acc(t)$ values measured on the validation set when the threshold t is used ($\mathbf{p}(t)=[g(t), acc(t)]$), and let $\mathbf{C}$ be the point with coordinates (1,1). The value $\bar{t}$ is given by:

$$\bar{t}=\arg\min_t(\|\mathbf{p}(t)-\mathbf{C}\|) \quad (2)$$

Hence, $\bar{t}$ is the value of t returning the pair (g, *acc*) closest to the north–east point in the plot. As we show in the next section, this choice corresponds to maximize both performance measures.

3.2. Multi-objective optimality

Let us consider the pair $P=(F,C)$, where $F=\{f_1(\boldsymbol{x}),\ldots,f_k(\boldsymbol{x})\}$ is a set of $k\geq 1$ objective functions $f_i:\Re^n\rightarrow\Re$, and $C=\{\boldsymbol{x}\in\Re^n : c_1(\boldsymbol{x})\leq 0,\ldots,c_m(\boldsymbol{x})\leq 0\}$, is a subset of $\Re^n$ defined by $m\geq 1$ constraints $c_i:\Re^n\rightarrow\Re$.

Definition. Any point $\boldsymbol{x}\in\Re^n$ is said to be an *admissible point* for P, iff $\boldsymbol{x}\in C$.

Definition. Given two admissible points $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ for P, we say that $\boldsymbol{x}^{(1)}$ *dominates* $\boldsymbol{x}^{(2)}$ *according to Pareto* (and write $\boldsymbol{x}^{(1)}\succ\boldsymbol{x}^{(2)}$) when:

(1) $f_i(\boldsymbol{x}^{(1)})\leq f_i(\boldsymbol{x}^{(2)})$ for any $i=1,\ldots,k$;
(2) $f_j(\boldsymbol{x}^{(1)})<f_j(\boldsymbol{x}^{(2)})$ for at least one $j\in\{1,\ldots,k\}$.

Definition. An admissible point $\boldsymbol{x}^*$ for P is said to be a *Pareto optimum* iff there exists no other admissible points $\boldsymbol{x}$ for P s.t. $\boldsymbol{x} \succ \boldsymbol{x}^*$.

Note that a Pareto optimum is an admissible point that tries to minimize all functions in F under the constraints c_i, $i = 1,\ldots,m$, at the same time, i.e. in some way it solves a multi-objective minimization problem over P.

According to the Fritz-John Theorem [23], the point $\overline{\boldsymbol{x}}$ is a Pareto optimum, if there exist $\boldsymbol{\lambda} \in \Re^k$ and $\boldsymbol{\mu} \in \Re^m$ s.t. the following system is satisfied:

$$\begin{cases} \sum_{i=1}^{k} \lambda_i \nabla f_i(\overline{\boldsymbol{x}}) + \sum_{j=1}^{m} \mu_j \nabla c_j(\overline{\boldsymbol{x}}) = 0, \\ \sum_{j=1}^{m} \mu_j c_j(\overline{\boldsymbol{x}}) = 0, \\ \lambda_i \geq 0,\ i = 1,\ldots,k, \\ \mu_j \geq 0,\ j = 1,\ldots,m, \\ (\boldsymbol{\lambda},\boldsymbol{\mu}) \neq (\boldsymbol{0},\boldsymbol{0}). \end{cases}$$

On this basis, we prove now that $\overline{t}$ is also a Pareto optimum of our problem. Indeed, we have two objective functions g and acc of the scalar variable t assuming values in [0,1]. Since we want to maximize as much as possible both functions, it seems reasonable to search for a value of t which is a Pareto optimum for the following multi-objective minimization problem:

$$\begin{cases} F = \{-g(t), -acc(t)\}, \\ C = [0,1] \end{cases}$$

where we have changed the sign of g and acc to take into account the fact that our original problem was a maximization one. Note that C may also be defined using the constraints on t:

$$\begin{cases} c_1(t) = t - 1 \leq 0, \\ c_2(t) = -t \leq 0. \end{cases} \tag{3}$$

For the above-mentioned theorem, a value $\overline{t}$ is a Pareto optimum of this problem iff there exists a quintuple $(\overline{\lambda_1},\overline{\lambda_2},\overline{\mu_1},\overline{\mu_2},\overline{t})$ satisfying the system:

$$\begin{cases} \lambda_1(-\dot{g}(t)) + \lambda_2(-\dot{acc}(t)) + \mu_1 - \mu_2 = 0, \\ \mu_1(t-1) - \mu_2 t = 0, \\ \lambda_1 \geq 0, \quad \lambda_2 \geq 0, \\ \mu_1 \geq 0, \quad \mu_2 \geq 0. \end{cases}$$

Now, choosing on the "g vs. acc" curve the point nearest to (1,1) does correspond to solving the following minimization problem:

$$\min((g(t)-1)^2 + (acc(t)-1)^2) \tag{4}$$

providing that $0 \leq t \leq 1$, which can be expressed again by the constraints (3).

