Soft Comput (2015) 19:3357–3368
DOI 10.1007/s00500-014-1407-5

FOCUS

Boosted SVM with active learning strategy for imbalanced data

Maciej Zięba · Jakub M. Tomczak

Published online: 7 August 2014

Abstract In this work, we introduce a novel training method for constructing *boosted Support Vector Machines* (SVMs) directly from imbalanced data. The proposed solution incorporates the mechanisms of active learning strategy to eliminate redundant instances and more properly estimate misclassification costs for each of the base SVMs in the committee. To evaluate our approach, we make comprehensive experimental studies on the set of 44 benchmark datasets with various types of imbalance ratio. In addition, we present application of our method to the real-life decision problem related to the short-term loans repayment prediction.

Keywords Imbalanced data · Boosted SVM · Active learning

1 Introduction

The imbalanced data phenomenon is known to be one of the fundamental problems in data analysis and prediction. In general, every dataset which exhibits disproportions in the class distribution can be treated as imbalanced. In the context of binary classification problem, we call the majority class to be the one which dominates the training examples and the less prominent one is the minority class. For further consistency we refer to the minority class to as *positive* and to the major5ity class to as *negative*. In practice, the imbalanced data issue is observed when disproportion between classes has the impact on the constructed learner that is biased toward majority class. For extremely uneven data distribution, typical learning methods may construct classifiers that have tendency to classify all examples as members of the majority class. The problem of imbalanced data is widely observed in various domains such as medical diagnosis, fraud detection, consumer credit risk assessment and many others (Japkowicz and Stephen 2002).

To solve the manner of the disproportions between classes, various techniques can be applied (He and Garcia 2009). The issue can be solved externally, by applying preprocessing on data before the training procedure. Two techniques are commonly observed in this group: generating artificial examples from minority class (*oversampling*) and eliminating observations from majority class (*undersampling*). The most commonly used oversampling technique is SMOTE (Synthetic Minority Over-sampling TEchnique) (Chawla et al. 2002), which generates additional examples situated on the path connecting two neighbors from minority class. Another method in this group is Borderline-SMOTE which is an extension of SMOTE that incorporates in the sampling procedure only the minority data points with a high percentage of the nearest neighbors from majority class (Hui et al. 2005). The policy of undersampling methods is to remove those instances from majority class that are redundant in training procedure and bias the classifier. It is usually performed by random elimination, using K-NN algorithm (Mani and Zhang 2003) or evolutionary algorithms (García et al. 2009).

The problem of imbalanced data can be solved directly at the training stage by incorporating proper mechanisms for well-known training methods. In this group, it is possible to distinguish ensemble classifiers such as *SMOTEBoost* (Chawla et al. 2003), *SMOTEBagging* (Wang and Yao 2009),

Communicated by E. Lughofer.

M. Zięba (✉) · J. M. Tomczak
Faculty of Computer Science and Management,
Wroclaw University of Technology, Wybrzeze Wyspianskiego 27,
50-370, Wroclaw, Poland
e-mail: maciej.zieba@pwr.edu.pl

J. M. Tomczak
e-mail: jakub.tomczak@pwr.edu.pl

RAMOBoost (Chen et al. 2010), which make use of oversampling to diversify the base learners, and models such as *UnderBagging* (Tao et al. 2006), *Roughly Balanced Bagging* (Hido et al. 2009), *RUSBoost* (Seiffert et al. 2010) which apply undersampling before creating each of the component classifiers. In addition to the mentioned learning methods for imbalanced data, other internal techniques are successively applied to construct balanced classifiers, e.g., *active learning strategies* (Ertekin et al. 2007), *granular computing* (Tang et al. 2007), or *one-sided classification* (Manevitz and Yousef 2002).

Beside the internal and external approaches, we can distinguish cost-sensitive techniques that put higher misclassification costs to the minority examples. This group of methods perform inference by assigning weights to each of the examples in the training data as well as adjusting training procedure by introducing different misclassification costs. In this group of techniques, we can identify the algorithms for constructing cost-sensitive models such as decision trees (Drummond and Holte 2000), neural networks (Kukar and Kononenko 1998), SVMs (Morik et al. 1999) and ensemble classifiers (Fan et al. 1999; Wang and Japkowicz 2010; Zięba et al. 2014).

Modern solutions utilize *boosted SVM* classifiers as high-quality, cost-sensitive predictors (Wang and Japkowicz 2010; Zięba et al. 2014). Despite the high accuracy of prediction of such models confirmed by numerous experiments, the problem of setting proper values of misclassification costs arises during training. To avoid time-consuming calibration of the parameters for each of the classification problems separately, the ratio between negatives and positives is taken as a basis for penalty cost calculation. In such approach, we assume that the value of global imbalance ratio for entire data is similar to the ratio between negatives and positives situated near the borderline. This statement is not always satisfied because of different distribution of examples for different datasets.

To overcome the stated issue, we propose a novel training method for *boosting SVM* that makes use of active learning strategy to select the most informative examples and more accurately calculate misclassification costs. Each of the base learners of the ensemble is trained on the reduced number of instances, selected to be significant by the previously created component classifier. In this approach, the considered dataset is composed only of the examples situated near the borderline and the penalization terms are calculated basing on local cardinalities of positives and negatives. As a consequence, the consecutive training sets used to construct the base classifiers are more balanced and do not contain redundant and noisy cases.

We identify the borderline examples by introducing the "wide margin" for the base SVM that was created in the previous iteration of constructing the ensemble model. The "wide margin" is the extended "soft margin" obtained in standard training procedure of this component classifier. Therefore, we select all the examples situated in the "wide margin"—the support vectors (beside the "noisy" support vectors located outside) as well as the examples located close to the "soft margin".

We compare the predictive performance of our solution with other reference methods dedicated to solve the imbalanced data problem. The experiment is carried out for 44 benchmark datasets. In addition, we apply our training method to the problem of the short-term loans risk analysis and present how to induce reasonable rules from *boosting SVM*. The short-term loans risk analysis is a typical situation in which data are imbalanced and irregularly distributed.

The paper is organized as follows. In Sect. 2, we describe the novel procedure for constructing *boosted SVM*. Section 3 contains the results of an experiment showing the quality of the proposed approach. In Sect. 3.2, we present the case study related to the problem of predicting short-term loans risk assessment. The paper is summarized with conclusions in Sect. 4.

2 Methods

2.1 SVM for imbalanced data

The standard SVM[1] is trained by finding the optimal hyperplane H of the following form (Vapnik 1998):

$$H : \mathbf{a}^{\top}\phi(\mathbf{x}) + b = 0, \tag{1}$$

where $\mathbf{x}$ is the vector of the input values, $\mathbf{a}$ is the vector of the parameters, b is the bias term and $\phi(\cdot)$ is fixed feature-space transformation.

