Applied Soft Computing 14 (2014) 99–108

Contents lists available at ScienceDirect

Applied Soft Computing

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

# Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients

Maciej Zięba*, Jakub M. Tomczak, Marek Lubicz, Jerzy Świątek

*Faculty of Computer Science and Management, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland*

ARTICLE INFO

*Article history:*
Received 27 March 2013
Received in revised form 12 July 2013
Accepted 22 July 2013
Available online 6 September 2013

*Keywords:*
Imbalanced data
Boosted SVM
Decision rules
Post-operative life expectancy prediction

ABSTRACT

In this paper, we present boosted SVM dedicated to solve imbalanced data problems. Proposed solution combines the benefits of using ensemble classifiers for uneven data together with cost-sensitive support vectors machines. Further, we present oracle-based approach for extracting decision rules from the boosted SVM. In the next step we examine the quality of the proposed method by comparing the performance with other algorithms which deal with imbalanced data. Finally, boosted SVM is used for medical application of predicting post-operative life expectancy in the lung cancer patients.



## 1. Introduction

The main difficulty in learning classification models is the character of data. Usually, raw data gathered from many sources cannot be used directly in the training process due to various circumstances, i.e., missing values of some attributes [1], sequential nature of delivering data [2], or disproportions in class distribution [3]. In the literature, the third issue is known as *imbalanced data problem*. In general, each dataset with unequal class distribution can be considered as imbalanced. In practice, the problem of disproportion between classes occurs when the classifier trained with typical methods has a tendency to make decisions biased toward majority class. Extreme imbalance data problem is observed when the classifier trained using traditional methods is classifying all objects only to the majority class, independently on the vector of features. Applying techniques which deal with the problem of unequally distributed data is essential for learning decision models with high predictive accuracy.

The problem of imbalanced data is widely observed in medical decision making, particularly in post-operative risk evaluation domain. Considering the short period of planning (1 year) the number of patients surviving the assumed interval is often significantly higher than the number of deceases. Moreover, the misclassification related to treating deceases as survivals is much more troublesome than the decision mistake made in opposite direction.

The second important issue connected to the problem of post-operative life expectancy prediction is the interpretability of the decision model. Using so-called "black box" models in the considered application is not recommended. It is mainly caused by the patient's fear about being treated by machines with hidden and difficult-to-understand inference process and distrust among doctors of being supported by vaguely-working models. To dissipate the doubts it is necessary to propose a method to extract knowledge in the form of decision rules or trees from "black box" models.

The main goal of this paper is to propose a general decision model which obtains high predictive accuracy in case of imbalanced data and use this model for extraction of interpretable knowledge in the form of decision rules. The core of our proposition is a generalization of the learning task for SVM by introducing individual penalty parameters for each example, and then application of AdaBoost-like algorithm to learn ensemble of SVMs. We call this approach *boosted* SVM for imbalanced data. At the end, we apply *boosted* SVM as an oracle to re-label examples and use the new dataset for the rules induction in order to obtain an interpretable model.

Additionally, we aim at using thoracic surgery clinical diagnostics and treatment issues as an important application of the boosted

* Corresponding author. Tel.: +48 71 320 44 53.
*E-mail addresses:* maciej.zieba@pwr.wroc.pl (M. Zięba), jakub.tomczak@pwr.wroc.pl (J.M. Tomczak), marek.lubicz@pwr.wroc.pl (M. Lubicz), jerzy.swiatek@pwr.wroc.pl (J. Świątek).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.07.016

SVM proposed in this paper. Our approach may contribute current clinical treatment twofold. First, to support decisions on patient selection for surgery made by clinicians. Second, to identify the cases of higher risk of patient's death after surgery by extracting decision rules.

The paper is organized as follows. In Section 2 we review approaches that are applied to imbalanced data and solutions used to extract rules from uninterpretable models. In Section 3 we present the ensemble SVM classifier (*boosted* SVM) for imbalanced data and describe the method of extracting decision rules from the model. Section 4 contains the results of an experiment showing the quality of the proposed approach for imbalanced data and presents post-operative life expectancy rules extracted using *boosted* SVM. This paper is briefly summarized in Section 5.

## 2. Related work

Typical methods used to learn classifiers are biased toward the majority class if they are trained with imbalanced data. Various techniques are used to deal with the stated problem which can be divided into three categories [3,4]:

- *Data level (external) approaches*, which use oversampling and undersampling techniques to equalize the number of instances from classes.
- *Algorithm level (internal) approaches*, which incorporate balancing techniques in training process.
- *Cost-sensitive solutions*, which use both data level transformations (by adding costs to instances) together with algorithm level modifications (by modifying the learning process to accept costs).

The first group of methods operates independently on the training procedure by modifying the class distribution, either by generating artificial samples or by eliminating non-informative examples. The most popular and commonly used sampling method is SMOTE (Synthetic Minority Over-sampling TEchnique) [5], which generates artificial examples situated on the path connecting two neighbours from minority class. Borderline-SMOTE is an extension of SMOTE, which incorporates in the sampling procedure only the minority data points located in the neighbourhood of the separating hyperplane [6]. On the other hand, non-informative majority examples can be eliminated to obtain more balanced data in the undersampling procedure. This type of methods is mainly used as a component of more sophisticated, internal approaches, and detection of non-informative examples is usually made by random selection, using *K-NN* algorithm [7] or with evolutionary algorithms [8].

The second group of methods solves the problem of uneven data directly during training phase, and it includes ensemble-based approaches which make use of under- and oversampling methods to construct diverse and balanced base classifiers. The most common approach in this group is *SMOTEBoost* [9], that combines the benefits of boosting with multiple sampling procedure using SMOTE. Alternatively, synthetic data sampling can be included in the process of constructing bagging-based ensemble [10]. Some of ensemble solutions make use of undersampling techniques to construct class-unbiased base learners [11,12].

Other important group of inner methods uses granular computing techniques to solve the problem of uneven data [13–16]. The main feature of such approaches is knowledge-oriented decomposition of the main problem into parallel, balanced sub-problems, named *information granules*. *Active learning* solutions are also used to deal with imbalanced data issue [17,18]. Originally, *active learning* was used to detect informative, unlabelled examples and query them about class values to create representative training set. Application of *active learning* techniques for imbalanced data is based on the assumption that the distribution of detected examples is significantly more equalized than the distribution of examples from the initial dataset. According to [17], the data points accumulated near the borderline tends to be more equalized than the points in the entire dataset.

The last group of methods assigns a weight to each element of the training set. In general, the process of weighting the examples is made to increase the significance of the minority observations at the expense of the majority class. Large set of cost-sensitive methods make use of ensemble classifiers which update the weights while constructing base learners. As a consequence, the weights related with minority examples are updated stronger than examples from majority class if they are misclassified by already constructed base classifiers. In this group we can distinguish mainly boosting-based approaches such as: *AdaCost* [19], *CSB1*, *CSB2* [20], *RareBoost* [21], *AdaC1*, *AdaC2*, *AdaC3* [22].

Cost-sensitive techniques are widely applied together with *Support Vector Machines* (SVM). The different misclassification costs for classes are considered in learning criterion to achieve soft margin unbiased toward majority class. The approaches differ in the manner of including cost values in penalization term of learning criterion to construct balanced SVM [23,24]. An interesting extension of typical cost-sensitive approaches is SVM with boosting proposed in [25].

In real-life applications, there are two major tasks to be solved: (i) prediction based on properties learned from data, and (ii) discovery of unknown properties from data. In the field of machine learning there are several successful approaches that achieve high predictive accuracy, e.g., neural networks and SVM. On the other hand, there are methods that allow to understand the considered phenomenon and they are easily interpretable by human, e.g., tree-based or rule-based models. Unfortunately, the methods with high predictive accuracy are hard to interpret and the understandable models performs poorly in the prediction task. Therefore, there arises a need to combine high predictive performance with interpretability of the model.

In the literature, there are three main approaches to extract understandable rules from uninterpretable models with high predictive accuracy [26]. The first one, called *decompositional*, focuses on extracting rules from the structure of the trained model, e.g., from neural networks [27] or margin determined by SVM [28]. The second approach, called *pedagogical*, aims at generating examples from the trained model or to re-label training examples according to the trained model and then applying a method for rules extraction from data, e.g., using neural networks [29]. All other techniques which do not fall clearly into one of the above categories are called *eclectic*, e.g., rules extraction from SVM [30].

