Central European Journal of Operations Research (2022) 30:1403–1425
https://doi.org/10.1007/s10100-021-00782-1

# Constrained Naïve Bayes with application to unbalanced data classification

Rafael Blanquero[1,4] · Emilio Carrizosa[1,4] · Pepa Ramírez-Cobo[2,4] · M. Remedios Sillero-Denamiel[3,4]

Accepted: 15 September 2021 / Published online: 20 October 2021

## Abstract

The Naïve Bayes is a tractable and efficient approach for statistical classification. In general classification problems, the consequences of misclassifications may be rather different in different classes, making it crucial to control misclassification rates in the most critical and, in many realworld problems, minority cases, possibly at the expense of higher misclassification rates in less problematic classes. One traditional approach to address this problem consists of assigning misclassification costs to the different classes and applying the Bayes rule, by optimizing a loss function. However, fixing precise values for such misclassification costs may be problematic in realworld applications. In this paper we address the issue of misclassification for the Naïve Bayes classifier. Instead of requesting precise values of misclassification costs, threshold values are used for different performance measures. This is done by adding constraints to the optimization problem underlying the estimation process. Our findings show that, under a reasonable computational cost, indeed, the performance measures under consideration achieve the desired levels yielding a user-friendly constrained classification procedure.

**Keywords** Probabilistic classification · Constrained optimization · Parameter estimation · Efficiency measures · Naïve Bayes

✉ M. Remedios Sillero-Denamiel
sillerom@tcd.ie

[1] Departamento de Estadística e Investigación Operativa, Universidad de Sevilla, Seville, Spain

[2] Departamento de Estadística e Investigación Operativa Universidad de Cádiz, Cádiz, Spain

[3] School of Computer Science and Statistics, Trinity College Dublin (TCD), Dublin, Ireland

[4] IMUS, Instituto de Matemáticas de la Universidad de Sevilla, Seville, Spain

# 1 Introduction

Naïve Bayes (NB) is a classification technique that has played a prominent role in the literature. Hand and Yu (2001), Hastie et al. (2001) and Mehra and Gupta (2013) highlight its tractability, simplicity and efficiency. The implicit hypothesis of independent attributes conditioned to the class eases its implementation significantly because it allows to express the sample likelihood to be maximized as the product of univariate marginals. Moreover, this classifier is less prone to overfitting since it estimates fewer parameters than other current classification techniques (Domingos and Pazzani 1997; Hand and Yu 2001). As a consequence, NB has been applied in a number of real contexts, for example, genetics (Chandra and Gupta 2011; Minnier et al. 2015), medicine [see Wei et al. 2011; Rosen et al. 2010; Parthiban et al. 2011; Wolfson et al. 2015], risk (Minnier et al. 2015), reliability (Turhan and Bener 2009; Menzies et al. 2007), document analysis (Bermejo et al. 2011; Guan et al. 2014) and a number of variants have been proposed in the literature [see Jiang et al. 2016; Boullé 2007; Wu et al. 2015; Yager 2006].

Although classifiers are built so that an overall performance measure is optimized, misclassification rates for different classes may be different, and they may not be in accordance with misclassification costs, since the classes of least interest may be much better classified than the critical ones. This is of particular concern in some real contexts, such as early detection of diseases (since fewer observations of diseased population are often available), risk management and credit card fraud detection, see Carrizosa et al. (2008), He and Yunqian (2013), Prati et al. (2015), Sun et al. (2009) for more details and applications. Consider, as an example, the well-referenced *Breast Cancer Wisconsin (Diagnostic)* data set from the UCI repository (Lichman 2013). It is a slightly unbalanced dataset composed by 30 continuous variables and two classes: *Benign* (63% of the total samples) and *Malignant* (37%). It is relevant to remark that, for this dataset, it is more important to classify correctly the *Malignant* class (the critical one) than the *Benign* class. If the classic NB is performed, setting equal both misclassification costs, then the estimated performance rate for the control group is about 0.96, higher than the rate for the sick group (0.89). One can easily modify the misclassification costs structure, but this way only an indirect control on misclassification rates is obtained.

In this paper we propose a novel way of controlling misclassification rates, that do not call for using misclassification costs which may be hard to choose and are not usually given (Sun et al. 2007, 2009). In particular, a new version of the NB is obtained by modeling performance contraints where the *Recall* (proportion of instances of a given class correctly classified) for the classes of interest is forced to be lower-bounded by certain thresholds. In this way, the user is allowed to assign different importance to the different classes according to her preferences. For example, in the previously considered *Breast Cancer* dataset, it may be desiderable to increase the *Recall* for the *Malignant* class, which was equal to 0.89. As it will be shown in Sect. 3, for this case such rate can be increased up to 0.91. Other example where performance constraints are useful is when fair classification is a requirement as a social criterion, and then the sensitive groups should be protected to avoid the discrimination against race, or other sensitive data (Romei and Ruggieri 2014). Acceptable values for the *Recall* of groups at risk could be fixed via the proposed method in this work. A direct application of our proposal is to handle highly unbalanced datasets, with two or more classes, where the inclusion of performance constrains allows us to improve the results associated with the most damaged classes while controlling the *Recall* related to the rest of the classes.