Assume that the training set $\mathbb{S}_N = \{\mathbf{x}_n, y_n\}_{n=1}^N$ is given, where $y_n \in \{-1, 1\}$. The problem of training standard SVM can be formulated as the following optimization task:

$$\begin{aligned} \min_{\mathbf{a},b} \; & Q(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{a} + C\sum_{n=1}^{N}\xi_n \\ \text{s.t.} \; & y_n(\mathbf{a}^T\phi(\mathbf{x}_n) + b) \geq 1 - \xi_n \\ & \text{for all } n = 1, \ldots, N \end{aligned} \tag{2}$$

where ξ_n are slack variables, such that $\xi_n \geq 0$ for $n = 1, \ldots, N$, and C is the parameter that controls the trade-off between the slack variable penalty and the margin, $C > 0$.

The application of the following criterion to imbalanced training data may result in constructing highly biased classifier toward majority class. Therefore, in Zięba et al. (2014),

[1] We refer SVM in case of balanced data to as *standard SVM*.

a modified criterion was proposed:

$$\begin{aligned}\min_{\mathbf{a},b}\ & Q(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\top}\mathbf{a} \\ & + \frac{CN^2}{2}\Big(\frac{1}{N_+}\sum_{n\in\mathbb{N}_+} w_n\xi_n + \frac{1}{N_-}\sum_{n\in\mathbb{N}_-} w_n\xi_n\Big) \\ & \text{s.t. } y_n(\mathbf{a}^T\phi(\mathbf{x}_n)+b) \geq 1-\xi_n \\ & \quad \text{for all } n = 1,\ldots,N\end{aligned} \tag{3}$$

where $\mathbb{N}_+ = \{n \in \{1,\ldots,N\} : y_n = 1\}$ is the set of all positive examples, $\mathbb{N}_- = \{n \in \{1,\ldots,N\} : y_n = -1\}$ is the set of all negative examples, and N_+, N_- are corresponding cardinalities of the sets. Weights w_n are the penalty parameters that fulfill the following conditions:

$$\sum_{n=1}^{N} w_n = 1 \quad \text{and} \quad w_n > 0 \text{ for all } n = 1,\ldots,N. \tag{4}$$

Notice that weights w_n satisfy the properties of a probability distribution. If we assume equal distribution for w_n, i.e., $w_n = \frac{1}{N}$, the accumulated penalization term for the selected positive example is equal $C\frac{N}{2N_+}$ and for chosen negative case is $C\frac{N}{2N_-}$. The cardinality of instances from the positive class is significantly lower than for negative one because of the considered imbalanced data phenomenon. Therefore, the negative receives a higher penalty for improper location relative to the separating hyperplane than the improperly situated positive, and thus the trained classifier is unbiased toward majority class. The classifier trained in this fashion is known as a popular cost-sensitive SVM variation for imbalanced data (further named C-SVM). Parameter C has the same interpretation as for standard SVM. The within-class imbalance issue is handled by applying different values of w_n. The process of determining the values of weights will be discussed further in this work.

The stated optimization problem (3) can be formulated in its dual form:

$$\begin{aligned}\min_{\boldsymbol{\lambda}}\ & Q_D(\boldsymbol{\lambda}) = \sum_{n=1}^{N}\lambda_n - \frac{1}{2}\sum_{i,j=1}^{N}\lambda_i\lambda_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) \\ & \text{s.t. } 0 \leq \lambda_n \leq C\frac{N^2}{2N_+}w_n \\ & \quad \sum_{n=1}^{N}\lambda_n y_n = 0 \\ & \quad \text{for all } n = 1,\ldots,N,\end{aligned} \tag{5}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. In addition, we have applied the kernel trick, i.e., we have replaced the inner product with the kernel function, $\phi(\mathbf{x}_i)^{\top}\phi(\mathbf{x}_j) = k(\mathbf{x}_i,\mathbf{x}_j)$.

The procedure of classification is made by applying the model:

$$h(\mathbf{x}_i) = \text{sign}\Big(\sum_{n\in\mathcal{SV}} y_n\lambda_n k(\mathbf{x}_n,\mathbf{x}_i) + b\Big), \tag{6}$$

where $\mathcal{SV}$ denotes the set of indices of the support vectors,[2] sign(a) is the signum function that returns -1 for $a < 0$, and 1 – otherwise, and the bias parameter is determined as follows:

$$b = \frac{1}{N_{\mathcal{SV}}}\sum_{n\in\mathcal{SV}}\Big(y_n - \sum_{m\in\mathcal{SV}}\lambda_m y_m k(\mathbf{x}_n,\mathbf{x}_m)\Big), \tag{7}$$

where $N_{\mathcal{SV}}$ is the total number of the support vectors.

2.2 The issue of determining penalty costs

The main drawback of the presented method is a need of incorporating the cardinalities N_+, N_- in the penalty term of the criterion (3) which is minimized to construct the SVM using imbalanced data. The presented method makes the explicit assumption about the ratio between N_- and N_+, so that its growth has very significant impact on the bias degree of the constructed learner. In other words, the higher disproportions between classes are observed, the stronger the tendency of the trained classifier to classify positive examples as negatives. Such an assumption is not always correct in real-life problems. We discuss this issue on a toy example.

Let us consider two artificially generated imbalanced datasets presented in Fig. 1. Both of them have the same cardinality of positive and negative examples, but the distribution over them is noticeable different. In the first case (Fig. 1a), most of the majority examples are situated in the close neighborhood of the separating hyperplane. In the second case (Fig. 1b), the examples from dominating class are clustered far from the classification borderline. Training SVM using the first dataset results in very good predictive accuracy, i.e., the points from two classes seem to be almost perfectly separated. By the introduction of different penalization terms in the training criterion,[3] the hyperplane is stabilized in such a form, that two of the minority cases supporting positive class in the region at issue are located on the proper side of the separator and the remaining positive instance was dedicated at the expense of correct classification of a few negatives (see Fig. 1a).

On the other hand, if the same type of classifier is trained on the second dataset with the same misclassification costs, the quality of the trained separator is highly debatable. Contrary to the previous case, the considered dataset is balanced in the disputed region, but due to the significantly higher penalty weights for positive examples the classifier is biased toward minority class (see Fig. 1b). As a consequence, the trained classifier "sacrificed" 6 majority points at the expense of 2 correctly classified positives, because the ratio between

[2] Support vectors are the examples, for which the corresponding Lagrange multiplier is >0.

[3] The penalization terms are proportional to $\frac{1}{N_+}$ (minority examples) and $\frac{1}{N_-}$ (majority cases).

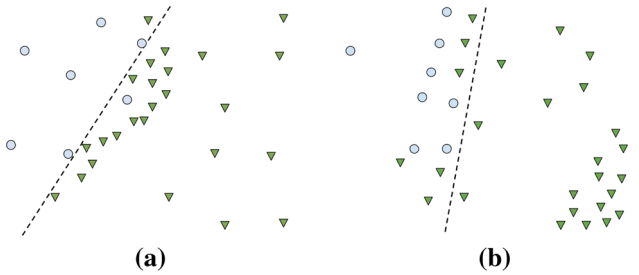

Fig. 1 Optimal hyperplanes for cost-sensitive SVM trained on imbalanced datasets, when: **a** the most of the majority examples are situated near the hyperplane, **b** the most of the majority examples are situated far from the hyperplane

N_+ and N_- calculated on the entire dataset is equal almost 4. If we associate this result with the ratio of misclassification costs, such debatable behavior of the trained classifier is justified.