## 3. Boosted SVM for imbalanced data

### 3.1. SVM for imbalanced data

#### 3.1.1. Problem statement

Let $\mathbf{x}$ be a $M$-dimensional vector of inputs, $\mathbf{x} \in \mathcal{X}$, and $y$ – an output (or class label), $y \in \{-1, 1\}$. For given data $\mathcal{S} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, we are interested in finding a discriminative hyperplane of the following form:

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

where $\mathbf{a}$ is a vector of weights, $\mathbf{a} \in \mathbb{R}^M$, $\phi(\mathbf{x})$ denotes a fixed feature-space transformation, and $b$ is *bias*, $b \in \mathbb{R}$.

The solution of the problem of determining adaptive parameters of the discriminative hyperplane is known as Support Vector Machines (SVM) [31]. The typical learning task for SVM is stated as follows:

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

where $\xi_n$ 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, $C > 0$.

In the problems where data is well-balanced the SVM achieves high predictive accuracy. However, it may fail in case of imbalanced datasets. One of the possible solutions for this problem is to include cost-sensitivity of classes [32]. In other words, instead of one penalty parameter $C$ there are two real-valued parameters $C_+$ and $C_-$, namely, for minority and majority classes.[1] It is implemented by transforming the learning task for SVM into:

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

where $\mathcal{N}_+ = \{n \in \{1, \ldots, N\} : y_n = 1\}$, and $\mathcal{N}_- = \{n \in \{1, \ldots, N\} : y_n = -1\}$.

Further, we generalize the problem (3) by introducing individual penalty parameters $w_n$ for each $n$th observation which yields:

$$\begin{aligned} \min_{\mathbf{a},b} \quad & Q(\mathbf{a}) = \frac{1}{2}\mathbf{a}^\top\mathbf{a} + C\sum_{n=1}^{N}w_n\xi_n \\ \text{s.t.} \quad & y_n(\mathbf{a}^T\phi(\mathbf{x}_n) + b) \geq 1 - \xi_n \\ & \text{for all } n = 1, \ldots, N \end{aligned} \tag{4}$$

Because we have maintained the individual penalty parameter $C$, the penalty parameters $w_n$ have to fulfill the following condition:

$$\sum_{n=1}^{N} w_n = N. \tag{5}$$

The introduction of $\mathbf{w}$ allows to diversify the costs of misclassification not only between classes but also within classes. Further, we will use the penalty parameters $\mathbf{w}$ in the boosting procedure during the classifiers construction. For the primal optimization problem (4) we get the following dual optimization problem[2]:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & Q_D(\boldsymbol{\lambda}) = \sum_{n=1}^{N}\lambda_n - \frac{1}{2}\sum_{i,j=1}^{N}\lambda_i\lambda_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \leq \lambda_n \leq Cw_n \\ & \sum_{n=1}^{N}\lambda_n y_n = 0 \\ & \text{for all } n = 1, \ldots, N \end{aligned} \tag{6}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers.

The solution of the quadratic optimization task in the form (6) is optimal iff the matrix with the elements $y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$ is semipositive-definite and for all $n = 1, \ldots, N$ the Kuhn-Tucker conditions are fulfilled:

$$\begin{aligned} & \lambda_n = 0, \\ & 0 < \lambda_n < Cw_n, \\ & \lambda_n = Cw_n. \end{aligned} \tag{7}$$

In practice, only some of the Lagrange multipliers $\lambda_n$ will be nonzero and all examples for which $\lambda_n \geq 0$ are called *support vectors*. Finally, the classification function is expressed by:

$$y(\mathbf{x}_i) = \text{sign}\left(\sum_{n\in\mathcal{SV}} y_n\lambda_n k(\mathbf{x}_n, \mathbf{x}_i) + b\right), \tag{8}$$

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

$$b = \frac{1}{N_{\mathcal{SV}}}\sum_{n\in\mathcal{SV}}\left(y_n - \sum_{m\in\mathcal{SV}}\lambda_m y_m k(\mathbf{x}_n, \mathbf{x}_m)\right), \tag{9}$$

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

In the literature, there are several training algorithms for SVM like *chunking* [33], *Osuna's algorithm* [34], and *Sequential Minimal Optimization* (SMO) [35] with its modification [36]. Further, we will use SMO algorithm for solving the optimization task (6).

The difference between solutions of different SVM formulations is presented in Fig. 1. For the imbalanced data the solution of the typical formulation of SVM (2) has a tendency

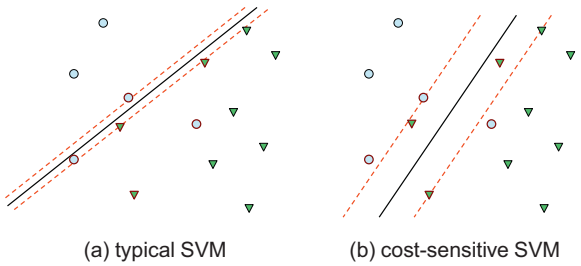


**Fig. 1.** Optimal hyperplane for SVM trained on imbalanced dataset: (a) using the typical formulation (2), (b) using the cost-sensitive formulation (4). (a) Typical SVM. (b) Cost-sensitive SVM.

[1] We use the following convention. The data points within the minority class are called *positive examples*, i.e., $y = 1$, while data with majority class label, $y = -1$ – *negative examples*.

[2] In the following formulation, we have already applied the kernel trick, i.e., we have replaced the inner product with the kernel function, $\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$.

to *move* towards examples from the majority class (Fig. 1a). Application of different values of penalty parameters for positive and negative examples prevents this undesirable phenomenon (Fig. 1b).

#### 3.1.2. Weights determination

The crucial issue in the optimization task (6) is the determination of penalty weights **w**. In our studies we use the following weights' values:

$$w_n = \begin{cases} \dfrac{N}{2N_+}, & \text{for } n \in \mathcal{N}_+ \\ \dfrac{N}{2N_-}, & \text{for } n \in \mathcal{N}_- \end{cases} \tag{10}$$

where $N_+$, and $N_-$ are cardinalities of the sets $\mathcal{N}_+$, and $\mathcal{N}_-$, respectively. The values (10) fulfil the equations:

$$\sum_{n\in\mathcal{N}_+} w_n = \sum_{n\in\mathcal{N}_-} w_n = \frac{N}{2}, \tag{11}$$

and thus the condition (5) is also fulfilled. Such fashion of defining weights solves only the problem of between-class imbalanceness, without distinguishing weight values within the class. To solve this issue we propose boosted algorithm to determine the different weight values and make use of (10) only in the process of weights initialization.

### 3.2. Boosted SVM for imbalanced data

Ensemble classifiers are usually used to improve the prediction accuracy of so called *weak learners*. In this work, we propose boosted SVM to solve inner and between-class imbalanced data problems. In this considered case, ensemble approach is used to determine the weights' values in SVM learning criterion during external sequential error minimization in boosting loop. The training procedure of boosting SVM is performed in two alternating steps: by solving optimization problem (6) for fixed values of **w**, and by iterative updating **w** in external optimization process of exponential error:

$$E_{\exp,Imb} = \frac{1}{N_+}\sum_{n\in\mathcal{N}_+} \exp(-y_n f_k(\mathbf{x}_n)) + \frac{1}{N_-}\sum_{n\in\mathcal{N}_-} \exp(-y_n f_k(\mathbf{x}_n)), \tag{12}$$

where $f_k(\mathbf{x}_n)$ is the linear combination of $k$ base classifiers $h_l(\mathbf{x}_n)$[3]:

$$f_k(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{k} c_l h_l(\mathbf{x}_n), \tag{13}$$

where $c_l$ is the weight of $l$th base classifier $h_l$, $l = 1, \ldots, k$.

[3] Constant 1/2 is used for convenience to prove that presented boosting method minimizes sequentially the exponential error function.