According to the Kuhn–Tucker Theorem [24], solving this minimization problem is equivalent to solving the system:

$$\begin{cases} 2(g(t)-1)\dot{g}(t) + 2(acc(t)-1)\dot{acc}(t) + \mu_1 - \mu_2 = 0, \\ \mu_1(t-1) = 0, \\ t - \mu_1 = 0 \end{cases} \tag{5}$$

with $\mu_1 \geq 0$, $\mu_2 \geq 0$. This system may be rewritten as:

$$\begin{cases} 2(1-g(t))(-\dot{g}(t)) + 2(1-acc(t))(-\dot{acc}(t)) + \mu_1 - \mu_2 = 0, \\ (\mu_1 - \mu_2)t - \mu_1 = 0 \end{cases} \tag{6}$$

where the second equation is obtained as a difference between the second and third equation of (5).

Let the triple $(\overline{\mu_1},\overline{\mu_2},\overline{t})$ be a solution of system (6). Choosing $\overline{\lambda_1} = 2(1-g(\overline{t}))$ and $\overline{\lambda_2} = 2(1-acc(\overline{t}))$ implies that the quintuple $(\overline{\lambda_1},\overline{\lambda_2},\overline{\mu_1},\overline{\mu_2},\overline{t})$ satisfies (4), since $\overline{\lambda_1} > 0$ because of $0 < g(t) < 1$, and $\overline{\lambda_2} > 0$ because of $0 < acc(t) < 1$.

Hence the value $\overline{t}$ minimizing (4) with constraints (3) is a Pareto optimum of our multi-objective maximization problem.

4. Experimental evaluation

In this section we introduce the used datasets and the classifier configurations tested. Next, we present and discuss the results.

4.1. Datasets

We test the RbB approach on ten UCI datasets belonging to real-world problems [25]. These datasets vary both in number of features and in cardinality as well as in class distribution (Table 2). According to a general practice reported in previous works on imbalanced TS, for datasets having originally more than two classes we choose the class with fewer instances as the minority one and collapsed the others into the majority class. Hence, in the Glass set the problem was transformed to discriminate class 7 against all other classes, in the Ecoli dataset the task consists in classifying class 4 against the others, in the Vehicle set we aim at discriminating "opel" class against all others, in PageBlocks set the target class is the "horizontal line", in Yeast dataset the task consists in classifying class 4 (named as "ME3") against the others, and in the SatImage set the problem was transformed to discriminate class 4 (named as "damp grey soil") against all others.

For every dataset, we perform a 5-fold cross validation. The whole process is repeated ten times, averaging out the final values of g and acc.

4.2. Classifier configurations

As presented in Section 3, the RbB method chooses the final output between the predictions of IC and BC. In order to evaluate the RbB performance we compare nine recognition systems with different configurations. More specifically, the following five systems are initially used as single classifiers:

(1) IC: a classifier trained on the original skewed distribution which does not apply any learning methods for skewed TS, as reported in Sections 2.2 and 3;
(2) SMOTE: a classifier trained according to synthetic minority oversampling technique. In its implementation, we considered the three nearest neighbours, as suggested in [2];
(3) OSS: a classifier trained according to one-sided selection [5,7];
(4) MES-DCS: a multi-experts system where each classifier is trained on a subset $N_i \cup P$. According to results reported

Table 2
Summary of the used datasets.

Dataset	Number of samples	Number of features	Class distribution (%) (minority, majority)
Ecoli	336	7	(10.4, 89.6)
Glass	214	9	(7.9, 92.1)
Hepatitis	155	19	(20.8,79.2)
Pima	768	8	(34.8,65.2)
Phoneme	5404	5	(29.4, 70.6)
Breast Cancer Wisconsin	699	9	(34.5, 65.5)
Vehicle	846	18	(33.4, 66.6)
SatImage	6435	36	(10.8, 89.2)
PageBlocks	5473	10	(6.0, 94.0)
Yeast	1484	8	(10.9, 89.1)

in [16], we apply random selection to sample N and dynamic classifier selection with local accuracy [21] to combine the outputs of base classifiers;
(5) BaCa: a multi-experts system trained according to the BalanceCascade serial scheme [12].

We chose to test IC because it permits us to estimate the performance achievable under class skew, whereas classifiers 2–5 permit us to measure the performance achievable using "traditional" learning methods for skewed TS. Notice that traditional learning methods for skewed TS are referred to as BC in the rest of the paper, according to notation introduced in Section 3.

The same classifiers are then used as RbB components, giving rise to four further schemes:

(1) RbB method choosing the final output between the predictions of IC and SMOTE. In the following, this recognition system is named as RbB:IC+SMOTE;
(2) RbB method choosing the final output between the predictions of IC and OSS. Henceforth, this recognition system is referred to as RbB:IC+OSS;
(3) RbB method choosing the final output between the predictions of IC and MES-DCS. In the rest of the paper, it is referred to as RbB:IC+MES-DCS;
(4) RbB method choosing the final output between the predictions of IC and BaCa. Hereafter, this recognition system is named as RbB:IC+BaCa.