The problem of cost imbalance has been addressed in the literature from two different perspectives: Data-Level techniques and Algorithm-Level approches, see Leevy et al. (2018). Whereas the former include data sampling methods and feature selection, the latter encompass cost-sensitive and hybrid/ensemble methods which adapt the base classifier to overcome the imbalance. Particularly, our approach can be seen as a cost-sensitive method. Cost-sensitive approaches have been already considered in the literature for well-known classifiers. For example, Datta and Das (2015), Carrizosa et al. (2008) and Lee et al. (2017) focus on the support vector machine (SVM) classifier. In Datta and Das (2015) the decision boundary shift is combined with unequal misclassification penalties. On the other hand, in Carrizosa et al. (2008) a biobjective problem, which simultaneous minimizes the misclassification rates, is performed. In Lee et al. (2017), the authors propose a new weight adjustment factor that is applied to a weighted SVM. In the context of decision trees, Freitas et al. (2007), Ling et al. (2004) introduce tree-building strategies which choose the splitting criterion by minimizing the misclassification costs, whereas Bradford et al. (1998) performs the pruning of a subtree following the cost information. Cost-sensitive versions of neural networks for unbalanced data classification have also been studied in the literature (Cao et al. 2013; Zhou and Liu 2006). Other approaches can be found, for example in Peng et al. (2014), where a new version of the so-called data gravitation-based classification model is proposed.

However, there is a lack of methodologies allowing the user to control the different performance measures of interest at the same time. The application of mathematical optimization tools, the approach that we undertake in this paper, seems to be a promising (Carrizosa and Romero Morales 2013) and not fully explored option: one overall criterion is to be optimized, while constraints are introduced in the model to demand admissible values for the efficiency measures under consideration. Recently, this approach has been considered either in classification (Benítez-Peña et al. 2019; Blanquero et al. 2021) or in regression (Blanquero et al. 2021). In this paper, this technique is explored for improving the NB performance in the classes of most interest to the user. It will be seen that unlike the traditional NB, which is a two-step classifier (estimation first and classification next), the novel approach integrates both stages. In particular, maximum likelihood estimation is formulated as an optimization problem in which thresholds on classification rates are imposed. In other words, maximum likelihood estimates are replaced here by constrained maximum likelihood estimates, where the constraints control the *Recall* values of the classes of interest.

This paper is organized as follows. In Sect. 2 the NB is briefly reviewed and the proposed version of constrained NB (CNB from now on) is described. Section 3 illustrates the usefulness of our novel approach. Eight real databases with different sampling properties are thoroughly analyzed, and a detailed discussion concerning the *Recall* values of the proposed approach compared with the classic NB is given. Some conclusions and further related research are considered in Sect. 4.

## 2 The constrained Naïve Bayes

In our approach, the estimation is performed by solving a constrained maximum likelihood estimation problem, constraints being related with thresholds on the *Recall* values for different classes. The aim of this section is to describe the associated optimization problem. As a result, a computationally tractable classifier that allows the user to control its performance is obtained.

### 2.1 Preliminaries on NB classification

Consider a random vector $(\mathbf{X}, Y)$, where $\mathbf{X} = \left(X_1, \ldots, X_p\right)$ contains $p$ features and $Y$ identifies the class label. Assume that we have a single-label (one class label per observation) classification problem with $K$ classes. Then, for each class $k \in \{1, \ldots, K\}$, let $\pi_k$ denote the prior probability of the class, $\pi_k = P(Y = k)$, and assume that $X_j|(Y = k)$ has a probability density function $f_{\theta_{jk}}(x)$, where $\theta_{jk} \in \Theta_{jk}$. For $k = 1, \ldots, K$, define $\boldsymbol{\theta}_k = (\theta_{1k}, \ldots, \theta_{pk})$.

Let $\mathbf{x} = (x_1, \ldots, x_p)$ be a new observation. Then the aim is to label it on one of the $K$ classes. Then, under the 0–1 loss function, Bayesian Decision Theory establishes that $\mathbf{x}$ is classified in the most probable class according to the conditional distribution. The estimation of the associated parameters may be cumbersome if the number of features $p$ is large. However, the use of the Bayes theorem, in addition to the assumption of independence (conditioned to the class) ease the previous estimation process. As it is well known, the latter assumption implies that the joint density function can be expressed as

$$\begin{aligned} f(x_1, \ldots, x_p, k) &= P(Y = k) f(x_1, \ldots, x_p \mid k) \\ &= \pi_k f_{\boldsymbol{\theta}_k}(\mathbf{x}) \\ &= \pi_k \prod_{j=1}^{p} f_{\theta_{jk}}(x_j), \end{aligned}$$

and thus the estimation process is reduced to estimate the parameters of each marginal distribution. Then, the NB classifier performs by assigning $\mathbf{x}$ to class $k$ satisfying

$$\pi_k \prod_{j=1}^{p} f_{\theta_{jk}}(x_j) \geq \pi_i \prod_{j=1}^{p} f_{\theta_{ji}}(x_j) \quad \forall i = 1, \ldots, K. \tag{1}$$