#### 3.2.1. Boosting framework for imbalanced data – external procedure

**Algorithm 1** *(General boosting algorithm for imbalanced data).*

**Input** : $\mathcal{S}$: training set, $\mathcal{S}_{val}$: validation set, $\mathcal{Y} = \{-1, 1\}$: set of class labels, $K$: number of iterations

**Output**: Boosted classifier for imbalanced data:

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

```
Initialize: D_1(n) ⟵ 1/N for n ∈ {1, ..., N} ;
G ⟵ 0;
K_final ⟵ 1;
for k = 1 → K do
    Train base learner h_k by minimizing error e_k
    given by (14);
    Calculate e_k on S achieved by h_k;
    if e_k < 0.5 then
        c_k ⟵ ln (1−e_k)/e_k;
        Calculate Gmean value g_k on S_val achieved
        by H_k(x) = arg max_{y∈Y} Σ_{l=1}^{k} c_l I(h_l(x) = y) ;
        if g_k > G then
            G ⟵ g_k;
            K_final ⟵ k;
        end
        Update:
        D_{k+1}(n) ⟵ D_k(n) exp(c_k I(h_k(x_n) ≠ y_n)) ;
        Normalize: D_{k+1}(n) ⟵ D_{k+1}(n) / Σ_{n=1}^{N} D_{k+1}(n) ;
    else
        c_k ⟵ 0;
        Initialize: D_1(n) ⟵ 1/N ;
        Change parameters for training base
        classifiers ;
end
```

Algorithm 1 presents general boosting method for imbalanced data that minimizes exponential error function given by (12). In the first step the initial values for the distribution (data weighting coefficients) $D_1(n)$ are equal $1/N$. Next, the values of boosting coefficient are iteratively updated in the following manner. First, base learner $h_k$ is built by minimizing weighted error function:

$$e_k = \frac{E_{Imb}}{(1/N_+)\sum_{n\in\mathcal{N}_+} D_k(n) + (1/N_-)\sum_{n\in\mathcal{N}_-} D_k(n)} \tag{14}$$

where $E_{Imb}$ is equal:

$$E_{Imb} = \frac{1}{N_+}\sum_{n\in\mathcal{N}_+} D_k(n) I(h_k(\mathbf{x}_n) \neq y_n) + \frac{1}{N_-}\sum_{n\in\mathcal{N}_-} D_k(n) I(h_k(\mathbf{x}_n) \neq y_n), \tag{15}$$

where $I(\cdot)$ denotes the indicator function.

**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 problem of uneven data is solved by proposing weighted error function with different misclassification costs, $1/N_+$ and $1/N_-$, for positive and negative examples respectively.

The value of $e_k$ is calculated on the training data $\mathcal{S}$ and is further used to compute coefficient $c_k$ of $k$th base learner. Finally, the data weights are updated using typical *AdaBoost* [37] procedure to increase the significance of misclassified examples.

Additionally, basing on *boosting* SVM training procedure described in [25], we select only the subset of base classifiers with the highest *geometric mean* (*Gmean*) value. *Gmean* is one of the most commonly used quality rates for imbalanced data and is described by equation:

$$Gmean = \sqrt{TP_{rate} \cdot TN_{rate}}, \tag{16}$$

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

$$TN_{rate} = \frac{TN}{TN + FP}, \tag{17}$$

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

$$TP_{rate} = \frac{TP}{TP + FN} \tag{18}$$

True positive (*TP*), false negative (*FN*), false positive (*FP*) and true negative (*TN*) are situated in confusion matrix (see Table 1), which illustrates prediction tendencies of considered classifier.

### 3.2.2. Why does it work?

The training procedure sequentially minimizes exponential error described by Eq. (12) and maximizes *Gmean* in parallel. The process of maximization of *Gmean* value is performed simply by selecting subsequence of base classifiers performing best on validation set $\mathcal{S}_{val}$. The justification for sequential minimization of (12) is more complex and will be discussed below.

We assume that the training process in boosting loop is reduced to estimation $c_1, \ldots, c_k$ and base classifiers $h_1, \ldots, h_k$ by minimizing error function (12). The optimization process is made sequentially, therefore in each iteration values $c_1, \ldots, c_{k-1}$, and base learners $h_1, \ldots, h_{k-1}$ are fixed and the process of estimation is made with respect to $c_k$ and $h_k$. Exponential error function (12) can be presented in the following form:

$$\begin{aligned} E_{\exp,Imb} &= \frac{1}{N_+} \sum_{n \in \mathcal{N}_+} \exp\left(-y_n f_{k-1}(\mathbf{x}_n) - \frac{1}{2} y_n c_k h_k(\mathbf{x}_n)\right) \\ &\quad + \frac{1}{N_-} \sum_{n \in \mathcal{N}_-} \exp\left(-y_n f_{k-1}(\mathbf{x}_n) - \frac{1}{2} y_n c_k h_k(\mathbf{x}_n)\right) \\ &= \frac{1}{N_+} \sum_{n \in \mathcal{N}_+} D_k(n) \exp\left(-\frac{1}{2} y_n c_k h_k(\mathbf{x}_n)\right) \\ &\quad + \frac{1}{N_-} \sum_{n \in \mathcal{N}_-} D_k(n) \exp\left(-\frac{1}{2} y_n c_k h_k(\mathbf{x}_n)\right), \end{aligned} \tag{19}$$

where $D_k(n) = \exp(-y_n f_{k-1}(\mathbf{x}_n))$ is a constant value, because it is independent on $c_k$ and $h_k$. Next, we assume that $\mathcal{T}_+$ represents total number of positive examples classified correctly by $h_k$ and $\mathcal{F}_+$ the rest, misclassified positive examples. Analogously, we denote cardinalities of correctly and incorrectly classified negative examples by $\mathcal{T}_-$ and $\mathcal{F}_-$, respectively. Hence, we can rewrite error function in the the following form:

$$\begin{aligned} E_{\exp,Imb} &= \frac{1}{N_+} \exp\left(-\frac{c_k}{2}\right) \sum_{n \in \mathcal{T}_+} D_k(n) + \frac{1}{N_+} \exp\left(\frac{c_k}{2}\right) \sum_{n \in \mathcal{F}_+} D_k(n) \\ &\quad + \frac{1}{N_-} \exp\left(-\frac{c_k}{2}\right) \sum_{n \in \mathcal{T}_-} D_k(n) + \frac{1}{N_-} \exp\left(\frac{c_k}{2}\right) \sum_{n \in \mathcal{F}_-} D_k(n). \end{aligned} \tag{20}$$

Minimizing exponential error with respect to $c_k$ gives:

$$\exp(c_k^*) = \frac{(1/N_+)\sum_{n \in \mathcal{T}_+} D_k(n) + (1/N_-)\sum_{n \in \mathcal{T}_-} D_k(n)}{(1/N_+)\sum_{n \in \mathcal{F}_+} D_k(n) + (1/N_-)\sum_{n \in \mathcal{F}_-} D_k(n)}. \tag{21}$$

Making use of the fact:

$$\begin{aligned} &\frac{1}{N_+} \sum_{n \in \mathcal{T}_+} D_k(n) + \frac{1}{N_-} \sum_{n \in \mathcal{T}_-} D_k(n) \\ &= \frac{1}{N_+} \left( \sum_{n \in \mathcal{N}_+} D_k(n) - \sum_{n \in \mathcal{F}_+,} D_k(n) \right) \\ &\quad + \frac{1}{N_-} \left( \sum_{n \in \mathcal{N}_-} D_k(n) - \sum_{n \in \mathcal{F}_-,} D_k(n) \right), \end{aligned} \tag{22}$$

and putting $e_k$ given by Eq. (14) to (21) we get:

$$\exp(c_k^*) = \frac{1 - e_k}{e_k} \tag{23}$$

Taking logarithm on both sides we gain formula for calculating optimal value of $c_k$ (see step 8, Algorithm 1).

For optimization process with respect to base classifier $h_k$, we present error function in the following manner:

$$\begin{aligned} E_{\exp,Imb} &= \left(\exp\left(\frac{-c_k}{2}\right) + \exp\left(\frac{c_k}{2}\right)\right) \\ &\quad \times \left( \frac{1}{N_+} \sum_{n \in \mathcal{N}_+} D_k(n) I(h_k(\mathbf{x}_n) \neq y_n) \right. \\ &\quad \left. + \frac{1}{N_-} \sum_{n \in \mathcal{N}_-} D_k(n) I(h_k(\mathbf{x}_n) \neq y_n) \right) + \exp\left(\frac{-c_k}{2}\right) \\ &\quad \times \left( \frac{1}{N_+} \sum_{n \in \mathcal{N}_+} D_k(n) + \frac{1}{N_-} \sum_{n \in \mathcal{N}_-} D_k(n) \right) \end{aligned} \tag{24}$$

It can be noticed, that the minimization of exponential error respecting $h_k$ is reduced to minimization of the error function given by Eq. (15), because the second term in the sum is constant and the value of $(\exp(-c_k/2) + \exp(c_k/2))$ does not affect the location of the minimum. In step 5 (Algorithm 1) the optimal classifier is constructed by minimization of (14), which is normalized value of error given by (15).