From this simple example, we can notice that the ratio between N_+ and N_- does not always inform about how real data are imbalanced in the classification problem. Therefore, it would be essential to propose a technique for selecting informative examples for the training process.

The stated issue can be solved by applying so-called *active learning* techniques (Settles 2010). These kinds of methods are widely used for given unlabelled data and when the costs of discovering the class labels are too high to receive complete training set. Therefore, there is a need to find the most informative candidates to *inquire* about objects' class values. Authors of Ertekin et al. (2007) present the application of the *active learning* strategy to deal with the imbalanced data issue. In the first step, they generate small pool of the balanced data and train the classifier. Next, they select the most informative example to be incorporated to the training data by applying a novel searching approach. Finally, they correct the location of separating hyperplane by retraining the classifier on the updated data. The entire procedure is repeated until the set of the most informative examples is selected. This approach makes an explicit assumption that the examples located near the borderline tend to be much more balanced than the entire data. Referring to the example presented in Fig. 1a, such a statement is not always satisfied.

2.3 Our Approach

In this work, we propose a *boosted SVM* with a novel active learning strategy that solves the issue of imbalanced data by proper informative examples selection and misclassification costs estimation. Each of the base SVMs of an ensemble is trained by solving (5) with actual values of weight $w_n^{(k)}$ on the reduced dataset that contains the most informative examples situated near the separating hyperplane. The process of active selection is performed using previously constructed base classifier as an oracle-based selector that makes use of extended margin to locate the most important observations.

Algorithm 1: Boosted SVM with active learning for imbalanced data

Input : $\mathbb{S}_N$: training set, $\mathbb{S}_{val}$: validation set, $\mathcal{Y} = \{-1, 1\}$: set of class labels, K: number of iterations, l: rescaled distance between extended and separating margins, γ: rescaling parameter

Output: Boosted SVM: $H(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{k=1}^{K_{final}} c_k I(h_k(\mathbf{x}) = y)$

Initialize: $w_n^{(1)} \longleftarrow 1/N$ for $n \in \{1, \ldots, N\}$;
$G \longleftarrow 0$;
$K_{final} \longleftarrow 1$;
for $k = 1 \rightarrow K$ **do**
  `//active learning strategy for selecting examples`
  if $k > 1$ **then**
    Determine $\mathbb{S}_{N_k}$ given by (8);
  else
    Determine $\mathbb{S}_{N_k}$ by applying one-sided selection;
  end
  `//procedure of learning boosting SVM`
  Train SVM h_k on $\mathbb{S}_{N_k}$ by solving (5) with actual values of $w_n^{(k)}$, $N_+^{(k)}$ and $N_-^{(k)}$.;
  Calculate e_k given by (9) on $\mathbb{S}_{N_k}$ achieved by h_k;
  if $e_k < 0.5$ **then**
    $c_k \longleftarrow \ln \frac{1-e_k}{e_k}$;
    Calculate *GMean* value g_k on $\mathbb{S}_{val}$ achieved by $H_k(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{l=1}^{k} c_l I(h_l(\mathbf{x}) = y)$;
    if $g_k > G$ **then**
      $G \longleftarrow g_k$;
      $K_{final} \longleftarrow k$;
    end
    Update: $w_n^{(k+1)} \longleftarrow w_n^{(k)} \exp\left(c_k I(h_k(\mathbf{x}_n) \neq y_n)\right)$;
    Normalize: $w_n^{(k+1)} \longleftarrow \frac{w_n^{(k+1)}}{\sum_{n=1}^{N} w_n^{(k+1)}}$;
  else
    $c_k \longleftarrow 0$;
    Update: $w_n^{(k+1)} \longleftarrow 1/N$;
    $C \longleftarrow (1-\gamma)C$;
  end
end

The entire procedure is described in Algorithm 1. In the initial step, the starting weights $w_n^{(k)}$ are equal $\frac{1}{N}$. Next, if $k > 1$, the dataset $\mathbb{S}_{N_k}$ used to construct the k-th base learner is determined in the following way:

$$\mathbb{S}_{N_k} = \{(\mathbf{x}_n, y_n) \in \mathbb{S}_N : y_n y_{k-1}(\mathbf{x}_n) \leq 1 + l\}, \tag{8}$$

where $y_{k-1}(\mathbf{x}_n)$ represents the output of $(k-1)$-th base SVM and l ($l \geq 0$) is the parameter that stays behind the rescaled distance between extended and separating margins. In the

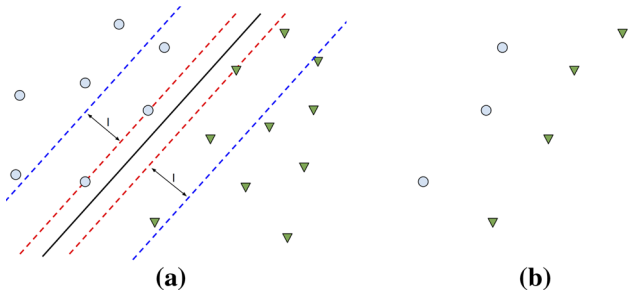

Fig. 2 The application of the wide margin for active selection

rescaled data space, the width of the margin is equal 2, and the separating hyperplane is located exactly in the middle, diving it into equidistant space regions. Therefore, the parameter l represents the percentage extension of the margin extracted in the process of training SVM. As a conclusion, the higher value of l is observed, the more examples are selected. Figure 2a presents the exemplary wide margin for an exemplary dataset and Fig. 2b represents the data after the active learning procedure, i.e., the examples selection.

The issue of determining the dataset for the first base learner ($k = 1$) can be solved by applying one of the typical undersampling techniques. In this work, we recommend to use method called *one-sided selection* (Kubat and Matwin 1997). The idea of this approach is as follows. First, a negative example is randomly selected from the training set. Next, each of the remaining negatives in the dataset is examined if it is located closer to the selected sample than to any of the positives. If the considered example is located closer to the one of the minority cases, it remains in the training set. Otherwise it is removed from the dataset. This solution is successively applied to identify and eliminate the majority instances located far from the borderline and can be also repeated to eliminate such located examples from the minority class. An exemplary application of the one-sided selection is presented in Fig. 3.

Next, after the active learning strategy in the Algorithm 1, the set of base learners h_k represented by SVMs is iteratively constructed in the loop. Each of the classifiers is trained on $\mathbb{S}_{N_k}$ by solving the optimization problem (5) with actual weight values ($w_n = w_n^{(k)}$ for $n \in \{1, \ldots, N\}$) and with the

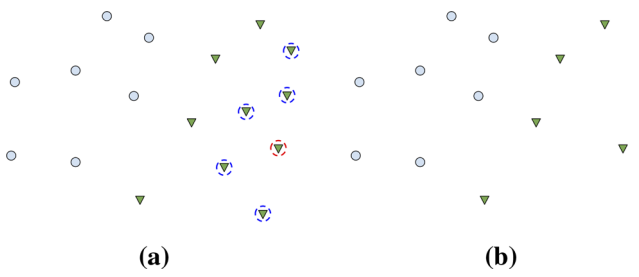

Fig. 3 The application of one-sided selection

actual cardinalities of positive examples ($N_+ = N_+^{(k)}$, $N_- = N_-^{(k)}$). Therefore, the imbalanced data issue is handled each time the base learner is trained by solving the problem with updated penalization terms calculated basing on cardinalities of positives and negatives situated close to the borderline.