Both Support Vector Machines (SVMs) with a Radial Basis Function Kernel and AdaBoost with decision stumps for base hypotheses have been used as base classifiers. Hence, we tested 18 classification systems (nine recognition schemes per two choices of base classifiers), which allows us to analyse and compare not only the results of the proposed method, but also those related to traditional learning methods.

As remarks, the reliability of SVMs classifications is evaluated using the distance of the pattern x from the optimal separating hyperplane in the feature space induced by the chosen kernel [26], whereas the reliability of AdaBoost classifications is estimated using the magnitude of the final hypothesis [27].

4.3. Results and discussion

Tables 3 and 5 report the average values of the global accuracy (acc) measured for each recognition schemes. The former table reports the results achieved using SVMs as base classifier, whereas the latter reports the results attained when we apply AdaBoost.

Tables 4 and 6 report the average values of the geometric mean of accuracies (g) measured for each scheme: Table 4 refers to g values measured using SVMs, while Table 6 shows the results achieved using AdaBoost.

In order to facilitate the analysis of these results, we perform an exhaustive comparison between these data. Table 7 shows the results of the comparison among the performance of the recognition systems measured in terms of acc, where a win–tie–loss scheme has been adopted. Upper and lower triangles of the table show the results of the comparison when SVMs and AdaBoost are used as base classifiers, respectively. Furthermore, each tabular of Table 7 reports in round parentheses the number of comparisons where the performance is statistically different according to t-test with a significance level of 0.05. Table 8 shows the results of an analogous comparison carried out on performance measured in terms of g.

Table 3
Average values of the global accuracy (acc) when SVMs are used as base classifier.

Classifier	Dataset									
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast
IC	91.7	95.7	82.2	76.6	75.5	96.3	79.5	90.3	97.1	94.5
MES-DCS	83.7	90.3	75.4	74.6	74.5	96.1	70.1	88.5	94.7	89.4
SMOTE	88.7	94.7	83.4	74.4	76.4	96.2	74.7	90.3	96.3	93.7
OSS	87.3	93.5	81.5	72.6	74.0	96.2	72.5	89.7	96.5	93.4
BaCa	88.9	94.3	82.2	74.5	75.2	95.8	76.9	90.3	96.7	93.4
RbB:IC+MES-DCS	93.0	96.0	80.9	77.5	77.1	97.0	79.9	90.1	96.0	92.7
RbB:IC+SMOTE	92.6	96.1	84.5	77.0	77.9	96.8	79.8	90.9	97.2	95.0
RbB:IC+OSS	92.2	96.1	83.1	75.5	77.1	97.2	76.9	90.9	97.1	94.8
RbB:IC+BaCa	92.0	96.1	84.0	77.0	76.7	97.0	80.0	90.8	97.2	95.0

Table 4
Average values of the geometric mean of accuracies (g) when SVMs are used as base classifier.

Classifier	Dataset									
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast
IC	65.7	88.5	67.6	70.0	64.4	95.9	61.9	11.9	77.4	81.4
MES-DCS	87.2	89.1	72.1	73.8	74.9	96.4	70.4	65.7	92.2	89.3
SMOTE	79.3	88.9	73.6	73.4	74.1	96.8	68.3	18.1	81.9	85.5
OSS	82.6	89.1	71.8	73.3	75.1	97.2	69.5	22.5	82.6	86.1
BaCa	66.2	90.0	63.6	69.9	64.9	96.2	58.7	16.0	79.9	83.1
RbB:IC+MES-DCS	88.8	90.2	74.9	74.1	75.8	96.9	72.6	69.4	93.4	90.4
RbB:IC+SMOTE	81.4	91.6	76.4	74.2	74.9	97.0	70.0	22.0	83.1	86.7
RbB:IC+OSS	85.0	92.3	74.0	75.1	76.0	97.5	70.1	24.4	83.8	87.0
RbB:IC+BaCa	69.2	91.1	66.7	71.8	65.5	96.7	60.9	20.4	81.1	85.4

The rest of this section discusses the experimental results from different points of view. In the first paragraph we analyse IC performance in relation to performance of BC, i.e. SMOTE, OSS, MES-DCS and BaCa. In the second paragraph, we compare the performance of traditional methods for class imbalance learning. In the third paragraph we analyse the RbB performance, and in the last paragraph we report a global comparison between tested schemes.

4.3.1. Performance comparison between IC and SMOTE, OSS, MES-DCS and BaCa

This paragraph compares the performance of IC with the performance of classifiers trained with traditional learning methods for skewed data. The results measured in terms of *acc* (Tables 3, 5 and 7) show that IC achieves larger values than MES-DCS, SMOTE, OSS and BaCa in the 95%, 75%, 100% and 70% of experiments, respectively.