### 3.3. SVM as base learner – internal procedure

Algorithm 1 presents general procedure of sequential minimizing exponential error function given by Eq. (12). In previous

subsection we have shown that the process of building each of base learners is performed by minimization of the weighted error function given by (15). The problem of error function minimization is stated theoretically, because $\mathcal{S}$ is a sample taken from the entire data space. Optimizing error only on training set may result in overfitting. In this work we present boosted *SVM* for imbalanced data (further named *BoostingSVM-IB*) and the theoretical problem formulation will be transformed into optimization task presented in (6). For each boosting iteration penalty parameters $w_n^{(k)}$ in (6) are set in the following fashion:

$$w_n^{(k)} = \begin{cases} \frac{N}{2N_+} D_k(n) & \text{for } n \in \mathcal{N}_+ \\ \frac{N}{2N_-} D_k(n) & \text{for } n \in \mathcal{N}_- \end{cases} \tag{25}$$

$K$th SVM base learner is built by solving optimization problem (6) with actual $w_n^{(k)}$ penalty values.

### 3.4. Rules extraction from boosted SVM for imbalanced data

In the previous section we have outlined the boosted SVM, an ensemble of SVM classifiers. Additionally, we have justified that it works in case of imbalanced datasets. Hence, we will use this model for decision rules induction.

**Algorithm 2** *(Rules extraction process using BoostingSVM-IB).*

**Input** : $\mathcal{S}$: training set, $H(\mathbf{x})$: *BoostingSVM-IB* classifier
**Output**: $\mathcal{R}$: set of decision rules
$\tilde{\mathcal{S}} \longleftarrow \emptyset$ ;
Train $H(\mathbf{x})$ on $\mathcal{S}$ with optimal parameters;
**foreach** $(\mathbf{x}_n, y_n) \in \mathcal{S}$ **do**
  $\tilde{y}_n \longleftarrow H(\mathbf{x}_n)$;
  $\tilde{\mathcal{S}} \longleftarrow \tilde{\mathcal{S}} \cup \{(\mathbf{x}_n, \tilde{y}_n)\}$ ;
**end**
Construct set of decision rules $\mathcal{R}$ from $\tilde{\mathcal{S}}$ using rules inducer;

Once the proposed model obtains high predictive accuracy, it seems proper to use it as an oracle for re-labelling examples. The procedure of extracting rules is presented in Algorithm 2. In the first step we train boosted SVM, $H(\mathbf{x})$, on entire training set $\mathcal{S}$.[4] Next, each example is re-labelled with output $\tilde{y}_n$ returned by $H(\mathbf{x})$ ($\tilde{\mathcal{S}}$ denotes dataset after re-labelling procedure). Finally, set of decision rules is obtained by applying a rules inducer, e.g., RIPPER algorithm [38]. This is a typical application of the *pedagogical approach*.

## 4. Experiments

The main issue of this paper is to propose the novel *BoostingSVM-IB* algorithm for imbalanced data which is further used in the rule extraction. Therefore, we perform two experiments. First, the performance of the boosted SVM is examined on a set of 44 benchmark imbalanced datasets respecting *Gmean* value criterion. Second, the proposed approach with boosted SVM and rules extraction is applied to the prediction of the post-operative life expectancy in the lung cancer patients.

### 4.1. Benchmark datasets

In the first experiment we aim at evaluating predictive performance of the proposed boosted SVM. Stratified cross-validation is used as a research methodology in the experiment. The *BoostingSVM-IB* algorithm was implemented in *KEEL Software* [39,40] for comparative analysis with other methods dedicated for imbalanced problems.

**Table 2**
Characteristic of datasets used in experiment [4].

| Dataset | # *Inst.* | # *Attr.* | %*P* | %*N* | $Imb_{rate}$ |
|---|---|---|---|---|---|
| Glass1 | 214 | 9 | 35.51 | 64.49 | 1.82 |
| Ecoli0vs1 | 220 | 7 | 35.00 | 65.00 | 1.86 |
| Wisconsin | 683 | 9 | 35.00 | 65.00 | 1.86 |
| Pima | 768 | 8 | 34.84 | 66.16 | 1.90 |
| Iris0 | 150 | 4 | 33.33 | 66.67 | 2.00 |
| Glass0 | 214 | 9 | 32.71 | 67.29 | 2.06 |
| Yeast1 | 1484 | 8 | 28.91 | 71.09 | 2.46 |
| Vehicle1 | 846 | 18 | 28.37 | 71.63 | 2.52 |
| Vehicle2 | 846 | 18 | 28.37 | 71.63 | 2.52 |
| Vehicle3 | 846 | 18 | 28.37 | 71.63 | 2.52 |
| Haberman | 306 | 3 | 27.42 | 73.58 | 2.68 |
| Glass0123vs456 | 214 | 9 | 23.83 | 76.17 | 3.19 |
| Vehicle0 | 846 | 18 | 23.64 | 76.36 | 3.23 |
| Ecoli1 | 336 | 7 | 22.92 | 77.08 | 3.36 |
| New-thyroid2 | 215 | 5 | 16.89 | 83.11 | 4.92 |
| New-thyroid1 | 215 | 5 | 16.28 | 83.72 | 5.14 |
| Ecoli2 | 336 | 7 | 15.48 | 84.52 | 5.46 |
| Segment0 | 2308 | 19 | 14.26 | 85.74 | 6.01 |
| Glass6 | 214 | 9 | 13.55 | 86.45 | 6.38 |
| Yeast3 | 1484 | 8 | 10.98 | 89.02 | 8.11 |
| Ecoli3 | 336 | 7 | 10.88 | 89.77 | 8.77 |
| Page-blocks0 | 5472 | 10 | 10.23 | 89.77 | 8.77 |
| Yeast2vs4 | 514 | 8 | 9.92 | 90.08 | 9.08 |
| Yeast05679vs4 | 528 | 8 | 9.66 | 90.34 | 9.35 |
| Vowel0 | 988 | 13 | 9.01 | 90.99 | 10.10 |
| Glass016vs2 | 192 | 9 | 8.89 | 91.11 | 10.29 |
| Glass2 | 214 | 9 | 8.78 | 91.22 | 10.39 |
| Ecoli4 | 336 | 7 | 6.74 | 93.26 | 13.84 |
| Yeast1vs7 | 459 | 8 | 6.72 | 93.28 | 13.87 |
| Shuttle0vs4 | 1829 | 9 | 6.72 | 93.28 | 13.87 |
| Glass4 | 214 | 9 | 6.07 | 93.93 | 15.47 |
| Page-blocks13vs2 | 472 | 10 | 5.93 | 94.07 | 15.85 |
| Abalone9vs18 | 731 | 8 | 5.65 | 94.25 | 16.68 |
| Glass016vs5 | 184 | 9 | 4.89 | 95.11 | 19.44 |
| Shuttle2vs4 | 129 | 9 | 4.65 | 95.35 | 20.5 |
| Yeast1458vs7 | 693 | 8 | 4.33 | 96.67 | 22.10 |
| Glass5 | 214 | 9 | 4.20 | 95.80 | 22.81 |
| Yeast2vs8 | 482 | 8 | 4.15 | 95.85 | 23.10 |
| Yeast4 | 1484 | 8 | 3.43 | 96.57 | 28.41 |
| Yeast1289vs7 | 947 | 8 | 3.17 | 96.83 | 30.56 |
| Yeast5 | 1484 | 8 | 2.96 | 97.04 | 32.78 |
| Ecoli0137vs26 | 281 | 7 | 2.49 | 97.51 | 39.15 |
| Yeast6 | 1484 | 8 | 2.49 | 97.51 | 39.15 |
| Abalone9 | 4174 | 8 | 0.77 | 99.23 | 128.87 |

The performance of *BoostingSVM-IB* method is examined using 44 benchmark imbalanced datasets available in *KEEL* tool and on website.[5] Multiclass datasets are modified to obtain two-class imbalanced data by merging some of possible class values [4]. 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 ratio between negative and positive examples.