Given a training sample of size $N_1$, $(\mathbf{x}_1, k_1), \ldots, (\mathbf{x}_{N_1}, k_{N_1})$, then $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ is estimated in NB via maximum likelihood (Hogg et al. 2005), and therefore computed as the solution of the optimization problem:

$$\max_{\boldsymbol{\theta}} \sum_{n=1}^{N_1} \log f_{\boldsymbol{\theta}_{k_n}}(\mathbf{x}_n) \tag{2}$$

Therefore, the classic NB can be seen as a two-step classifier, where the model parameter is first estimated as $\hat{\boldsymbol{\theta}}$ from a training sample, and then (1) is applied under $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$.

### 2.2 A novel formulation with performance constraints

In order to calibrate the performance of a classifier, many measures have been defined in the literature, see Sokolova and Lapalme (2009). In particular, the so-called $Recall_k$, for $k = 1, \ldots, K$, is defined as the sample fraction of individuals in class $k$ which are correctly classified.

Given a validation sample of size $N_2$, where $N_2 = \sum_k N_{2,k}$ and $N_{2,k}$ is the size of class $k$ in such a validation sample, $(\mathbf{x}_1^{(k)}, k), \ldots, (\mathbf{x}_{N_{2,k}}^{(k)}, k)$, then the *Recall* for class $k$ can be expressed as functions of $\hat{\boldsymbol{\theta}}$,

$$Recall_k(\hat{\boldsymbol{\theta}}) = \frac{1}{N_{2,k}} \sum_{n=1}^{N_{2,k}} C_k(\hat{\boldsymbol{\theta}},\ \mathbf{x}_n^{(k)}),\ k = 1, \ldots, K, \tag{3}$$

where

$$C_k(\hat{\boldsymbol{\theta}},\ \mathbf{x}_n^{(k)}) = \begin{cases} 1 & \text{if the individual}\mathbf{x}_n^{(k)}\text{is classified in class } k, \\ \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Unlike the classic NB, based on a two-step approach, the CNB proposed in this paper integrates the performance of the classifier [according to expression (3)] within the estimation step. In particular, the pursued aim is to estimate $\boldsymbol{\theta}$ as the solution of an optimization problem where the objective function is given using a training sample of size $N_1$ as in (2) and, to prevent overfitting, constraints on (3) are imposed on an independent sample (validation set) of size $N_2 = \sum_{k=1}^K N_{2,k}$,

$$\begin{array}{ll} \max\limits_{\boldsymbol{\theta}} & \sum\limits_{n=1}^{N_1} \log f_{\boldsymbol{\theta}_{k_n}}(\mathbf{x}_n) \\ \text{s.t.} & \frac{1}{N_{2,k}} \sum\limits_{n=1}^{N_{2,k}} C_k(\boldsymbol{\theta},\ \mathbf{x}_n^{(k)}) \geq \alpha_k, \quad k = 1, \ldots, K. \end{array} \tag{CNB}$$

In the previous CNB optimization problem, $\alpha_k \in (0, 1)$ is a threshold, a lower-bound value close to 1, for $k = 1, \ldots, K$, which is fixed by the user according to her requirements about the classification in the different classes. From the point of view of optimization, we assume that the function $f_{\boldsymbol{\theta}_{k_n}}$ is smooth with respect to the parameter $\boldsymbol{\theta}_{k_n}$. Regarding the constraints, they are not smooth and therefore, gradient methods cannot be applied in order to solve Problem (CNB). This fact makes the resolution of (CNB) to be slow, especially for large datasets. However, a proxy version of (CNB) can be written in a more tractable way if the constraints are reformulated in terms of smooth functions as

$$\widetilde{C}_k(\boldsymbol{\theta}, \mathbf{x}^{(k)}; \lambda) = \prod_{i=1, i\neq k}^{K} F(y_{ki}(\boldsymbol{\theta}, \mathbf{x}^{(k)}); \lambda), \tag{5}$$

where $F(y; \lambda) = \frac{1}{1+e^{-\lambda y}}$ is the sigmoid function and

$$y_{ki}(\boldsymbol{\theta}, \mathbf{x}) = \pi_k \prod_{j=1}^{p} f_{\theta_{jk}}(x_j) - \pi_i \prod_{j=1}^{p} f_{\theta_{ji}}(x_j). \tag{6}$$