Turning our attention to performance measured in terms of *g*, we notice that classifiers trained using traditional learning methods for class skew outperform IC (Tables 4, 6 and 8). This consideration holds for all tests where we apply MES-DCS, SMOTE and OSS. We also observe that using SVMs as base classifier, BaCa compares favourably with IC in 70% of tests, whereas BaCa and IC have similar performance using AdaBoost.

For the sake of presentation, let us consider the following example concerning the performance achieved on the Ecoli dataset using SVMs as a base classifier (first column of Tables 3 and 5). On the one hand, from Table 3 we notice that IC provides larger *acc* values than classifiers trained according to traditional learning methods for skewed data (rows 1–5). For instance, IC and MES-DCS have an *acc* value of 91.7% and 83.7%, respectively. On the other hand, in Table 4 we observe the opposite situation: the *g* values of IC and MES-DCS are 65.7% and 87.2%, respectively.

Table 5
Average values of the global accuracy (*acc*) when AdaBoost is used as base classifier.

Classifier	Dataset									
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast
IC	92.1	96.0	81.2	74.0	76.9	94.5	76.6	90.3	97.9	94.7
MES-DCS	84.9	88.6	73.6	72.6	76.2	95.1	68.2	78.1	95.8	89.4
SMOTE	88.6	95.1	80.7	72.0	77.4	95.1	73.4	88.5	97.7	93.8
OSS	86.9	92.8	80.9	71.3	76.6	94.3	70.8	88.2	97.6	93.9
BaCa	90.5	95.2	81.7	73.6	76.9	95.0	76.2	89.9	98.0	93.1
RbB:IC+MES-DCS	93.1	95.0	80.7	75.4	77.6	95.5	72.1	84.8	96.8	94.4
RbB:IC+SMOTE	92.2	96.5	82.4	74.7	78.0	95.7	77.3	90.6	98.0	95.2
RbB:IC+OSS	93.1	96.2	82.2	74.0	77.3	95.5	77.0	90.4	98.0	94.9
RbB:IC+BaCa	92.2	96.0	83.2	75.8	77.6	95.5	77.4	90.6	98.3	95.2

Table 6
Average values of the geometric mean of accuracies (*g*) when AdaBoost is used as base classifier.

Classifier	Dataset									
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast
IC	59.4	90.0	61.7	68.1	70.6	93.5	40.1	10.3	90.2	88.7
MES-DCS	85.2	87.5	70.6	70.9	76.3	94.8	67.4	82.3	95.6	91.1
SMOTE	70.9	88.6	61.9	70.6	77.9	94.7	55.7	77.8	91.1	90.4
OSS	75.9	88.0	71.3	72.2	77.7	95.0	61.3	75.9	91.9	91.1
BaCa	62.2	83.2	61.1	67.4	71.6	94.2	52.6	15.0	89.3	79.9
RbB:IC+MES-DCS	86.2	92.5	71.7	72.6	77.4	95.1	65.7	83.1	95.9	93.1
RbB:IC+SMOTE	73.1	91.2	65.1	72.0	78.3	95.1	64.8	80.0	92.1	90.4
RbB:IC+OSS	78.3	91.7	75.0	72.7	78.2	95.8	63.4	81.7	92.9	91.4
RbB:IC+BaCa	65.2	90.0	68.0	69.7	72.9	94.7	59.6	21.8	91.7	89.7

Table 7
Exhaustive comparison between the performance of different classifiers expressed in terms of global accuracy (*acc*). Each tabular shows the amount of win–tie–loss of a method in a row comparing with a method in a column. The upper and lower triangles show the results in case of SVMs and AdaBoost, respectively. Round parentheses reports the number of comparisons where the performance is statistically different.

	IC	MES-DCS	SMOTE	OSS	BaCa	RbB:IC+MES-DCS	RbB:IC+SMOTE	RbB:IC+OSS	RbB:IC+BaCa
IC	–	10-0-0 (9)	7-1-2 (5)	10-0-0 (7)	8-2-0 (4)	4-0-6 (4)	0-0-10 (2)	2-1-7 (4)	0-0-10 (1)
MES-DCS	1-0-9 (10)	–	1-0-9 (8)	2-0-8 (7)	2-0-8 (7)	0-0-10 (10)	0-0-10 (9)	0-0-10 (9)	0-0-10 (10)
SMOTE	2-0-8 (6)	8-1-1 (8)	–	8-1-1 (2)	5-1-4 (1)	4-0-6 (7)	0-0-10 (7)	1-0-9 (5)	0-0-10 (7)
OSS	0-0-10 (8)	8-0-2 (7)	2-0-8 (2)	–	1-1-8 (2)	4-0-6 (8)	0-0-10 (8)	0-0-10 (9)	0-0-10 (9)
BaCa	3-1-6 (5)	9-0-1 (7)	7-0-3 (1)	9-0-1 (2)	–	4-0-6 (7)	0-0-10 (6)	0-1-9 (5)	0-0-10 (8)
RbB:IC+MES-DCS	4-0-6 (4)	10-0-0 (10)	5-1-4 (8)	7-0-3 (9)	5-0-5 (8)	–	4-0-6 (5)	3-1-6 (5)	3-1-6 (4)
RbB:IC+SMOTE	10-0-0 (2)	10-0-0 (10)	10-0-0 (8)	10-0-0 (9)	9-1-0 (7)	8-0-2 (5)	–	7-2-1 (2)	4-4-2 (1)
RbB:IC+OSS	9-1-0 (4)	10-0-0 (9)	9-0-1 (5)	10-0-0 (10)	9-1-0 (5)	6-2-2 (6)	1-1-8 (2)	–	4-1-5 (1)
RbB:IC+BaCa	9-1-0 (1)	10-0-0 (10)	10-0-0 (8)	10-0-0 (10)	10-0-0 (9)	7-2-1 (4)	4-3-3 (1)	7-1-2 (1)	–