The quality of *BoostingSVM-IB* (**BSI**) method was compared with other methods working in imbalanced fashion:

- *SVM* (**SVM**): SVM trained using SMO.
- *SVM +SMOTE* (**SSVM**): SVM trained on data oversampled by *SMOTE*.

[4] The proposed method is invariant on classifier, so *strong learners* can be used instead.

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

**Table 3**
Results of Wilcoxon test made between **BSI** and the strongest methods from Tables 4 and 5.

| Methods | $R^+$ | $R^-$ | Hypothesis ($\alpha = 0.05$) | $p$-Value |
|---|---|---|---|---|
| **BSI** vs. **UB** | 715.0 | 275.0 | rejected for **BSI** | 0.01007 |
| **BSI** vs. **RUS** | 709.0 | 281.0 | rejected for **BSI** | 0.01231 |
| **BSI** vs. **SSVM** | 779.0 | 167.0 | rejected for **BSI** | 0.00021 |

- *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 [24,41].
- *AdaCost* (**AdaC**): Cost-sensitive, ensemble classifier, in which the misclassification cost for minority class is higher than the misclassification cost for majority class [19].
- *SMOTEBoost* (**SBO**): Modified *AdaBoost* algorithm, in which base classifiers are constructed using *SMOTE* synthetic sampling [9].
- *RUSBoost* (**RUS**): Extension of *SMOTEBoost* approach, which uses additional undersampling in each boosting iteration [42].
- *SMOTEBagging* (**SB**): Bagging method, which uses *SMOTE* to oversample dataset before constructing each of base classifiers [10].
- *UnderBagging* (**UB**): Bagging method, which randomly undersamples dataset before constructing each of base classifiers [12].

Two groups of methods were selected for the analysis: SVM-based methods which deal with the problem of imbalanced data and ensemble based approaches presented and evaluated in [4]. In the first group of methods we have distinguished approaches which modify SVM classifier to enable adjusting to unequal data distribution. Additionally, we included simple SVM trained with SMO to compare the results with the method dedicated to be used for balanced data. The second group of methods is composed of ensemble techniques which achieved the highest results in empirical studies carried out in [4].

The detailed results are presented in Table 4 (**BSI** was compared with other SVM-based solutions) and in Table 5 (similar comparison was made for ensemble-based approaches). The highest average *Gmean* value was achieved by **BSI** method in both cases. For further analysis we used *Wilcoxon* non-parametric statistical test to investigate the differences in between results obtained by **BSI** and three other methods with the highest *Gmean* value: **UB**, **RUS** and **SSVM**. The results for the sum of positive and negative ranks and $p$-value for each pairwise comparison are presented in Table 3. Assuming the level of significance at 0.05 hypothesis about median equality were rejected for all pairs, what practically means that **BSI** significantly outperforms other considered methods.

### 4.2. Thoracic surgery dataset

One of the main clinical decision problems in thoracic surgery (TS) is the appropriate patient selection for surgery, taking into account risk and benefits for a patient, both in short term (e.g., post-operative complications, including 30-days mortality) and also under longer term perspective (e.g., 1-year or 5-year survival). Traditional methods aim at incorporating standard statistical modelling, based on Kaplan–Meier survival curves, hierarchical statistical models, multivariable logistic regression or Cox proportional hazards regression [43–46]. Particular sets of predictors and their relative importance in post-operative survival or complications prediction are reported as results suggested by standard statistical software packages, while other authors formulate explicit regression-type models [47] or develop web-based applications, like Thoracoscore [48–50] one of the standard-like TS risk stratification scoring systems. Taking into account the limitations of both classic statistical approaches and hospital datasets, which are often incomplete, due to missing values of attributes and unknown survivals for prospective analysis, and also – aiming at increasing not only formal prediction accuracy (which can easily be high with heavily imbalanced data) but also other performance measures like *AUC* or *Gmean*, other multivariable approaches have been proposed for the TS domain. Those include data mining and machine learning procedures for standardizing TS data analysis and report generation [51,52], as well as applications based on particular techniques, like decision trees [53,54] or artificial neural networks [55,56].

#### 4.2.1. Dataset description

In this chapter we report application of the proposed approach in modelling surgical risk for real-life clinical data from Thoracic Surgery domain. The data was collected retrospectively at Wroclaw Thoracic Surgery Centre for over 1200 consecutive patients who underwent major lung resections for primary lung cancer in the years 2007–2011. The Centre is associated with the Department of Thoracic Surgery of the Medical University of Wroclaw and Lower-Silesian Centre for Pulmonary Diseases, Poland, while the research database constitutes a part of the National Lung Cancer Registry, administered by the Institute of Tuberculosis and Pulmonary Diseases in Warsaw, Poland. The main dataset included 139 predictors, of which 36 from pre-operative, 37 from peri-operative, and 46 (including 17 pathology-related) from post-operative periods.

In this paper, we make use of the *BoostingSVM-IB* method to extract decision rules for the problem of 1-year survival period basing on the vector of attributes composed of 36 from pre-operative features.

#### 4.2.2. Features selection

First, the feature selection method was applied to evaluate the worth of each attribute. According to [57] this step is recommended in medical decision problems with imbalanced data. We used information gain criterion [58] at this stage, which resulted in reducing the number of features from 36 to 16 attributes. Features selected for further analysis are described in Table 6. Attributes *PRE4*, *PRE5* and *AGE* are numeric, *PRE14*, *DGN* and *PRE6* are nominal and the rest of features are binary.[6]

In the next step examples with missing values are eliminated from data.[7] The final dataset used in the experiment was composed of 470 examples and the imbalanced rate equalled 5.71.

#### 4.2.3. Method evaluation

Basing on the results from previous experiment we examined the quality of methods **UB**, **RUS**, **SSVM** and **BSI** for the problem of predicting post-operative life expectancy (see Table 7). Research methodology was the same as in the benchmark datasets experiment. According to *Gmean* criterion **BSI** method performed slightly better than the other three algorithms. Comparable results were achieved by **UB** but this ensemble technique has tendency to overbalance, what can be observed in higher values of $TP_{rate}$ and lower accuracy. Analysis of the results gathered in Tables 4, 5 and 7 leads to the conclusion that the most appropriate method for further rule extraction is **BSI**.

**BSI** is a highly uninterpretable classifier because it combines two "difficult to understand" models: SVM and ensembles. The oracle method for generating decision rules seems to be the best solution

[6] T(rue) if a symptom or clinical condition occurs, (F)alse otherwise

[7] The simplest method of dealing with missing values is proposed because the goal of this work is to present the novel approach for dealing with imbalanced data problem and extracting decision rules with oracle approach in the application of thoracic post-operative life expectancy, not to solve the problem itself.