On the one hand, from the definition of the sigmoid function, it can be seen that $\lim_{\lambda\to\infty} \widetilde{C}_k(\boldsymbol{\theta}, \mathbf{x}^{(k)}; \lambda) = C_k(\boldsymbol{\theta}, \mathbf{x}^{(k)})$, since for large values of $\lambda$, $F(y_{ki}(\boldsymbol{\theta}, \mathbf{x}^{(k)}); \lambda)$ will only take the values 0 or 1 depending on the sign of $y_{ki}(\boldsymbol{\theta}, \mathbf{x}^{(k)})$. Then, $\lambda$ is a hyperparameter big enough so that $C$ and $\widetilde{C}$ are as close as possible. On the other hand, the reason why we use the product function to define $\widetilde{C}$ is explained below. Note that if any class $i$ has associated a density much greater than class $k$, then $y_{ki}$ will take a large negative value which makes $F(y_{ki}(\boldsymbol{\theta}, \mathbf{x}^{(k)}); \lambda)$ close to 0 and therefore $\widetilde{C}_k(\boldsymbol{\theta},\ \mathbf{x}^{(k)}; \lambda)$ will also be close to 0. From the previous discussion, a differentiable version of the CNB problem is obtained as

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \sum_{n=1}^{N_1} \log f_{\boldsymbol{\theta}_{k_n}}(\mathbf{x}_n) \\ \text{s.t.} \quad & \frac{1}{N_{2,k}} \sum_{n=1}^{N_{2,k}} \tilde{C}_k(\boldsymbol{\theta},\ \mathbf{x}_n^{(k)}) \geq \alpha_k, \quad k = 1, \ldots, K. \end{aligned} \tag{SCNB}$$

The smooth formulation (SCNB) can be solved using efficient solvers for nonlinear constrained programming [see, e.g. Birgin and Martínez (2008)]. From now on, we refer to (SCNB) as our optimization problem.

Some important remarks need to be made at this point. The first one regards the feasibility of the (SCNB). In a real application, threshold values $\alpha_1, \ldots, \alpha_K$ have to be fixed. As a first option, they could be fixed by the user according to her demand, but it might be the case that (SCNB) is unfeasible. For that reason, we propose a procedure for determining the thresholds in such a way that (SCNB) is always feasible. If we consider a dataset with $K$ different classes, let $\boldsymbol{\theta}^{*}$ be the model parameter associated with (2) and $k_0$ be the critical class or the class where the method performs the worst. Suppose that the aim is to improve the *Recall* for such class $k_0$, say

$$\alpha_{k_0} = \frac{1}{N_{2,k_0}} \sum_{n=1}^{N_{2,k_0}} \tilde{C}_{k_0}(\boldsymbol{\theta}^{*},\ \mathbf{x}_n^{(k_0)}) + \Delta,$$

with $\Delta > 0$. Then, in order to know the maximum threshold $\tau$ for the other classes $k \neq k_0, k \in \{1, \ldots, K\}$, the next optimization problem can be solved:

$$\begin{array}{ll} \max\limits_{\boldsymbol{\theta},\tau} & \tau \\ \text{s.t.} & \dfrac{1}{N_{2,k_0}} \displaystyle\sum_{n=1}^{N_{2,k_0}} \tilde{C}_{k_0}(\boldsymbol{\theta},\ \mathbf{x}_n^{(k_0)}) \geq \dfrac{1}{N_{2,k_0}} \displaystyle\sum_{n=1}^{N_{2,k_0}} \tilde{C}_{k_0}(\boldsymbol{\theta}^*,\ \mathbf{x}_n^{(k_0)}) + \Delta \\ & \dfrac{1}{N_{2,k}} \displaystyle\sum_{n=1}^{N_{2,k}} \tilde{C}_k(\boldsymbol{\theta},\ \mathbf{x}_n^{(k)}) \geq \tau, \quad \forall k \neq k_0. \end{array}$$

This way we search the estimates $\boldsymbol{\theta}$ such that in the relevant class $k_0$ the *Recall* is improved in at least $\Delta$ with respect to the *Recall* in the traditional Bayes estimate and maximize the minimum *Recall* in the remaining classes.

Secondly, it should be highlighted that the parameters $\alpha_1, \ldots, \alpha_K$ involved in the model have a clear interpretation (the desired *Recall* for each of the classes), while allowing us to have full control over all of them. The third comment is related to the size of the considered dataset in terms of the number of predictor variables. Problem (SCNB) can be addressed when the number of features $p$ is large. However, to alleviate the computational cost and thus to improve the running times, we propose to perform a pre-processing to select relevant predictors for large datasets as a part of the procedure. This step will be explained in more detail in Sect. 3.2. Finally, the fourth remark concerns the solutions of (SCNB), which are not maximum likelihood estimates any more, but maximum constrained likelihood estimates instead. On the contrary, the problem yields a solution with the highest sample likelihood fulfilling the constraints on performance on the independent sample. Up to our knowledge, this is a breaking approach that has never been considered in NB models.

## 3 Numerical results

In this section, eight datasets from the UCI Machine Learning Repository and KEEL open source (Alcalá-Fdez et al. 2011, 2009) diverse, in both in the number of classes, sizes and imbalance ratio shall be analyzed. The description of the datasets can be found in Sect. 3.1 and the numerical experiments and obtained results will be considered in Sects. 3.2 and 3.3, respectively.

### 3.1 Datasets

The datasets `breast cancer`, `SPECTF`, `page-blocks`, `abalone`, `yeast`, `Satimage`, `RCV1` and `letter` will be considered. From all the available versions of the datasets, we have chosen those described in Table 1. The colums report the dataset name, the number of instances and features and finally, the class split of the eight considered datasets (`page-blocks`, `abalone`, `yeast`, `Satimage` and `RCV1` can be considered unbalanced datasets).