In the following step, the value of error function e_k is calculated using the formula:

$$e_k = \frac{E_{Imb}}{\frac{1}{N_+}\sum_{n\in\mathcal{N}_+} w_n^{(k)} + \frac{1}{N_-}\sum_{n\in\mathcal{N}_-} w_n^{(k)}} \tag{9}$$

where E_{Imb} is equal:

$$E_{Imb} = \frac{1}{N_+}\sum_{n\in\mathcal{N}_+} w_n^{(k)} I\left(h_k(\mathbf{x}_n) \neq y_n\right) + \frac{1}{N_-}\sum_{n\in\mathcal{N}_-} w_n^{(k)} I\left(h_k(\mathbf{x}_n) \neq y_n\right), \tag{10}$$

where $I(\cdot)$ denotes the indicator function. The application of such error function has theoretical justification (see Zięba et al. 2014 for details).

If the error value is lower than 0.5 it is further used to compute the value of parameter c_k, which represents the significance of the classifier h_k in the committee. The weights are updated using typical *AdaBoost* procedure to increase the impact of misclassified examples in the training set (Freund et al. 1996). Otherwise, the value of c_k is set to 0 to eliminate the impact of poor learner in the committee, and the weights are reset to the initial values. In addition, the value of parameter C is decreased by multiplying it by $(1 - \gamma)$, where $\gamma \in [0, 1]$ is an arbitrarily chosen rescaling parameter. As a consequence, the base learners created in the further steps will be more general because of the weaker penalization for incorrectly classified examples.

The output ensemble is composed of the set of base learners with the highest *geometric mean* (*GMean*) value. *GMean* is the typical evaluation criterion for imbalanced data and is described by the equation (Kubat and Matwin 1997):

$$GMean = \sqrt{TP_{rate} \cdot TN_{rate}}, \tag{11}$$

where TN_{rate} is *specificity* rate (*true negative rate*) defined by:

$$TN_{rate} = \frac{TN}{TN + FP}, \tag{12}$$

and TP_{rate} is *sensitivity* rate (*true positive rate*) described by the following equation:

$$TP_{rate} = \frac{TP}{TP + FN} \tag{13}$$

The meaning of true positive (TP), false negative (FN), false positive (FP) and true negative (TN) is explained by confusion matrix (see Table 1), which illustrates prediction tendencies of considered classifier.

Table 1 A confusion matrix

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

The process of selecting the proper values of parameters is an important issue for training the presented classifier.The K value should be large enough, because we select the subset of base learners with the highest GMean gained on the validation set. As a consequence, the problem of overfitting is handled. For the other parameters, we suggest to use validation set to find their optimal values. Moreover, for the sparsely populated data, we rather recommend to use the linear kernel, than more sophisticated functions, e.g., Radial Basis Functions. By selecting the base learner with lower number of degrees of freedom, we are able to achieve proper model generalization and we avoid overfitting.

3 Experiments

We carry out two experiments:

- **Experiment 1**: the presented method is evaluated on 44 benchmark datasets with varying value of imbalance ratio.
- **Experiment 2**: the presented approach is applied to the real-life decision problem related to the short-term loans repayment prediction.

3.1 Experiment 1: benchmark datasets

3.1.1 Description

In this part of the paper, we examine the quality of the presented approach in comparison to other methods dedicated for imbalanced data on the set of 44 benchmark datasets available in *KEEL* tool and on website.[4] Multiclass datasets are modified to obtain two-class imbalanced data by merging some of possible class values (Galar et al. 2012). Detailed description of the datasets is presented in Table 2, where **#Inst**. denotes total number of instances, **#Attr**. is the number of attributes in dataset, **%P** and **%N** represent percentage of positive and negative examples, respectively, and $\mathbf{Imb_{rate}}$ is the imbalance ratio, i.e., the ratio between negative and positive examples.

As it was noticed in Sect. 2.2, the imbalanced ratio calculated by dividing the cardinalities of the examples from the different classes does not always correspond to the real bias level of the constructed learner trained using typical procedure. To evaluate the real degree of misclassification ten-

Table 2 Characteristic of datasets used in experiment (Galar et al. 2012)

ID	Dataset	#In	#At	%P	%N	Imb_{rate}
1	Glass1	214	9	35.51	64.49	1.82
2	Ecoli0vs1	220	7	35.00	65.00	1.86
3	Wisconsin	683	9	35.00	65.00	1.86
4	Pima	768	8	34.84	66.16	1.90
5	Iris0	150	4	33.33	66.67	2.00
6	Glass0	214	9	32.71	67.29	2.06
7	Yeast1	1,484	8	28.91	71.09	2.46
8	Vehicle1	846	18	28.37	71.63	2.52
9	Vehicle2	846	18	28.37	71.63	2.52
10	Vehicle3	846	18	28.37	71.63	2.52
11	Haberman	306	3	27.42	73.58	2.68
12	Glass0123vs456	214	9	23.83	76.17	3.19
13	Vehicle0	846	18	23.64	76.36	3.23
14	Ecoli1	336	7	22.92	77.08	3.36
15	New-thyroid2	215	5	16.89	83.11	4.92
16	New-thyroid1	215	5	16.28	83.72	5.14
17	Ecoli2	336	7	15.48	84.52	5.46
18	Segment0	2,308	19	14.26	85.74	6.01
19	Glass6	214	9	13.55	86.45	6.38
20	Yeast3	1,484	8	10.98	89.02	8.11
21	Ecoli3	336	7	10.88	89.77	8.77
22	Page-blocks0	5,472	10	10.23	89.77	8.77
23	Yeast2vs4	514	8	9.92	90.08	9.08
24	Yeast05679vs4	528	8	9.66	90.34	9.35
25	Vowel0	988	13	9.01	90.99	10.10
26	Glass016vs2	192	9	8.89	91.11	10.29
27	Glass2	214	9	8.78	91.22	10.39
28	Ecoli4	336	7	6.74	93.26	13.84
29	Yeast1vs7	459	8	6.72	93.28	13.87
30	Shuttle0vs4	1,829	9	6.72	93.28	13.87
31	Glass4	214	9	6.07	93.93	15.47
32	Page-blocks13	472	10	5.93	94.07	15.85
33	Abalone9vs18	731	8	5.65	94.25	16.68
34	Glass016vs5	184	9	4.89	95.11	19.44
35	Shuttle2vs4	129	9	4.65	95.35	20.5
36	Yeast1458vs7	693	8	4.33	96.67	22.10
37	Glass5	214	9	4.20	95.80	22.81
38	Yeast2vs8	482	8	4.15	95.85	23.10
39	Yeast4	1,484	8	3.43	96.57	28.41
40	Yeast1289vs7	947	8	3.17	96.83	30.56
41	Yeast5	1,484	8	2.96	97.04	32.78
42	Ecoli0137vs26	281	7	2.49	97.51	39.15
43	Yeast6	1,484	8	2.49	97.51	39.15
44	Abalone9	4,174	8	0.77	99.23	128.87

[4] http://www.keel.es/dataset.php.