**Table 4**
Detailed test results table for SVM-based methods vs. *BoostingSVM-IB*.

| Dataset | SVM | SSVM | SBSVM | CSVM | BSI |
|---|---|---|---|---|---|
| Glass1 | 0.0 | 55.7 | 69.3 | 71.4 | 74.2 |
| Ecoli0vs1 | **98.7** | 98.3 | 83.3 | 97.0 | 98.3 |
| Wisconsin | 96.9 | **97.6** | 95.7 | 94.6 | 97.3 |
| Pima | 69.6 | 75.3 | 74.4 | 73.2 | 74.6 |
| Iris0 | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** |
| Glass0 | 48.1 | 70.7 | 74.8 | 77.4 | 77.8 |
| Yeast1 | 45.2 | 70.6 | 70.3 | 71.6 | 72.5 |
| Vehicle1 | 54.1 | 79.0 | 82.7 | 83.0 | 84.1 |
| Vehicle2 | 93.8 | 95.0 | **98.4** | 97.4 | 98.1 |
| Vehicle3 | 39.1 | 76.7 | 81.7 | 82.1 | 82.0 |
| Haberman | 0.0 | 55.3 | 62.0 | 62.4 | 64.2 |
| Glass0123vs456 | 88.3 | 89.4 | 89.3 | 89.3 | 91.4 |
| Vehicle0 | 95.0 | 96.5 | 96.5 | **97.8** | 97.1 |
| Ecoli1 | 82.8 | 89.7 | 88.9 | 88.0 | 90.1 |
| New-thyroid2 | 79.3 | 98.9 | 97.1 | 97.7 | 98.0 |
| New-thyroid1 | 77.5 | 98.6 | 98.0 | 99.4 | 99.2 |
| Ecoli2 | 77.2 | 91.1 | 92.4 | 91.9 | 92.2 |
| Segment0 | 99.1 | 99.3 | 99.4 | 99.5 | **99.8** |
| Glass6 | 84.4 | 89.5 | 86.9 | 88.8 | 88.6 |
| Yeast3 | 76.5 | 91.8 | 89.8 | 90.7 | 91.9 |
| Ecoli3 | 41.1 | 89.4 | 86.7 | 83.8 | 89.0 |
| Page-blocks0 | 65.5 | 95.4 | 96.3 | 96.0 | 97.8 |
| Yeast2vs4 | 74.0 | 89.4 | 87.1 | 88.3 | 89.2 |
| Yeast05679vs4 | 0.0 | 79.5 | 75.1 | 74.2 | 79.1 |
| Vowel0 | 97.1 | 98.8 | **100.0** | **100.0** | **100.0** |
| Glass016vs2 | 0.0 | 56.2 | 57.5 | 61.9 | **76.7** |
| Glass2 | 0.0 | 57.1 | 57.6 | 78.0 | 81.2 |
| Ecoli4 | 80.6 | 92.4 | 88.0 | 88.6 | 92.6 |
| Shuttle0vs4 | 99.6 | 99.6 | 99.6 | 99.6 | 99.6 |
| Yeast1vs7 | 0.0 | 75.1 | 54.3 | 69.1 | **79.4** |
| Glass4 | 39.2 | 90.7 | 82.2 | 86.6 | 92.9 |
| Page-blocks13vs2 | 70.2 | 90.6 | 90.2 | 93.4 | 93.4 |
| Abalone9vs18 | 0.0 | 87.1 | 72.1 | 86.0 | 89.9 |
| Glass016vs5 | 0.0 | 95.0 | 87.4 | 81.2 | 98.3 |
| Shuttle2vs4 | 90.9 | 99.6 | 91.3 | 91.3 | 91.3 |
| Yeast1458vs7 | 0.0 | 63.8 | 66.6 | 57.3 | 66.4 |
| Glass5 | 0.0 | 94.2 | 74.4 | 81.3 | **99.0** |
| Yeast2vs8 | 74.1 | 76.7 | 74.1 | 61.0 | 79.6 |
| Yeast4 | 0.0 | 81.2 | 62.0 | 77.3 | 81.4 |
| Yeast1289vs7 | 0.0 | 69.7 | 18.2 | 62.6 | 73.3 |
| Yeast5 | 21.3 | 96.6 | 84.6 | 94.0 | 94.8 |
| Ecoli0137vs26 | 84.2 | 87.5 | **96.7** | 74.6 | 83.9 |
| Yeast6 | 0.0 | 87.6 | 71.3 | 86.4 | 88.9 |
| Abalone9 | 0.0 | 68.4 | 17.6 | 61.2 | 76.6 |
| Average | 51.0 | 85.0 | 80.0 | 83.8 | **87.9** |

The best results for each dataset are bolded.

**Table 5**
Detailed test results table for ensemble-based methods vs. *BoostingSVM-IB*.

| Dataset | AdaC | SBO | RUS | SB | UB | BSI |
|---|---|---|---|---|---|---|
| Glass1 | 78.9 | **80.1** | 78.2 | 75.2 | 76.5 | 74.2 |
| Ecoli0vs1 | 97.0 | 97.0 | 97.7 | 98.3 | 98.0 | 98.3 |
| Wisconsin | 97.2 | 96.3 | 95.9 | 96.4 | 96.3 | 97.3 |
| Pima | 71.6 | 74.4 | 73.3 | **76.1** | 76.0 | 74.6 |
| Iris0 | 99.0 | 99.0 | 99.0 | 98.0 | 99.0 | **100.0** |
| Glass0 | 81.5 | 81.5 | **85.6** | 82.7 | 82.9 | 77.8 |
| Yeast1 | 64.6 | 70.7 | 70.6 | 72.9 | 72.2 | 72.5 |
| Vehicle1 | 79.5 | 74.4 | 74.0 | 77.1 | 77.6 | 84.1 |
| Vehicle2 | 98.1 | 97.7 | 97.6 | 97.0 | 95.9 | 98.1 |
| Vehicle3 | 76.7 | 73.9 | 77.5 | 75.6 | 79.0 | 82.0 |
| Haberman | 56.0 | 63.0 | 62.6 | 65.6 | 66.2 | 64.2 |
| Glass0123vs456 | 92.3 | 90.3 | 91.0 | 92.3 | 90.5 | 91.4 |
| Vehicle0 | 97.7 | 96.3 | 96.0 | 96.4 | 95.3 | 97.1 |
| Ecoli1 | 89.1 | 87.8 | **91.2** | 90.3 | 90.4 | 90.1 |
| New-thyroid2 | 95.7 | 96.9 | 95.5 | 96.6 | 94.9 | 98.0 |
| New-thyroid1 | 94.6 | 98.3 | 97.7 | 97.5 | 96.6 | 99.2 |
| Ecoli2 | 88.1 | 90.4 | 88.4 | 88.0 | 89.5 | 92.2 |
| Segment0 | 98.2 | 99.6 | 99.1 | 99.3 | 98.9 | **99.8** |
| Glass6 | 88.7 | 83.5 | 91.3 | **92.1** | 89.7 | 88.6 |
| Yeast3 | 89.2 | 89.3 | 91.6 | **94.1** | 93.1 | 91.9 |
| Ecoli3 | 82.2 | 81.5 | 87.1 | 86.9 | 89.0 | 89.0 |
| Page-blocks0 | **99.8** | 99.7 | 97.0 | 99.0 | 97.0 | 97.8 |
| Yeast2vs4 | 91.9 | 87.7 | 91.3 | 90.2 | **95.4** | 89.2 |
| Yeast05679vs4 | 78.1 | 77.3 | **84.4** | 79.7 | 79.1 | 79.1 |
| Vowel0 | 97.0 | 99.1 | 95.8 | 98.6 | 94.8 | **100.0** |
| Glass016vs2 | 55.6 | 60.6 | 59.8 | 66.0 | 73.3 | **76.7** |
| Glass2 | 71.9 | 76.9 | 70.4 | **83.6** | 77.0 | 81.2 |
| Ecoli4 | 92.7 | 88.0 | 92.6 | 92.9 | 88.7 | 92.6 |
| Shuttle0vs4 | **100.0** | **100.0** | **100.0** | **100.0** | **100.0** | 99.6 |
| Yeast1vs7 | 70.1 | 63.2 | 73.5 | 65.2 | 74.5 | **79.4** |
| Glass4 | 88.1 | 91.9 | 92.7 | 88.0 | 85.7 | 92.9 |
| Page-blocks13vs2 | 79.7 | 93.4 | 95.0 | 95.6 | **96.0** | 93.4 |
| Abalone9vs18 | 69.0 | 78.3 | 78.5 | 78.0 | 77.3 | 89.9 |
| Glass016vs5 | 86.4 | 92.9 | **98.9** | 85.4 | 94.1 | 98.3 |
| Shuttle2vs4 | 91.3 | **100.0** | **100.0** | **100.0** | **100.0** | 91.3 |
| Yeast1458vs7 | 42.1 | 43.8 | 61.9 | 54.6 | 64.2 | 66.4 |
| Glass5 | 97.3 | 98.3 | 86.7 | 92.0 | 94.7 | **99.0** |
| Yeast2vs8 | 49.8 | 73.7 | 77.1 | **79.7** | 76.2 | 79.6 |
| Yeast4 | 69.5 | 66.0 | 82.2 | 74.7 | 84.8 | 81.4 |
| Yeast1289vs7 | 57.7 | 59.5 | **74.5** | 58.1 | 71.5 | 73.3 |
| Yeast5 | 87.5 | 90.9 | 96.0 | 96.3 | 95.8 | 94.8 |
| Ecoli0137vs26 | 81.5 | 83.0 | 81.2 | 83.1 | 75.4 | 83.9 |
| Yeast6 | 67.8 | 80.2 | 83.7 | 82.4 | 87.0 | 88.9 |
| Abalone9 | 17.5 | 17.6 | 68.5 | 38.7 | 69.0 | 76.6 |
| Average | 80.9 | 82.8 | 86.0 | 84.8 | 86.3 | **87.9** |

The best results for each dataset are bolded.

because of the high quality of **BSI** and independence on learning process and simplicity of the approach.