Table 1 Datasets description

| Name | Intances | Features | Class split (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| breast cancer | 569 | 30 | Benign | | | | Malignant | | | |
| | | | 63 | | | | 37 | | | |
| SPECTF | 267 | 44 | Abnormal | | | | Normal | | | |
| | | | 79 | | | | 21 | | | |
| page-blocks | 5473 | 10 | text | horiz. line | | graphic | vert. line | | picture | |
| | | | 89.8 | 6 | | 0.5 | 1.6 | | 2.1 | |
| abalone | 4177 | 8 | 1-5 | | 6-10 | | 11-15 | | 16-29 | |
| | | | 4.52 | | 28.39 | | 6.25 | | 60.83 | |
| yeast | 1484 | 8 | CYT | EXC | ME1 | ME3 | MIT | NUC | POX | VAC |
| | | | 32.42 | 2.45 | 3.08 | 11.41 | 17.09 | 30.04 | 1.40 | 2.10 |
| Satimage | 6435 | 36 | 1 | 2 | | 3 | 4 | 5 | | 7 |
| | | | 23.82 | 10.92 | | 21.10 | 9.73 | 10.99 | | 23.43 |
| RCV1 | 18758 | 21531 | C15 | CCAT | | E21 | ECAT | GCAT | | M11 |
| | | | 23.70 | 20.12 | | 5.73 | 9.54 | 22.43 | | 5.61 |
| letter | 20000 | 16 | From A to Z (26 classes) | | | | | | | |
| | | | *Equally distributed* | | | | | | | |

## 3.2 Design of experiments

### Probability distributions setting and resolution of the optimization problem

As comented in Sect. 2.1, a probability model needs to be selected for the features conditioned to the class. If the feature is continuous, in this paper we will assume the normal distribution. For discrete features, we consider the categorical distribution, and the Poisson distribution for non-negative integers. From the point of view of the optimization, (SCNB) will be solved using solvers for smooth optimization. In particular, `auglag` and `mma` functions from R package `nloptr` will be used in this work to obtain all numerical results.

### Estimation of the performance rates

The performance of the proposed classifier will be estimated using a stratified 25 Monte-Carlo cross-validation (Xu and Liang 2001). The dataset will be split into three sets, the so-called training, validation and testing sets. One-third of the dataset is used as testing set, and the remaining two-thirds for training set and validation set. Specifically, the training set is formed by two-thirds of those two-thirds of the dataset, whereas the remaining one-third is used for the validation set. As explained in Sect. 2, the objective function will be optimized on the training set while the constraints will be evaluated on the validation set. Once the SCNB problem is solved, *Recall* values are estimated on the testing set. It must be highlighted that at each run, the training sample is built in a stratified way so that the proportion of samples per class is similar to the proportions depicted by Table 1. Finally, regarding the hyperparameter $\lambda$, after an extensive simulation study considering a wide grid of values, the choice $\lambda = 2^3$ is set in the experiments since it provides a good match between $C$ and $\tilde{C}$ as in (4) and (5).

### Pre-processing for large datasets

As commented at the end of Sect. 2.2, Problem (SCNB) turns out computationally costly for large datasets as the considered `RCV1` dataset. As it is common in the literature [see Leevy et al. (2018) and references therein], we suggest to pre-processing such datasets in a way that irrelevant variables are removed in a first step previous to the resolution of (SCNB). That is, at each fold of the stratified 25 Monte-Carlo cross-validation previously commented, the importance of the predictor variables are measured using the training set so that the predictor variables with low importance are not considered when solving Problem (SCNB). Specifically, in this work the importance of the predictor variables composing `RCV1` were measured using the R function `information.gain` from `FSelector`. In this case, most of the variables have an associated importance close to 0 and, then, only 392 of the total are going to be kept when solving (SCNB) for the `RCV1` dataset.

### The choice of thresholds

In order to select the threshold values $\alpha_k$ in Problem (SCNB), the classic NB classifier (2) was first run. Table 2 shows the *Recall* estimates for each class. For `letter` dataset, the average *Recall* values of classic NB are in the first row of Table 4.

Throughout this work we consider the classes where the classic NB performs the worst as the classes of interest or at risk and thus the aim is to improve the rates for such classes. From results in Table 2 and the first row in Table 4, the set of thresholds to be tested in the numerical experiments shall be given by Table 3 and the second row in Table 4. Specifically, the better rates for the classes with the worst associated *Recall* are selected by increasing in steps of two points those results obtained by the classic classifier, whereas admissible values for the rest of classes are also fixed.

Additionally, to highlight the versatility of our proposal, for three of the datasets (`page- blocks`, `yeast` and `letter`) we aim to improve the *Recall* of more than one class at the same time. Thus, for instance, for `yeast` dataset, we will improve the *Recall* of classes `CYT` and `NUC`, which are the two classes in the dataset with the lowest *Recall* values. Then, we first run Problem (SCNB) with thresholds 0.060 for `CYT` and 0.340 for `NUC` and, then, we run it again by imposing 0.080 for `CYT` and 0.360 for `NUC`.