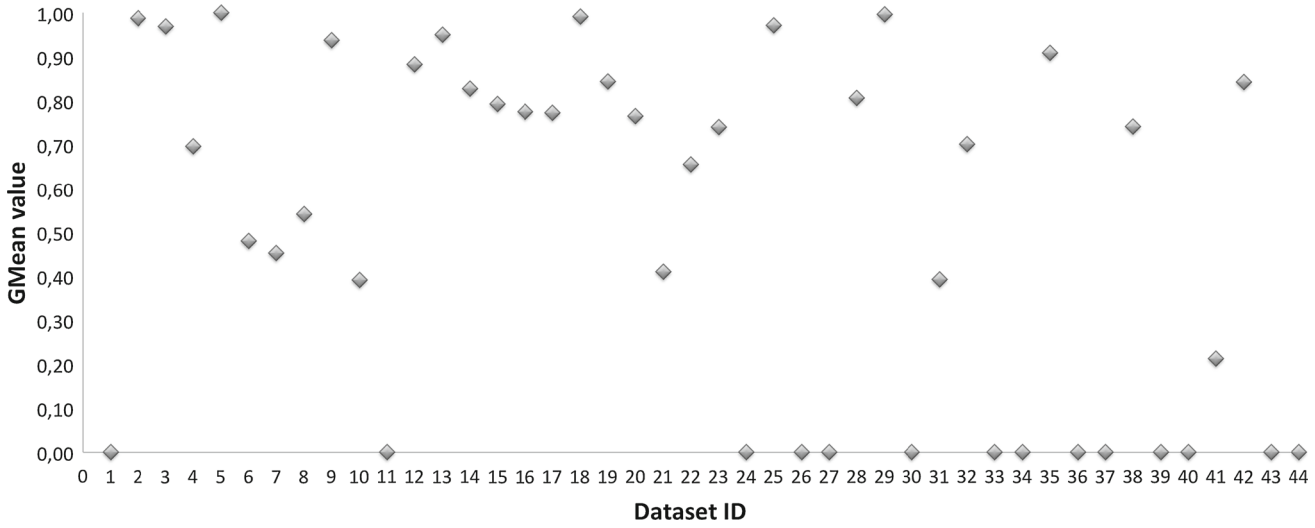

Fig. 4 The *GMean* values for each of the 44 benchmark datasets gained by standard SVM trained with SMO

dency, we examined the quality of standard SVM trained on each of the datasets. We applied fivefold cross validation and used the *GMean* as an evaluation criterion. The plot of the results is presented in Fig. 4. It can be observed that the value of imbalanced ratio is weakly correlated with the $GMean$ value achieved by SVM.[5] For instance, the classifier trained on *Ecoli0137vs26* dataset ($ID = 42$, $\mathbf{Imb_{rate}} = 39.15$) was significantly more balanced ($GMean = 0.84$) than the predictor of the same type trained on *Haberman* data ($ID = 11$, $\mathbf{Imb_{rate}} = 2.68$, $GMean = 0$) despite the fact that the first of them contains only 2.5 % positives and for the second of the considered benchmarks almost 27.5 % minority cases were identified. Therefore, the application of the active learning strategy presented in this work for constructing *boosted SVM* classifier seems to be justified.

3.1.2 Methods

The quality of the *boosting SVM* with active learning strategy (**BSIA**) was compared with other methods suitable for the imbalanced data:

- *SVM* (**SVM**): SVM trained using SMO.
- *SVM + SMOTE* (**SSVM**): SVM trained on data oversampled by *SMOTE*.
- *SMOTEBoostSVM* (**SBSVM**): Boosted SVM which uses SMOTE to generate artificial samples before constructing each of base classifiers.
- *C-SVM* (**CSVM**): Cost-sensitive SVM described in details in Veropoulos et al. (1999).
- *AdaCost* (**AdaC**): Cost-sensitive, ensemble classifier, in which the misclassification cost for minority class is higher than the misclassification cost for majority class (Fan et al. 1999).
- *SMOTEBoost* (**SBO**): modified *AdaBoost* algorithm, in which base classifiers are constructed using *SMOTE* synthetic sampling (Chawla et al. 2003).
- *RUSBoost* (**RUS**): extension of *SMOTEBoost* approach, which uses additional undersampling in each boosting iteration (Seiffert et al. 2010).
- *SMOTEBagging* (**SB**): bagging method, which uses *SMOTE* to oversample dataset before constructing each of base classifiers (Wang and Yao 2009).
- *UnderBagging* (**UB**): bagging method, which randomly undersamples dataset before constructing each of base classifiers (Tao et al. 2006).
- *BoostingSVM-IB* (***BSI***): boosted SVM trained with cost-sensitive approach presented in Zięba et al. (2014).

3.1.3 Methodology

As a testing methodology we used fivefold stratified cross validation with a single repetition and each of the methods was tested on the same folds. The values of the training parameters for the reference methods were set basing on the experimental results described in Galar et al. (2012). For **BSI** and **BSIA**, we identified the most proper values of the training parameters experimentally by testing their quality on validation set. The quality criterion selected for our studies was *GMean* because of its very strict penalization for biased models.

[5] The datasets are sorted ascending by the value of this imbalance ratio (for the higher ID of the dataset, we observe the higher $\mathbf{Imb_{rate}}$ value).

3.1.4 Results and discussion

The results of the comprehensive study are presented in Table 3. The analysis of the performance of the considered classifiers leads us to the conclusion that **BSIA** outperforms other methods by archiving the average *GMean* value equal 0.8845. We observed the slight increase of **BSIA** in comparison to the results obtained by *boosted SVM* trained without applying additional mechanisms of active selection (**BSI**). To evaluate the significance of the results, we applied the Holm–Bonferroni method Holm (1979) that is used to counteract the problem of multiple comparisons. First, the set of pairwise Wilcoxon tests is conducted to calculate the p values for the hypothesis about the equality of medians of the both samples. Next, the calculated p values are sorted ascending and the following inequality is examined:

$$pval_i \leq FWER_i, \tag{14}$$

where $pval_i$ represents i-th p value in the sequence. The factor $FWER_i$ is familywise error rate and for the given significance level α it can be calculated using the equation:

$$FWER_i = \frac{\alpha}{M+1-i}, \tag{15}$$

where M is the number of tested hypothesis. If the inequality (15) is satisfied, then the hypothesis about medians equality is rejected. The results for the pairwise tests between **BSIA** and the reference methods are presented in Table 4. For the set of Wilcoxon test, the p values are lower than the corresponding values $FWER_i$ for a given significance rate α equal 0.05. Therefore, with the probability equal 95 %, we can say that our approach constructs better predictors than the other methods considered in the experimental studies. To get better insight into the results of *GMean*, we have presented the boxplot for the best performing methods, including **BSIA**, see Fig. 5. It can be noticed that **BSIA** outperforms all methods and performs similarly to **UB** and **BSI**. However, it obtains better first quartile in comparison to **UB** and slightly higher value of minimum of *GMean*.