Table 7 shows the results obtained by typical rule extraction method named RIPPER (also named **JRip**) algorithm. It can be observed (see Table 7) that applying **JRip** alone for the medical problem resulted in constructing only one decision rule (*Risk1Y*=F), according to which each example should be classified to the majority class ($TP_{rate}$ = 0.00, $TN_{rate}$ = 100.00). Detecting cases in which patient dies in 1 year period is crucial in post-operative risk management domain. Therefore, it is essential to extract decision rules that covers minority examples, even at the expense of high accuracy decreases in majority class. Table 7 contains also the results obtained by using relabelling technique with **BSI** as oracle before extracting rules by **JRip** (**JRip** +**BSI**). Results achieved by **JRip** trained using relabelled data were comparable with results gained by **BSI**, which means that the rules imitates the uninterpretable model at a satisfactory level.

### 4.2.4. Extracted rules

The process of extracting rules using **BSI** as an oracle resulted in generating 9 decision rules presented in Table 8. In the machine learning literature, in order to evaluate induced rules coverage and

**Table 6**
Characteristic of selected pre-operative features.

| ID | Description | InfoGain |
|---|---|---|
| *PRE14* | T in clinical TNM (size of the original tumour, from OC11 (smallest) to OC14 (largest)) | 0.029 |
| *DGN* | Diagnosis (specific combination of ICD-10 codes for primary and secondary as well multiple tumours if any) | 0.013 |
| *PRE4* | Forced vital capacity (FVC) | 0.008 |
| *PRE7* | Pain (pre-surgery) | 0.008 |
| *AGE* | Age at surgery | 0.008 |
| *PRE6* | Performance status (Zubrod scale) | 0.007 |
| *PRE11* | Weakness (pre-surgery) | 0.004 |
| *PRE9* | Dyspnoea (pre-surgery) | 0.004 |
| *PRE10* | Cough (pre-surgery) | 0.003 |
| *PRE8* | Haemoptysis (pre-surgery) | 0.003 |
| *PRE25* | PAD (peripheral arterial diseases) | 0.003 |
| *PRE19* | MI up to 6 months | 0.003 |
| *PRE5* | Volume that has been exhaled at the end of the first second of forced expiration (FEV1) | 0.002 |
| *PRE32* | Asthma | 0.002 |
| *PRE30* | Smoking | 0.002 |
| *PRE17* | Type 2 DM (diabetes mellitus) | 0.002 |
| *Risk1Y* | 1 year survival period ((T)rue value if died) | |

**Table 7**
Results for thoracic surgery data.

| Method | $TP_{rate}$ | $TN_{rate}$ | Gmean |
|---|---|---|---|
| **UB** | **68.57** | 61.75 | 65.07 |
| **RUS** | 52.85 | 65.50 | 58.84 |
| **SSVM** | 57.14 | 68.25 | 62.45 |
| **BSI** | 60.00 | 72.00 | **65.73** |
| **JRip** | 0.00 | **100.00** | 0.00 |
| **JRip +BSI** | 60.00 | 70.00 | 64.81 |

The best results for each dataset are bolded.

**Table 8**
Rules extracted from thoracic surgery data.

| Rules | Coverage | Accuracy |
|---|---|---|
| (DGN = DGN5) => Risk1Yr = T | 0.03 | 0.47 |
| (PRE14 = OC14) => Risk1Yr = T | 0.04 | 0.41 |
| (PRE17 = T) and (PRE30 = T) and (AGE > = 57) =>Risk1Yr=T | 0.05 | 0.38 |
| (PRE11 = T) and (PRE5 < = 2.16) and (PRE4 > = 2.44) =>Risk1Yr = T | 0.05 | 0.35 |
| (PRE9 = T) and (AGE > = 54) and (PRE5 < = 66.4) =>Risk1Yr = T | 0.05 | 0.35 |
| (PRE14 = OC13) => Risk1Yr = T | 0.04 | 0.32 |
| (DGN = DGN2) and (PRE30 = T) and (PRE14 = OC12) and (PRE5 < = 3.72) =>Risk1Yr = T | 0.04 | 0.30 |
| (PRE8 = T) and (PRE30 = T) and (PRE4 < = 3.52) =>Risk1Yr = T | 0.08 | 0.26 |
| OTHERWISE =>Risk1Yr = F | 0.62 | 0.97 |

accuracy measures are typically used. The coverage measure determines the percentage of examples covered by rule on average while the accuracy measure expresses the relative frequency of examples satisfying the complete rule among those satisfying only the antecedent. The accuracy for rules combined with the minority class was from 0.26 to 0.47. Each of the rules covered from 3% to 8% of the training data. Remaining subspace of features covered 62% of examples with accuracy of 97% for the majority class.

Results show that it is possible to identify cases of higher risk of patient's death after surgery by applying oracle-based approach combined with boosted SVM method. Extracted rules together with coverage and accuracy values give important information about patients suffering special treatment due to high risk of death. It is also important to highlight, that patients uncovered by the minority rules in 97% of cases will survive the considered survival period.

## 5. Conclusions

In this paper, we have proposed a novel *boosted* SVM method for imbalanced data problem which was further used for rules extraction. We have evaluated the quality of the proposed approach by comparing it with other solutions dedicated for imbalanced data problem. Next, we have used the proposed method to solve the problem for prediction of the post-operative life expectancy in the lung cancer patients. We have shown that our approach can be successfully applied to the problem by making additional experimental comparison on real-life dataset. Finally, we extracted decision rules using oracle-based approach.

## Acknowledgements

The research by Maciej Zięba was co-financed by the European Union as part of the European Social Fund.

The research by Marek Lubicz was partly financed by the National Science Centre under the grant N N115 090939 "Models and Decisions in the Health Systems. Application of Operational Research and Information Technologies for Supporting Managerial Decisions in the Health Systems".

## References

[1] P.J. Garcia-Laencina, J.L. Sancho-Gomez, A.R. Figueiras-Vidal, Pattern classification with missing data: a review, Neural Computing and Applications 19 (2009) 263–282.
[2] T. Dietterich, Machine learning for sequential data: a review, in: T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, D. de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Springer, Berlin Heidelberg, 2002, pp. 227–246.
[3] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 1263–1284.
[4] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews 42 (2012) 3358–3378.
[5] N.V. Chawla, K.W. Bowyer, L.O. Hall, SMOTE: Synthetic Minority Over-sampling TEchnique, Journal of Artificial Intelligence Research 16 (2002) 321–357.
[6] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, 2005, pp. 878–887.
[7] I. Zhang, J. Mani, KNN approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of International Conference on Machine Learning (ICML 2003), Workshop Learning from Imbalanced Data Sets, 2003.
[8] S. García, A. Fernández, F. Herrera, Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems, Applied Soft Computing 9 (2009) 1304–1314.
[9] N. Chawla, A. Lazarevic, L. Hall, K. Bowyer, Smoteboost: improving prediction of the minority class in boosting, in: N. Lavrac, D. Gamberger, H. Blockeel, L. Todorovski (Eds.), Knowledge Discovery in Databases: PKDD 2003, Springer, Berlin Heidelberg, 2003, pp. 107–119.
[10] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: 2009 IEEE Symposium on Computational Intelligence and Data Mining Proceedings, 2009, pp. 324–331.
[11] E. Chang, B. Li, G. Wu, K. Goh, Statistical learning for effective visual information retrieval, in: IEEE Proceedings of the 2003 International Conference on Image Processing, vol. 3, 2003, pp. 609–613.
[12] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1088–1099.
[13] Y. Tang, B. Jin, Y. Zhang, Granular support vector machines with association rules mining for protein homology prediction, Artificial Intelligence in Medicine 35 (2005) 121–134.
[14] Y. Tang, B. Jin, Y. Zhang, H. Fang, B. Wang, Granular support vector machines using linear decision hyperplanes for fast medical binary classification, in: IEEE Proceedings of the 14th IEEE International Conference on Fuzzy Systems, 2005, pp. 138–142.
[15] Y. Tang, Y. Zhang, Granular svm with repetitive undersampling for highly imbalanced protein homology prediction, in: Y.-Q. Zhang, T.Y. Lin (Eds.), 2006 IEEE International Conference on Granular Computing, 2006, pp. 457–460.
[16] Y. Tang, Y. Zhang, Z. Huang, Development of two-stage svm-rfe gene selection strategy for microarray expression data analysis, IEEE/ACM Transactions on Computational Biology and Bioinformatics 4 (2007) 365–381.
[17] S. Ertekin, J. Huang, L. Bottou, L. Giles, Learning on the border: active learning in imbalanced data classification, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, 2007, pp. 127–136.
[18] S. Ertekin, J. Huang, C. Giles, Active learning for class imbalance problem, in: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, ACM, 2007, pp. 823–824.
[19] W. Fan, S. Stolfo, J. Zhang, P Chan, Adacost: misclassification cost-sensitive boosting, in: I. Bratko, S. Dzeroski (Eds.), Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Morgan Kaufmann, 1999, pp. 97–105.
[20] K. Ting, A comparative study of cost-sensitive boosting algorithms, in: P. Langley (Ed.), Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Morgan Kaufmann, 2000, pp. 983–990.
[21] M. Joshi, V. Kumar, R. Agarwal, Evaluating boosting algorithms to classify rare classes: Comparison and improvements, in: N. Cercone, T.Y. Lin, X. Wu (Eds.), Proceedings. 2001 IEEE International Conference on Data Mining, IEEE, Los Alamitos, 2001, pp. 257–264.
[22] Y. Sun, M. Kamel, A. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (2007) 3358–3378.
[23] K. Morik, P. Brockhausen, T. Joachims, Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring, in: I. Bratko, S. Dzeroski (Eds.), Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Morgan Kaufmann, 1999, pp. 268–277.