## 3.3 Results

The estimated rates are reported in Tables 5, 6, 7, 8, 9, 10, 11, and 12. The first row shows the results for the classic NB, when no thresholds are imposed. The first column shows the imposed thresholds for the *Recall* of each class, whereas the column and thresholds in bold correspond to the classes at risk (where the classic NB presents the poorest performance). For example, in Table 6, it is required that the *Recall* of `Normal` class is at least 0.900, while over the `Abnormal` class the threshold varies from 0.660 to 0.700. The remaining columns, except for the last one, provide the average *Recall* values measured on the test set. Finally, the last column contains the value of the micro-averaged $F_1$ (Yang and Liu 1999), an aggregate performance measure of the classifier. From the $F_1$ values, the sign-test was used to test if both approaches are statistically significantly different. In particular, the significance codes follow the following nomenclature: '**' , '*' and '.' mean respectively that the $p$-value is smaller than 0.01, 0.05 and 0.1.

As expected, the results under the constrained NB version differ from the results provided by the classic NB. For example, for the `page-blocks` dataset, the *Recall* values under the classic NB are 0.915, 0.673, 0.644, 0.942 and 0.400, for the `text`, `horiz. line`, `graphic`, `vert. line` and `picture` classes, respectively (Table 7). As commented before, we are interested in increasing the *Recall* of the classes worst classified. According to Table 7, if the minima 0.710, 0.680, 0.440 are imposed for the `horiz. line`, `graphic` and `picture` classes, the final rates change from 0.673 to 0.697, from 0.644 to 0.694 and from 0.400 to 0.457, respectively. It is important to highlight two different facts concerning the previous results. First, note that better rates for the `horiz. line`, `graphic` and `picture` classes

**Table 2** Average *Recall* of classic NB (25 Monte-Carlo cross-validation)

| Name | *Recall* classic NB | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| breast cancer | Benign | | | | Malignant | | | |
| | 0.957 | | | | 0.891 | | | |
| SPECTF | Abnormal | | | | Normal | | | |
| | 0.638 | | | | 0.900 | | | |
| page-blocks | text | horiz. line | | graphic | vert. line | | picture | |
| | 0.915 | 0.673 | | 0.644 | 0.942 | | 0.400 | |
| abalone | 1-5 | | 6-10 | | 11-15 | | 16-29 | |
| | 0.927 | | 0.584 | | 0.244 | | 0.536 | |
| yeast | CYT | EXC | ME1 | ME3 | MIT | NUC | POX | VAC |
| | 0.038 | 0.684 | 0.703 | 0.460 | 0.405 | 0.317 | 0.513 | 0.391 |
| Satimage | 1 | 2 | | 3 | 4 | 5 | | 7 |
| | 0.794 | 0.899 | | 0.893 | 0.659 | 0.736 | | 0.750 |
| RCV1 | C15 | CCAT | | E21 | ECAT | GCAT | | M11 |
| | 0.874 | 0.061 | | 0.766 | 0.209 | 0.753 | | 0.888 |

**Table 3** Tested thresholds

| Name | *Recall* classic NB | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| breast cancer | Benign | | | | Malignant | | | |
| | 0.950 | | | | 0.910/0.930/0.950 | | | |
| SPECTF | Abnormal | | | | Normal | | | |
| | 0.660/0.680/0.700 | | | | 0.900 | | | |
| page-blocks | text | horiz. line | | graphic | vert. line | | picture | |
| | 0.910 | 0.690/0.710 | | 0.660/0.680 | 0.940 | | 0.420/0.440 | |
| abalone | 1-5 | | 6-10 | | 11-15 | | 16-29 | |
| | 0.920 | | 0.580 | | 0.260/0.280/0.300 | | 0.530 | |
| yeast | CYT | EXC | ME1 | ME3 | MIT | NUC | POX | VAC |
| | 0.060/0.080 | 0.680 | 0.700 | 0.460 | 0.400 | 0.340/0.360 | 0.510 | 0.390 |
| Satimage | 1 | 2 | | 3 | 4 | 5 | | 7 |
| | 0.790 | 0.890 | | 0.890 | 0.680/0.700 | 0.730 | | 0.750 |
| RCV1 | C15 | CCAT | | E21 | ECAT | GCAT | | M11 |
| | 0.870 | 0.080 | | 0.760 | 0.200 | 0.750 | | 0.880 |