These results confirm our initial observation motivating the paper: learning methods tailored for skewed data improve the geometric mean of accuracies harming the global accuracy.

4.3.2. Performance comparison between SMOTE, OSS, MES-DCS and BaCa

This paragraph analyses the performance of classifiers trained according to traditional class imbalance learning methods. With reference to Tables 3, 5 and 7, where the performance of MES-DCS, SMOTE, OSS and BaCa is measured in terms of *acc*, we observe that: (i) MES-DCS performs worse than SMOTE, OSS and BaCa in 85%, 80% and 85% of experiments, respectively, using SVMs and AdaBoost as base classifiers, (ii) BaCa outperforms OSS in 85% of tests using SVMs and AdaBoost, (iii) BaCa outperforms SMOTE in 70% of tests using AdaBoost, whereas using SVMs there are five wins and one tie, (iv) SMOTE outperforms OSS in 80% of experiments using SVMs and AdaBoost.

With reference to results expressed in terms of *g* (Tables 4, 6 and 8) we observe that: (i) MES-DCS outperforms SMOTE and BaCa in 80% and 95% of experiments, respectively, using SVMs and AdaBoost, (ii) MES-DCS outperforms OSS in 70% of experiments using SVMs, whereas there are five wins and one tie using AdaBoost, (iii) OSS outperforms SMOTE and BaCa in 70% and 95% of experiments, respectively, using SVMs and AdaBoost, (iv) SMOTE outperforms BaCa in 95% of experiments using SVMs and AdaBoost.

These results show that none of such methods outperforms the others in terms of both *g* and *acc*. Neverthless, we notice that methods providing the largest values of *g* return the lowest values of *acc*. Hence, the more the balance between the accuracies of the two classes, the more the decrease in global accuracy is. Such an observation confirms once again the challenge of learning under class skew.

4.3.3. Results of RbB method

This paragraph compares the performance of RbB schemes with the performance of other methods. The application of RbB method using SVM and AdaBoost provides values of *acc* that are larger than those achieved by IC in 65 out of 80 tests (Table 7). Furthermore, in the same table we notice that in most cases RbB compares favourably against any BCs. As example, consider again the results attained on the Ecoli dataset: in Table 3 we observe that *acc* values of four RbB schemes range between 92.0% and 93.0%, the *acc* value of IC is 91.7% and the *acc* values of BCs range between 83.7% and 88.9%. On the one hand, RbB schemes improves a bit the global recognition performance with respect to IC. On the other hand, the RbB improvement is larger when compared with global performance of traditional learning methods for skewed data.

With reference to results expressed in terms of *g*, in Table 8 we notice that the RbB method outperforms the corresponding BCs in 77 out of 80 tests, using either SVMs or AdaBoost. For instance, on the Ecoli dataset we observe that *g* values of RbB method range between 69.2% and 88.8%, *g* value of IC is 65.7% and *g* values of BCs range between 66.2% and 87.2% (Table 4). Hence, RbB schemes provides *g* values larger than IC as well as BCs.

It is worth noting that such results differ from those presented in previous paragraphs, where achieving more balanced accuracies introduces the side effect of lowering the global recognition accuracy. Now, the RbB method achieves *acc* values larger than corresponding values of IC. At the same time, it attains *g* values larger than those provided by BCs. Indeed, the RbB method mainly aims at achieving balanced performance over the two classes without harming the global accuracy, differently from traditional learning methods for skewed data. This explains why difference of *acc* values between RbB and IC are moderate as well as are the difference of *g* values between RbB and BCs.

Table 8
Exhaustive comparison between the performance of different classifiers expressed in terms of geometric mean of accuracies (*g*). Each tabular shows the amount of win–tie–loss of a method in a row comparing with a method in a column. The upper and lower triangles show the results in case of SVMs and AdaBoost, respectively. Round parentheses reports the number of comparisons where the performance is statistically different.