[24] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of support vector machines, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence, (IJCAI999), Workshop ML3, vol. 1999, 1999, pp. 55–60.
[25] B. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, Knowledge and Information Systems 25 (2010) 1–20.
[26] A. Tickle, R. Andrews, M. Golea, J. Diederich, The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEEE Transactions on Neural Networks 9 (1998) 1057–1068.
[27] J. Chorowski, J.M. Zurada, Extracting rules from neural networks as decision diagrams, IEEE Transactions on Neural Networks 22 (2011) 2435–2446.
[28] H. Nú nez, C. Angulo, A. Català, Rule extraction from support vector machines, in: Proceedings of the European Symposium on Artificial Neural Networks, 2002, pp. 107–112.
[29] M. Craven, J. Shavlik, Rule extraction: where do we go from here?, Technical Report, University of Wisconsin Machine Learning Research Group Working Paper, 1999.
[30] N. Barakat, J. Diederich, Eclectic rule-extraction from support vector machines, International Journal of Computational Intelligence 2 (2005) 59–62.
[31] I. Steinwart, A. Christmann, Support Vector Machines, Springer, New York, 2008.
[32] H. Masnadi-Shirazi, N. Vasconcelos, Risk minimization, probability elicitation, and cost-sensitive SVMs, in: J. Fürnkranz, T. Joachims (Eds.), Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, 2010, pp. 204–213.
[33] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, Inc., New York, 1998.
[34] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: J. Principe, L. Gile, N. Morgan, E. Wilson (Eds.), Neural Networks for Signal Processing VII, Proceedings of the 1997 IEEE Signal Processing Workshop, IEEE, 1997, pp. 276–285.
[35] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, 1999.
[36] S. Keerthi, S. Shevade, C. Bhattacharyya, K. Murthy, Improvements to platt's smo algorithm for svm classifier design, Neural Computation 13 (2001) 637–649.
[37] Y. Freund, R.E. Schapire, M. Hill, Experiments with a New Boosting Algorithm, in: L. Saitta (Ed.), Machine Learning, Proceedings of the Thirteenth International Conference (ICML'96), Morgan Kaufmann, 1996.
[38] W.W. Cohen, Fast effective rule induction, in: A. Prieditis, S. Russell (Eds.), in Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, Tahoe City, CA, 1995, pp. 115–123.
[39] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-valued Logic and Soft Computing 17 (2010) 255–287.
[40] A. Fernández, J. Luengo, J. Derrac, J. Alcalá-Fdez, F. Herrera, Implementation and integration of algorithms into the keel data-mining software tool, in: E. Corchado, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning-IDEAL 2009, Springer, Berlin, 2009, pp. 562–569.
[41] Y. Tang, Y. Zhang, N. Chawla, S. Krasser, Svms modeling for highly imbalanced classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (2009) 281–288.
[42] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost A hybrid approach to alleviating class imbalance, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 40 (2010) 185–197.
[43] M. Shapiro, S.J. Swanson, C.D. Wright, C. Chin, S. Sheng, J. Wisnivesky, T.S. Weiser, Predictors of major morbidity and mortality after pneumonectomy utilizing the society for thoracic surgeons general thoracic surgery database, Annals of Thoracic Surgery 90 (2010) 927–935.
[44] U. Aydogmus, L. Cansever, Y. Sonmezoglu, K. Karapinar, C.I. Kocaturk, M.A. Bedirhan, The impact of the type of resection on survival in patients with n1 non-small-cell lung cancers, European Journal of Cardio-Thoracic Surgery 37 (2010) 446–450.
[45] P. Icard, M. Heyndrickx, L. Guetti, F. Galateau-Salle, P. Rosat, J.P. Le Rochais, J.-L. Hanouz, Morbidity, mortality and survival after 110 consecutive bilobectomies over 12 years, Interactive Cardiovascular and Thoracic Surgery 16 (2013) 179–185.
[46] D. Shahian, F. Edwards, Statistical risk modeling and outcomes analysis, Annals of Thoracic Surgery 86 (2008) 1717–1720.
[47] R. Berrisford, A. Brunelli, G. Rocco, T. Treasure, M. Utley, The european thoracic surgery database project: modelling the risk of in-hospital death following lung resection, European Journal of Cardio-Thoracic Surgery 28 (2005) 306–311.
[48] P.E. Falcoz, M. Conti, L. Brouchet, S. Chocron, M. Puyraveau, M. Mercier, J.P. Etievent, M. Dahan, The thoracic surgery scoring system (thoracoscore): risk model for in-hospital death in 15,183 patients requiring thoracic surgery, The Journal of Thoracic and Cardiovascular Surgery 133 (2007) 325–332.
[49] A. Barua, S.D. Handagala, L. Socci, B. Barua, M. Malik, N. Johnstone, A.E. Martin-Ucar, Accuracy of two scoring systems for risk stratification in thoracic surgery, Interactive Cardiovascular and Thoracic Surgery 14 (2012) 556–559.
[50] G. Rocco, ecomment. re: Accuracy of two scoring systems for risk stratification in thoracic surgery, Interactive Cardiovascular and Thoracic Surgery 14 (2012) 559.
[51] E. Rivo, J. de la Fuente, Á. Rivo, E. García-Fontán, M.-Á. Ca nizares, P. Gil, Cross-industry standard process for data mining is applicable to the lung cancer surgery domain, improving decision making as well as knowledge and quality management, Clinical and Translational Oncology 14 (2012) 73–79.
[52] N. Voznuka, H. Granfeldt, A. Babic, M. Storm, U. Lönn, H.C. Ahn, Report generation and data mining in the domain of thoracic surgery, Journal of Medical Systems 28 (2004) 497–509.
[53] J. Dowie, M. Wildman, Choosing the surgical mortality threshold for high risk patients with stage Ia non-small cell lung cancer: Insights from decision analysis, Thorax 57 (2002) 7–10.
[54] M.K. Ferguson, J. Siddique, T. Karrison, Modeling major lung resection outcomes using classification trees and multiple imputation techniques, European Journal of Cardio-Thoracic Surgery 34 (2008) 1085–1089.
[55] H. Esteva, T.G. Nú nez, R.O. Rodríguez, Neural networks and artificial intelligence in thoracic surgery, Thoracic Surgery Clinics 17 (2007) 359–367.
[56] G. Santos-Garcıa, G. Varela, N. Novoa, M.F. Jiménez, Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble, Artificial Intelligence in Medicine 30 (2004) 61–69.
[57] C.-Y. Lee, Z.-J. Lee, A novel algorithm applied to classify unbalanced data, Applied Soft Computing 12 (2012) 2481–2485.
[58] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: D.H. Fisher (Ed.), Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Morgan Kaufmann, 1997, pp. 412–420.