**Table 4** Average *Recall* of classic NB (25 Monte-Carlo cross-validation) and tested thresholds for letter dataset

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Average *Recall* classic NB | 0.882 | 0.662 | 0.777 | 0.648 | 0.364 | 0.695 | 0.484 | 0.318 |
| Tested thresholds | 0.880 | 0.660 | 0.770 | 0.640 | 0.380 | 0.690 | 0.480 | 0.330 |
| | I | J | K | L | M | N | O | P |
| Average *Recall* classic NB | 0.785 | 0.659 | 0.437 | 0.741 | 0.847 | 0.703 | 0.721 | 0.739 |
| Tested thresholds | 0.780 | 0.650 | 0.430 | 0.740 | 0.840 | 0.700 | 0.720 | 0.730 |
| | Q | R | S | T | U | V | W | X |
| Average *Recall* classic NB | 0.510 | 0.606 | 0.231 | 0.731 | 0.726 | 0.746 | 0.788 | 0.437 |
| Tested thresholds | 0.510 | 0.600 | 0.250 | 0.730 | 0.720 | 0.740 | 0.780 | 0.430 |
| | Y | Z | | | | | | |
| Average *Recall* classic NB | 0.325 | 0.585 | | | | | | |
| Tested thresholds | 0.340 | 0.580 | | | | | | |

**Table 5** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for breast cancer

| Thresholds (Benign/Malignant) | *Recall* Benign | *Recall* Malignant | micro $F_1$ |
|---|---|---|---|
| *Classic NB* | 0.957 | **0.891** | 0.933 |
| 0.950/**0.910** | 0.951 | **0.898** | 0.932 |
| 0.950/**0.930** | 0.947 | **0.903** | 0.931 |
| 0.950/**0.950** | 0.939 | **0.906** | 0.926 |

Sign test. Signif. codes: '**', '*' and '.' mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

**Table 6** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for SPECTF

| Thresholds (Abnormal/Normal) | *Recall* Abnormal | *Recall* Normal | micro $F_1$ |
|---|---|---|---|
| Classic NB | **0.638** | 0.900 | 0.690 |
| **0.660**/0.900 | **0.660** | 0.854 | $0.698^{\cdot}$ |
| **0.680**/0.900 | **0.669** | 0.858 | $0.707^{*}$ |
| **0.700**/0.900 | **0.671** | 0.850 | $0.706^{**}$ |

Sign test. Signif. codes: '**', '*' and '.' mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

have been obtained, but at the expense of slightly decreasing the rates of the rest of the classes. Second, note that even though a rate equal to 0.710 was imposed for the horiz. line class, such value was not finally obtained, but a slightly smaller one (0.697) instead. This is not surprising, since the constraints are imposed for one sample, and tested on an independent set.

From the results shown in Tables 5, 6, 7, 8, 9, 10, 11, and 12, it can be concluded that the proposed approach allows the user to control the *Recall* values in such a way that

**Table 7** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for `page-blocks`

| *Thresholds* (`text`/`horiz. line`/<br>`graphic`/`vert. line`/`picture`) | *Recall* `text` | *Recall* `horiz. line` | *Recall* `graphic` | *Recall* `vert. line` | *Recall* `picture` | micro $F_1$ |
|---|---|---|---|---|---|---|
| *Classic NB* | 0.915 | **0.673** | **0.644** | 0.942 | **0.400** | 0.889 |
| 0.910/**0.690**/**0.660**/0.940/**0.420** | 0.915 | **0.737** | **0.665** | 0.940 | **0.434** | 0.893 |
| 0.910/**0.710**/**0.680**/0.940/**0.440** | 0.902 | **0.697** | **0.694** | 0.934 | **0.457** | 0.880 |

Sign test. Signif. codes: ‘**’, ‘*’ and ‘.’ mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

**Table 8** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for `abalone`

| *Thresholds* (`1-5/6-10/11-15/16-29`) | *Recall* `1-5` | *Recall* `6-10` | *Recall* `11-15` | *Recall* `16-29` | micro $F_1$ |
|---|---|---|---|---|---|
| *Classic NB* | 0.927 | 0.584 | **0.244** | 0.536 | 0.549 |
| 0.920/0.580/**0.260**/0.530 | 0.926 | 0.590 | **0.252** | 0.540 | 0.553 |
| 0.920/0.580/**0.280**/0.530 | 0.926 | 0.584 | **0.261** | 0.537 | 0.550 |
| 0.920/0.580/**0.300**/0.530 | 0.928 | 0.579 | **0.271** | 0.538 | 0.550 |

Sign test. Signif. codes: ‘**’, ‘*’ and ‘.’ mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

**Table 9** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for `yeast`

| *Thresholds* (CYT/EXC/ME1/ME3/<br>MIT/NUC/POX/VAC) | CYT | EXC | ME1 | ME3 | MIT | NUC | POX | VAC | micro $F_1$ |
|---|---|---|---|---|---|---|---|---|---|
| *Classic NB* | **0.038** | 0.684 | 0.703 | 0.460 | 0.405 | **0.317** | 0.513 | 0.391 | 0.281 |
| **0.060**/0.680/0.700/0.460/0.400/**0.340**/0.510/0.390 | **0.066** | 0.567 | 0.749 | 0.573 | 0.447 | **0.426** | 0.513 | 0.427 | 0.343** |
| **0.080**/0.680/0.700/0.460/0.400/**0.360**/0.510/0.390 | **0.079** | 0.578 | 0.763 | 0.611 | 0.433 | **0.430** | 0.513 | 0.378 | 0.350** |