	IC	MES-DCS	SMOTE	OSS	BaCa	RbB:IC+MES-DCS	RbB:IC+SMOTE	RbB:IC+OSS	RbB:IC+BaCa
IC	–	0-0-10 (7)	0-0-10 (7)	0-0-10 (8)	3-0-7 (1)	0-0-10 (9)	0-0-10 (9)	0-0-10 (8)	2-0-8 (3)
MES-DCS	9-0-1 (8)	–	8-0-2 (5)	7-1-2 (5)	9-0-1 (7)	0-0-10 (2)	5-1-4 (4)	5-0-5 (5)	8-0-2 (6)
SMOTE	9-0-1 (8)	2-0-8 (5)	–	2-0-8 (2)	9-0-1 (4)	0-0-10 (6)	0-0-10 (2)	0-0-10 (4)	8-0-2 (3)
OSS	9-0-1 (9)	5-1-4 (5)	7-0-3 (2)	–	9-0-1 (6)	1-0-9 (5)	4-0-6 (0)	0-0-10 (1)	9-0-1 (3)
BaCa	5-0-5 (1)	0-0-10 (8)	0-0-10 (5)	0-0-10 (7)	–	0-0-10 (8)	0-0-10 (7)	0-0-10 (8)	0-0-10 (1)
RbB:IC+MES-DCS	10-0-0 (10)	9-0-1 (2)	9-0-1 (6)	9-0-1 (5)	10-0-0 (9)	–	6-0-4 (5)	6-0-4 (4)	9-0-1 (8)
RbB:IC+SMOTE	10-0-0 (10)	4-0-6 (4)	9-1-0 (2)	6-0-4 (0)	10-0-0 (8)	1-1-8 (5)	–	1-0-9 (1)	10-0-0 (5)
RbB:IC+OSS	10-0-0 (9)	6-0-4 (5)	10-0-0 (4)	10-0-0 (1)	10-0-0 (9)	4-0-6 (4)	8-0-2 (1)	–	10-0-0 (6)
RbB:IC+BaCa	8-2-0 (3)	1-1-8 (6)	3-2-5 (3)	1-1-8 (3)	9-1-0 (1)	0-1-9 (9)	1-1-8 (6)	0-1-9 (7)	–

Table 9
Ranks of the classification methods when SVMs are used as base classifier. The first and the second values of each tabular are the ranks for performance measured in terms of geometric mean of accuracies (*g*) and global accuracy (*acc*), respectively.

Classifier	Dataset										Sum
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast	
IC	1-5	1-5	3-4	2-6	1-4	1-5	3-6	1-4	1-6	1-6	15-51
MES-DCS	8-1	3-1	5-1	6-4	5-2	3-2	8-1	8-1	8-1	8-1	62-15
SMOTE	4-3	2-4	6-7	5-2	4-5	5-3	4-3	3-4	4-3	4-5	41-39
OSS	6-2	3-2	4-3	4-1	7-1	8-3	5-2	6-3	5-4	5-3	53-24
BaCa	2-4	5-3	1-4	1-3	2-3	2-1	1-4	2-4	2-5	2-3	20-34
RbB:IC+MES-DCS	9-9	6-6	8-2	7-9	8-7	6-7	9-8	9-2	9-2	9-2	80-54
RbB:IC+SMOTE	5-8	8-7	9-9	8-7	5-9	7-6	6-7	5-8	6-8	6-8	65-77
RbB:IC+OSS	7-7	9-7	7-6	9-5	9-7	9-9	7-4	7-8	7-6	7-7	78-66
RbB:IC+BaCa	3-6	7-7	2-8	3-7	3-6	4-7	2-9	4-7	3-8	3-8	34-73

Table 10
Ranks of the classification methods when AdaBoost is used as base classifier. The first and the second values of each tabular are the ranks for performance measured in terms of geometric mean of accuracies (g) and global accuracy (acc), respectively.

Classifier	Dataset										Sum
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast	
IC	1-5	5-6	2-5	2-5	1-3	1-2	1-6	1-6	2-5	2-6	18-49
MES-DCS	8-1	2-1	6-1	5-3	4-1	5-4	9-1	8-1	8-1	6-1	61-15
SMOTE	4-3	4-4	3-2	4-2	7-6	3-4	3-4	5-4	3-4	4-3	40-36
OSS	6-2	3-2	7-4	7-1	6-2	6-1	5-2	4-3	5-3	6-4	55-24
BaCa	2-4	1-5	1-6	1-4	2-3	2-3	2-5	2-5	1-6	1-2	15-43
RbB:IC+MES-DCS	9-8	9-3	8-2	8-8	5-7	7-6	8-3	9-2	9-2	9-5	81-46
RbB:IC+SMOTE	5-6	7-9	4-8	6-7	9-9	7-9	7-8	6-8	6-6	4-8	61-78
RbB:IC+OSS	7-8	8-8	9-7	9-5	8-5	9-6	6-7	7-7	7-6	8-7	78-66
RbB:IC+BaCa	3-6	5-6	5-9	3-9	3-7	3-6	4-9	3-8	4-9	3-8	36-77

Table 11
Rank of the four applications of the RbB method, measured taking into account both acc and g. The two values in each tabular are the ranks achieved using SVMs and AdaBoost as base classifier, respectively.