It is important to highlight that if we select lower significance rate α (e.g. 0.02) we are not allowed to reject the hypothesis that corresponds to the comparison between **BSIA** and **BSI**. Therefore, the deeper analysis of these two methods should be made. The computational complexity of the training procedure for **BSI** is equal $O(K \cdot N_{svm} \cdot N)$, where K is the total number of base learners, N_{svm} is the maximal number of supporting vectors for each of the constructed SVMs and N is total number of examples in training data. For the **BSIA** computational complexity is equal $O(K \cdot N_{svm,active} \cdot N_{active})$, where N_{active} represents maximal number of examples selected in the active learning procedure and $N_{svm,active}$ number of detected supporting vectors in the reduced data. Therefore, if the number of active examples is significantly lower than total number of cases ($N_{active} << N$), the computational costs and memory requirements for training **BSIA** are visibly lower.

Furthermore, we consider deeper comparison between **BSIA** and **BSI** in the context of imbalance ratio. For this purpose, we constructed two subsets of the training datasets considered in the previous experiment. The first one is gained by eliminating 10 datasets with the lowest imbalance ratio and the second one is obtained by excluding 10 datasets with the highest values of imbalance ratio. For the first subset, we gained the mean value of *GMean* for **BSIA** equal 0.8924 and for **BSI** equal 0.8843. For the Wilcoxon test, the p value was equal 0.0120. For the second subset of datasets the mean value of *GMean* was equal 0.8913 for **BSI** and 0.8931 for **BSI2**. The p value for that comparison was equal 0.2699. The presented results show that **BSIA** outperforms **BSI** when the imbalance ratio is extremely high. The high quality of **BSIA** comparing to the results gained by **BSI** was especially noticeable for datasets *Shuttle2vs4* ($ID = 35$, $\mathbf{Imb_{rate}} = 20.50$) and *Ecoli0137vs26* ($ID = 42$, $\mathbf{Imb_{rate}} = 39.15$) that have high imbalance ratio, but they do not construct as biased learner as for the other sets (see Fig. 4).

3.2 Experiment 2: the short-term loans repayment prediction

3.2.1 Description

In this work, we also consider the problem of 30-day loans risk assessment as a case study for the proposed classifier. The issue of credit risk modeling was initially considered by Durman in 1941, who first proposed the discriminant function that separates "bad" and "good" clients. Recent developments dedicated to solve the problem of constructing decision models that classify credit applicants make use of modern machine learning techniques such as neural networks (West 2000), Gaussian processes (Huang 2011), SVMs (Huang et al. 2007), or ensemble classifiers (Nanni and Lumini 2009). The modern learning methods indicate the necessity to deal with the imbalanced data issue (Huang et al. 2006; Zięba and Świątek 2012), as well as with the need of constructing the comprehensible predictors (Martens et al. 2007).

The short-term loans are typically easier to qualify for, both in terms of income and credit rating, than other types of credits. They are unsecured one-payment loans where no additional collateral is required as a basis for the approval. Moreover, the maximum loan amount varies, depending on the lender, from few hundred to thousands of dollars, relatively to the applicant's monthly income.

Our goal is to construct the best decision model that can be used to predict whether the applicant will be able to pay the short-term loan. As a suitable model, we recommend to

Table 3 Results of the experiment on the set of the benchmarks according to *GMean* criterion

ID	IB	SVM	SSVM	SBSVM	CSVM	AdaC	SBO	RUS	SB	UB	BSI	BSIA
1	1.82	0.0000	0.5567	0.6932	0.7140	0.7893	**0.8008**	0.7824	0.7518	0.7649	0.7416	0.7179
2	1.86	**0.9869**	0.9835	0.8327	0.9700	0.9695	0.9695	0.9765	0.9835	0.9800	0.9835	0.9800
3	1.86	0.9686	**0.9758**	0.9570	0.9463	0.9724	0.9633	0.9590	0.9641	0.9628	0.9728	0.9688
4	1.90	0.6963	0.7534	0.7436	0.7321	0.7159	0.7439	0.7331	**0.7609**	0.7602	0.7456	0.7456
5	2.00	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9899	0.9899	0.9899	0.9798	0.9899	**1.0000**	**1.0000**
6	2.06	0.4807	0.7069	0.7481	0.7742	0.8150	0.8150	**0.8557**	0.8269	0.8292	0.7782	0.7994
7	2.46	0.4522	0.7057	0.7033	0.7163	0.6460	0.7070	0.7059	**0.7294**	0.7225	0.7245	0.7274
8	2.52	0.5409	0.7899	0.8266	0.8299	0.7953	0.7438	0.7404	0.7710	0.7758	0.8413	**0.8451**
9	2.52	0.9376	0.9503	**0.9837**	0.9744	0.9813	0.9774	0.9758	0.9701	0.9595	0.9807	0.9752
10	2.52	0.3914	0.7668	0.8173	**0.8214**	0.7668	0.7388	0.7747	0.7555	0.7898	0.8204	0.8206
11	2.68	0.0000	0.5529	0.6199	0.6241	0.5598	0.6302	0.6258	0.6559	0.6620	0.6421	**0.6640**
12	3.19	0.8828	0.8940	0.8925	0.8925	0.9231	0.9028	0.9101	0.9231	0.9054	0.9141	**0.9337**
13	3.23	0.9504	0.9646	0.9652	**0.9779**	0.9765	0.9633	0.9601	0.9638	0.9525	0.9714	0.9739
14	3.36	0.8277	0.8973	0.8894	0.8798	0.8912	0.8776	**0.9115**	0.9035	0.9035	0.9015	0.8953
15	4.92	0.7928	0.9888	0.9710	0.9774	0.9574	0.9690	0.9547	0.9663	0.9494	0.9801	**0.9972**
16	5.14	0.7746	0.9860	0.9801	0.9944	0.9464	0.9832	0.9774	0.9746	0.9663	0.9916	**0.9972**
17	5.46	0.7719	0.9108	0.9238	0.9188	0.8815	0.9035	0.8835	0.8801	0.8947	0.9221	**0.9270**
18	6.01	0.9906	0.9934	0.9944	0.9947	0.9824	0.9959	0.9914	0.9929	0.9891	**0.9985**	0.9954
19	6.38	0.8440	0.8948	0.8686	0.8882	0.8873	0.8347	0.9130	**0.9209**	0.8969	0.8857	0.8711
20	8.11	0.7653	0.9177	0.8977	0.9068	0.8918	0.8932	0.9162	**0.9413**	0.9311	0.9191	0.9249
21	8.77	0.4106	0.8938	0.8673	0.8377	0.8215	0.8151	0.8713	0.8687	0.8902	0.8897	**0.8946**
22	8.77	0.6547	0.9539	0.9625	0.9604	**0.9977**	0.9966	0.9703	0.9898	0.9703	0.9775	0.9944
23	9.08	0.7402	0.8941	0.8712	0.8826	0.9195	0.8770	0.9131	0.9021	**0.9536**	0.8920	0.8961
24	9.35	0.0000	0.7948	0.7507	0.7424	0.7810	0.7726	**0.8444**	0.7973	0.7907	0.7907	0.7958
25	10.10	0.9713	0.9882	**1.0000**	**1.0000**	0.9702	0.9911	0.9577	0.9861	0.9477	**1.0000**	0.9978
26	10.29	0.0000	0.5615	0.5751	0.6193	0.5561	0.6059	0.5980	0.6600	0.7331	**0.7674**	0.7492
27	10.39	0.0000	0.5710	0.5757	0.7795	0.7187	0.7689	0.7043	**0.8355**	0.7697	0.8123	0.8127
28	13.84	0.8062	0.9244	0.8802	0.8859	0.9274	0.8802	0.9259	0.9290	0.8866	0.9259	**0.9336**
29	13.87	0.9959	0.9959	0.9956	0.9956	0.9997	**1.0000**	**1.0000**	0.9997	**1.0000**	0.9959	0.9997
30	13.87	0.0000	0.7511	0.5432	0.6906	0.7011	0.6325	0.7351	0.6522	0.7454	**0.7939**	0.7738
31	15.47	0.3922	0.9067	0.8216	0.8661	0.8810	0.9192	0.9267	0.8801	0.8572	0.9292	**0.9463**
32	15.85	0.7015	0.9057	0.9016	0.9344	0.7967	0.9343	0.9499	0.9563	**0.9599**	0.9337	0.9371
33	16.68	0.0000	0.8706	0.7206	0.8603	0.6904	0.7831	0.7847	0.7796	0.7731	**0.8989**	0.8960
34	19.44	0.0000	0.9502	0.8743	0.8118	0.8641	0.9292	**0.9885**	0.8537	0.9411	0.9827	0.9769
35	20.50	0.9092	0.9959	0.9129	0.9129	0.9129	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9129	**1.0000**
36	22.10	0.0000	0.6382	0.6662	0.5734	0.4208	0.4384	0.6190	0.5460	0.6424	0.6636	**0.6686**
37	22.81	0.0000	0.9422	0.7435	0.8125	0.9728	0.9828	0.8667	0.9195	0.9474	**0.9902**	0.9753
38	23.10	0.7408	0.7670	0.7408	0.6102	0.4984	0.7368	0.7705	**0.7975**	0.7623	0.7957	0.7966
39	28.41	0.0000	0.8125	0.6196	0.7731	0.6954	0.6600	0.8217	0.7474	**0.8477**	0.8141	0.8222
40	30.56	0.0000	0.6973	0.1820	0.6256	0.5771	0.5945	0.7453	0.5809	0.7149	0.7326	**0.7455**
41	32.78	0.2132	0.9661	0.8463	0.9401	0.8754	0.9090	0.9600	0.9630	0.9575	0.9477	**0.9699**
42	39.15	0.8421	0.8755	**0.9665**	0.7462	0.8153	0.8296	0.8121	0.8312	0.7539	0.8390	0.8966
43	39.15	0.0000	0.8763	0.7132	0.8640	0.6782	0.8019	0.8374	0.8245	0.8698	0.8887	**0.9009**
44	128.87	0.0000	0.6842	0.1759	0.6120	0.1753	0.1759	0.6847	0.3867	0.6904	0.7658	**0.7807**
AV		0.5098	0.8501	0.8003	0.8379	0.8088	0.8281	0.8596	0.8478	0.8634	0.8785	**0.8845**

Best results are in bold

Table 4 Results of Wilcoxon test made between **BSIA** and reference methods

Methods	p value	**FWER**	Hypothesis ($\alpha = 0.05$)
BSIA versus **SVM**	0.0000	0.0050	rejected for **BSIA**
BSIA versus **SSVM**	0.0000	0.0056	rejected for **BSIA**
BSIA versus **SBSVM**	0.0000	0.0063	rejected for **BSIA**
BSIA versus **CSVM**	0.0000	0.0071	rejected for **BSIA**
BSIA versus **AdaC**	0.0000	0.0083	rejected for **BSIA**
BSIA versus **SBO**	0.0000	0.0100	rejected for **BSIA**
BSIA versus **UB**	0.0006	0.0125	rejected for **BSIA**
BSIA versus **RUS**	0.0008	0.0167	rejected for **BSIA**
BSIA versus **SB**	0.0008	0.0250	rejected for **BSIA**
BSIA versus **BSI**	0.0255	0.0500	rejected for **BSIA**

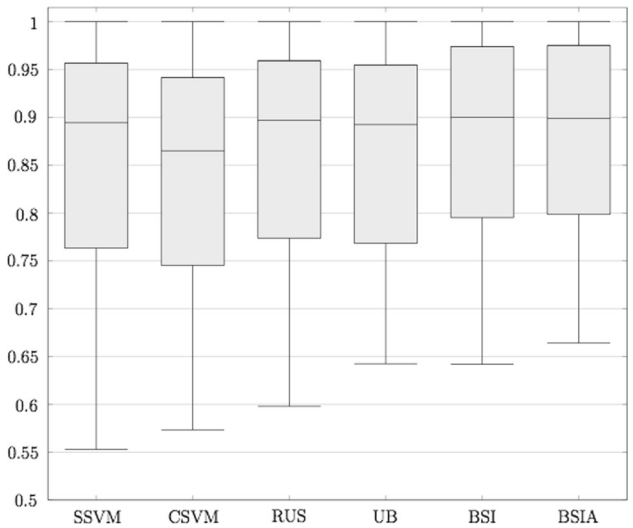

Fig. 5 Boxplot for the results of the experiment on the set of the benchmarks according to *GMean* criterion

apply *boosted SVM* trained with the active learning strategy presented in this paper. Therefore, we examined the quality of the solution in comparison to the reference methods on the real-life dataset gathered from a financial institution. In the experiment, we consider the most effective methods (basing on the results presented in Table 3) that deal with the imbalanced data issue: **UB**, **RUS**, **SSVM** and **BSI**. The intelligibility of the model is extremely important in the loan risk management domain. Therefore, we also took into account two comprehensible models, namely, decision rules inducer **JRip** and the algorithm for constructing decision trees **J48**. In addition, we applied the oracle-based procedure of decision rules induction which makes use of the *boosted SVM* trained with the active learning strategy to relabel the initial data. As the rule inducer we used **JRip** (we refer this approach in the experiment to as **JRip + BSIA**). Very similar approach was applied in Craven and Shavlik (1996) for neural netoworks and in Zięba et al. (2014) for SVMs.

Table 5 Results of the experiment for short-term loans dataset

Method	TP_{rate}	TN_{rate}	Acc	GMean
UB	0.6383	0.5930	0.5986	0.6153
RUS	0.4468	0.7393	0.7033	0.5747
SSVM	**0.6596**	0.5592	0.5716	0.6073
BSI	0.5957	0.6448	0.6387	0.6198
BSIA	0.6312	0.6388	0.6379	**0.6350**
JRip	0.0000	**1.0000**	**0.8770**	0.0000
J48	0.0000	**1.0000**	**0.8770**	0.0000
JRip + BSIA	0.6028	0.6537	0.6475	0.6277

Best results are in bold

The data used in the experiment were composed of 1,146 applicants, each described by 11 features including gender and age of the client, his monthly income and applied credit amount. We considered two-class problem in which the first class (assumed to be negative) represented the situation in which the consumer made timely repayment of the financial liability and the second class (positive) meant that the client had large problems with settling the debt. We identified a strong imbalance issue with negative/positive ratio equal 7.13.