Classifier	Dataset										Sum
	Ecoli	Glass	Hepatitis	Pima	Phoneme	Breast Cancer Wisconsin	Vehicle	SatImage	PageBlocks	Yeast	
RbB:IC+MES-DCS	3-2	4-4	1-4	4-4	4-4	4-2	2-3	3-2	4-2	4-4	33-31
RbB:IC+SMOTE	2-3	2-2	3-2	2-2	2-2	3-4	4-1	2-4	2-3	3-2	25-25
RbB:IC+OSS	4-4	3-3	4-3	3-3	3-3	2-3	3-4	4-3	3-4	2-3	31-33
RbB:IC+BaCa	1-1	1-1	2-1	1-1	1-1	1-1	1-2	1-1	1-1	1-1	11-11

For example, consider again the results on the Ecoli dataset, focusing on performance of both RbB:IC+MES-DCS and MES-DCS as well as IC (Tables 3 and 4). We observe that: (i) RbB:IC+MES-DCS in comparison with IC improves acc and g values of 1.3% and 23.1%, respectively, and (ii) RbB:IC+MES-DCS in comparison with MES-DCS improves acc and g values of 9.3% and 1.6%, respectively.

We deem that RbB method exhibits the favourable effect of increase both acc and g since it can choose between the predictions of IC and BCs.

4.3.4. Global comparison

This paragraph presents a global comparison between the performance of the tested methods. To this aim, we calculate the relative performance of each method with respect to the others. For each dataset, the nine rows with the values of acc are sorted individually, and each classification method is assigned a rank with respect to its place among the others. The largest rank is nine (assigned to the best method) and the lowest is one (assigned to the worst method). The ten ranks for each classification method are then summed up to give a measure of the overall dominance among the methods in terms of acc. An analogous procedure has been carried out in case of g. The results are reported in Tables 9 and 10 when SVMs and AdaBoost are used as base classifiers, respectively.

With regard to performance measured using SVMs (Table 9) and expressed in terms of acc, we observe that IC attains a rank larger than those of BCs, whereas all RbB configurations outperform IC. With regard to performance measured in terms of g we notice that, on the one hand, IC provides the worst rank and, on the other hand, RbB:IC+MES-DCS achieves the largest rank. Among BC schemes, we notice that MES-DCS provides the most balanced recognition accuracies of each class.

Similar considerations hold using AdaBoost (Table 10).

Finally, we would like to provide a comparison between the four RbB schemes. While comparisons reported in Tables 7 and 8 independently consider acc and g, we now consider together these two performance metrics. To this aim, we compute for each dataset the L^2 distance between the average performance of each RbB configuration and the ideal point **C** with coordinates (1,1). Similarly to previous comparison, we compute the rank for each dataset and then sum up the results. Data are presented in Table 11, where the two values in each tabular report the rank achieved when SVMs and AdaBoost are used as base classifier, respectively.

We notice that RbB configurations based on IC+MES-DCS and IC+OSS achieve the largest rank values using either SVMs or AdaBoost as base classifier, whereas RbB configurations based on IC+SMOTE and IC+BaCa attain the same rank values with the two base classifiers. Therefore, RbB configurations based on IC+MES-DCS and IC+OSS should be preferred to the two other configurations. The choice between the two former configurations may be driven by the analysis of the problem at hand, e.g. type of data, data distribution, constrains on training time, etc., well as by experimental tests. Finally, further to show which RbB configuration has best performance, these results also reveal that rank values are quite independent of base classifier, confirming the robustness of the method.

5. Conclusions

Many existing learning methods for skewed TS balance the recognition accuracies on each class but harm the global accuracy. To overcome such a drawback we have presented an approach that can choose between the predictions provided by a classifier trained on the original skewed TS and by a classifier trained according to a learning method suited for imbalanced TS. This approach maximizes both the global accuracy and the geometric mean of accuracies since it applies a criterion providing an optimum solution according to Pareto multi-objective optimisation theory.

A series of experiments on ten public datasets with different proportions between the majority and minority classes show that the proposed approach provides more balanced recognition accuracies than classifiers trained according to traditional learning methods for imbalanced TS as well as larger global accuracy than classifiers trained on the original skewed distribution, validating its effectiveness.