3.2.2 Results and discussion

The results of the experiment are presented in Table 5. For the methods that have incorporated mechanisms of dealing with imbalance data issue, the most stable classifier was **BSIA** that gained the results of TP_{rate}, TN_{rate} and $GMean$ near 0.63. The other reference algorithms were slightly biased either towards minority class (**UB**, **SSVM**) or majority class (**RUS**, **BSI**) and received lower $GMean$ value than **BSIA**. The comprehensible models failed completely in loan repayment prediction for the considered dataset and were totally biased toward the majority class. However, the **JRip** rules inducer trained on the relabelled data by *boosted SVM* performed comparably to the strongest "black box" imbalance resistant models considered in the experiment. Therefore, we can successfully obtain a comprehensible model using the **BSIA** as the oracle.

4 Conclusion and future work

In this work, we proposed the novel method for constructing *boosted SVM* that makes use of active learning strategy to eliminate redundant instances and more properly estimate the misclassification costs. The outlined method was compared to the ensemble of SVMs as well as to other reference methods that consider the imbalanced data issue. The results

obtained within the experiment, i.e., on the representative number of benchmark datasets, supported by the statistical tests show that the presented modification of the training procedure improves the prediction ability of *boosted SVM* significantly. We also presented the real-life case study related to the problem of the short-term loans repayment prediction for which our solution achieved promising results comparing to other approaches. Moreover, we showed that our approach can be successfully applied as the oracle for rules induction which is an important issue in credit risk assessment.

Furthermore, we plan to adjust our model to the multi-class problem. This issue can be handled by applying a technique that combines two-class models, e.g., *one-versus-rest*. In addition, it would be beneficial to propose a tuning method for finding optimal width of the "wide margin". However, we leave investigating these issues as future research.

Acknowledgments The work conducted by Maciej Zięba is co-financed by the European Union within the European Social Fund.

References

Chawla NV, Bowyer KW, Hall LO (2002) SMOTE : synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Chawla NV, Lazarevic A, Hall LO, Bowyer K (2003) SMOTEBoost: improving prediction of the minority class in boosting. In proceedings of the principles of knowledge discovery in databases, PKDD-2003, Springer, pp 107–119

Chen S, He H, Garcia E (2010) RAMOBoost: ranked minority oversampling in boosting. Neural Netw IEEE Trans 21(10):1624–1642

Craven MW, Shavlik JW (1996) Extracting tree-structured representations of trained networks. Advances in neural information processing systems pp 24–30

Drummond C, Holte R (2000) Exploiting the cost (in)sensitivity of decision tree splitting criteria. In: proceedings of the seventeenth international conference on machine learning, Morgan Kaufmann, pp 239–246

Ertekin S, Huang J, Bottou L, Giles L (2007) Learning on the border: active learning in imbalanced data classification. In: proceedings of the sixteenth ACM conference on information and knowledge management ACM, pp 127–136

Ertekin S, Huang J, Giles C (2007) Active learning for class imbalance problem. In: proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, ACM pp 823–824

Fan W, Stolfo S, Zhang J, Chan P (1999) AdaCost: misclassification cost-sensitive boosting. In: proceedings 16th international conference on machine learning, Morgan Kaufmann, pp 97–105

Freund Y, Schapire RE, Hill M (1996) Experiments with a new boosting algorithm. In machine learning: proceedings of the thirteenth international conference

Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Syst Man Cybern Soc 42(4):3358–3378

García S, Fernández A, Herrera F (2009) Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. Appl Soft Comput 9(4):1304–1314

He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284. doi:10.1109/TKDE.2008.239

Hido S, Kashima H, Takahashi Y (2009) Roughly balanced bagging for imbalanced data. Stat Anal Data Min 2(5–6):412–426

Holm S (1979) A simple sequentially rejective multiple test procedure. Scand Stat Theory Appl pp 65–70

Huang CL, Chen MC, Wang CJ (2007) Credit scoring with a data mining approach based on support vector machines. Expert Syst Appl 33(4):847–856

Huang SC (2011) Using gaussian process based kernel classifiers for credit rating forecasting. Expert Syst Appl 38(7):8607–8611

Huang YM, Hung CM, Jiau HC (2006) Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. Nonlinear Anal Real World Appl 7(4):720–747

Hui H, Wang W, Mao B (2005) Borderline-SMOTE : a new oversampling method in imbalanced data sets learning. In: advances in intelligent computing pp 878–887

Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intel Data Anal 6(5):429–449

Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. In: proceedings of the fourteenth international conference on machine learning, Morgan Kaufmann, pp 179–186

Kukar M, Kononenko I (1998) Cost-sensitive learning with neural networks. In: proceedings of the 13th European conference on artificial intelligence, Wiley, pp 445–449

Manevitz L, Yousef M (2002) One-class SVMs for document classification. J Mach Learn Res 2:139–154

Mani J, Zhang I (2003) KNN approach to unbalanced data distributions: a case study involving information extraction. In: proceedings of international conference on machine learning, Workshop learning from iImbalanced data sets

Martens D, Baesens B, Van Gestel T, Vanthienen J (2007) Comprehensible credit scoring models using rule extraction from support vector machines. Eur J Oper Res 183(3):1466–1476

Morik K, Brockhausen P, Joachims T (1999) Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring. In: proceedings of international conference on machine learning, Morgan Kaufmann, pp 268–277

Nanni L, Lumini A (2009) An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. Expert Syst Appl 36(2):3028–3033

Seiffert C, Khoshgoftaar T, Van Hulse J, Napolitano A (2010) RUSBoost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern A Syst Hum 40(1):185–197

Settles B (2010) Active learning literature survey. University of Wisconsin, Madison

Tang Y, Zhang Y, Huang Z (2007) Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. IEEE/ACM Trans Comput Biol Bioinform 4(3):365–381

Tao D, Tang X, Li X, Wu X (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal Mach Intell 28(7):1088–1099

Vapnik V (1998) Statistical learning theory. Wiley

Veropoulos K, Campbell C, Cristianini N (1999) Controlling the sensitivity of support vector machines. Proc Int Joint Conf Artif Intell 1999:55–60

Wang BX, Japkowicz N (2010) Boosting support vector machines for imbalanced data sets. Knowl Inf Syst 25(1):1–20

Wang S, Yao X (2009) Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE symposium on computational intelligence and data mining, IEEE pp 324–331

West D (2000) Neural network credit scoring models. Comput Oper Res 27(11):1131–1152

Zięba M, Tomczak JM, Świątek J, Lubicz M (2014) Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. Appl Soft Comput 14(Part A):99–108. doi:10.1016/j.asoc.2013.07.016

Zięba M, Świątek J (2012) Ensemble classifier for solving credit scoring problems. Technological innovation for value creation pp 59–66