Acronyms

acc	Classification accuracy
acc+	True positive rate or Recall
acc−	True negative rate
AUC	Classification accuracy
BaCa	Balance cascade
BC	Any classifier trained according to a traditional learning method addressing the course of imbalanced TS
g	Geometric mean of accuracies
IC	Ay classifier trained on the original skewed distribution that does not apply any learning methods for skewed TS
MES	Multi-experts system
MV	Majority voting rule
OSS	One-sided selection
P	Minority training set
N	Majority training set
NN	Nearest Neighbour classifier
RbB	Reliability-based Balancing
RbB:IC+SMOTE	RbB method choosing the final output between the predictions of IC and SMOTE
RbB:IC+OSS	RbB method choosing the final output between the predictions of IC and OSS
RbB:IC+MES-DCS	RbB method choosing the final output between the predictions of IC and MES-DCS
RbB:IC+BaCa	RbB method choosing the final output between the predictions of IC and BaCa
SMOTE	Synthetic minority oversampling technique
SVM	Support Vector Machine
TS	Training set

Acknowledgements

The author would like to thank Giulio Iannello for his continuous support and critical comments. He is grateful to Marco Papi and Francesco Tortorella for their suggestions on multi-objective optimisation.

References

[1] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 20–29.
[2] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (3) (2002) 321–357.
[3] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 40–49.
[4] G.M. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, Journal of Artificial Intelligence Research 19 (2003) 315–354.
[5] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Machine Learning-International Workshop Then Conference, Morgan Kaufmann Publishers Inc, 1997, pp. 179–186.
[6] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: improving classification performance when training data is imbalanced, in: 2009 Second International Workshop on Computer Science and Engineering, IEEE, 2009, pp. 13–17.
[7] G.E. Batista, A.C. Carvalho, M.C. Monard, Applying one-sided selection to unbalanced datasets, in: Lecture Notes in Computer Science, 2000, pp. 315–325.
[8] R. Barandela, J.S. Sanchez, V. Garca, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (3) (2003) 849–851.
[9] T. Eavis, N. Japkowicz, A recognition-based alternative to discrimination-based multi-layer perceptrons, in: AI'00: Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence, 2000, pp. 280–292.
[10] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of support vector machines, in: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence IJCAI99, 1999, pp. 55–60.
[11] K. Ezawa, M. Singh, S. Norton, Learning goal oriented Bayesian networks for telecommunications risk management, in: Machine Learning-International Workshop then Conference, 1996, pp. 139–147.
[12] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (2) (2009) 539–550.
[13] R. Barandela, R.M. Valdovinos, J.S. Sánchez, New applications of ensembles of classifiers, Pattern Analysis and Applications 6 (3) (2003) 245–256.
[14] M. Molinara, M.T. Ricamato, F. Tortorella, Facing imbalanced classes through aggregation of classifiers, in: ICIAP'07: Proceedings of the 14th International Conference on Image Analysis and Processing, 2007, pp. 43–48.
[15] S. Kotsiantis, P. Pintelas, Mixture of expert agents for handling imbalanced data sets, Annals of Mathematics Computing and Teleinformatics 1 (1) (2003) 46–55.
[16] P. Soda, An experimental comparison of MES aggregation rules in case of imbalanced datasets, in: 22nd IEEE International Symposium on Computer-Based Medical Systems, 2009. CBMS 2009, IEEE Computer Society, Los Alamitos, CA, USA, 2009, pp. 1–6.
[17] T. Fawcett, ROC graphs: notes and practical considerations for researchers, Machine Learning 31 (2004) 1–38.
[18] I. Tomek, Two modifications of CNN, IEEE Transactions on Systems, Man and Cybernetics 6 (6) (1976) 769–772.
[19] L.P. Cordella, P. Foggia, C. Sansone, F. Tortorella, M. Vento, Reliability parameters to improve combination strategies in multi-expert systems, Pattern Analysis and Applications 2 (3) (1999) 205–214.
[20] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.
[21] K. Woods, W.P. Kegelmeyer, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 405–410.
[22] P. Foggia, G. Percannella, C. Sansone, M. Vento, On rejecting unreliably classified patterns, Multiple Classifier Systems, vol. 4472, Springer-Verlag, Heidelberg, 2007, pp. 282–291.
[23] F. John, Extremum problems with inequalities as subsidiary conditions, in: J. Moser (Ed.), Fritz John, Collected Papers, vol. 2, Birkhäauser, Boston, 1985, pp. 543–560.
[24] H.W. Kuhn, A.W. Tucker, Nonlinear programming, ACM SIGMAP Bulletin (1982) 6–18.
[25] A. Asuncion, D.J. Newman, UCI machine learning repository, URL 〈http://www.ics.uci.edu/mlearn/MLRepository.html〉, 2007.
[26] G. Fumera, F. Roli, Support vector machines with embedded reject option, in: Lecture Notes in Computer Science, 2002, pp. 68–82.
[27] R. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336.

Paolo Soda was born in Rome, Italy, in 1981. He graduated with honours in Biomedical Engineering at Università Campus Bio-Medico, Rome, in 2004 and received a Ph.D. in Biomedical Engineering (Computer Science area) in 2008 from the same University. Currently he is Assistant Professor at the same University, developing a research program entitled "processing of data, signals and images in biomedical applications". His research interests are in bio-medical image analysis, pattern recognition and data mining, serving also as Special Track Chair of IEEE Symposium on these topics. He has been working extensively on computer-aided diagnosis tools for microscope applications.