Sign test. Signif. codes: ‘**’, ‘*’ and ‘.’ mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

**Table 10** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for `Satimage`

| Thresholds (1/2/3/4/5/7) | *Recall* 1 | *Recall* 2 | *Recall* 3 | *Recall* 4 | *Recall* 5 | *Recall* 7 | micro $F_1$ |
|---|---|---|---|---|---|---|---|
| *Classic NB* | 0.794 | 0.899 | 0.893 | **0.659** | 0.736 | 0.750 | 0.797 |
| 0.790/0.890/0.890/**0.680**/0.730/0.750 | 0.794 | 0.901 | 0.898 | **0.662** | 0.738 | 0.742 | 0.797 |
| 0.790/0.890/0.890/**0.700**/0.730/0.750 | 0.796 | 0.903 | 0.903 | **0.671** | 0.739 | 0.742 | 0.800 · |

Sign test. Signif. codes: ‘**’, ‘*’ and ‘.’ mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

**Table 11** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for RCV1 using 392 variables of the total

| *Thresholds* (C15/CCAT/ E21/ECAT/GCAT/M11) | *Recall* C15 | *Recall* CCAT | *Recall* E21 | *Recall* ECAT | *Recall* GCAT | *Recall* M11 | micro $F_1$ |
|---|---|---|---|---|---|---|---|
| *Classic NB* | 0.874 | **0.061** | 0.766 | 0.209 | 0.753 | 0.888 | 0.576 |
| 0.870/**0.080**/0.760/0.200/0.750/0.880 | 0.872 | **0.091** | 0.757 | 0.260 | 0.775 | 0.887 | 0.593 · |

Sign test. Signif. codes: ‘**’, ‘*’ and ‘.’ mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

**Table 12** Average *Recall* values of SCNB (25 Monte-Carlo cross-validation) for `letter`

| Thresholds | *Recall* A | *Recall* B | *Recall* C | *Recall* D | *Recall* E | *Recall* F | *Recall* G | *Recall* H |
|---|---|---|---|---|---|---|---|---|
| Classic NB | 0.882 | 0.662 | 0.777 | 0.648 | **0.364** | 0.695 | 0.484 | **0.318** |
| **0.380/0.330/0.250/0.340** | 0.884 | 0.671 | 0.777 | 0.653 | **0.388** | 0.720 | 0.490 | **0.342** |
| | *Recall* I | *Recall* J | *Recall* K | *Recall* L | *Recall* M | *Recall* N | *Recall* O | *Recall* P |
| Classic NB | 0.785 | 0.659 | 0.437 | 0.741 | 0.847 | 0.703 | 0.721 | 0.739 |
| **0.380/0.330/0.250/0.340** | 0.780 | 0.665 | 0.445 | 0.746 | 0.858 | 0.722 | 0.715 | 0.744 |
| | *Recall* Q | *Recall* R | *Recall* S | *Recall* T | *Recall* U | *Recall* V | *Recall* W | *Recall* X |
| Classic NB | 0.510 | 0.606 | **0.231** | 0.731 | 0.726 | 0.746 | 0.788 | 0.437 |
| **0.380/0.330/0.250/0.340** | 0.517 | 0.604 | **0.251** | 0.726 | 0.736 | 0.749 | 0.806 | 0.475 |
| | *Recall* Y | *Recall* Z | micro $F_1$ | | | | | |
| Classic NB | **0.325** | 0.585 | 0.622 | | | | | |
| **0.380/0.330/0.250/0.340** | **0.346** | 0.602 | 0.633 * | | | | | |

The first column only shows the imposed thresholds for the classes at risk, the admissible values for the rest of classes are given in Table 4. Sign test. Signif. codes: '**', '*' and '.' mean that a $p$-value smaller than 0.01, 0.05 and 0.1 is obtained

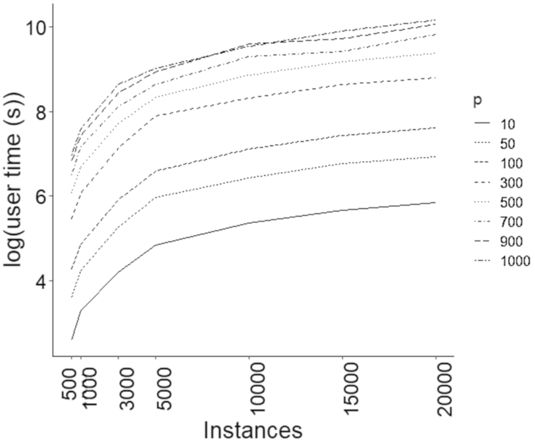

**Fig. 1** Scalability: X-axis represents the number of instances (with range from 500 to 20,000) whereas each line the number of features (with range from 10 to 1000)

the classes of interest, where the classic method performs the worst in this case, can be improved. Additionally, our approach reaches comparable or even better overall results than the classic NB [see micro $F_1$ scores throughout Tables 5, 6, 7, 8, 9, 10, 11, and 12]. Note that among the possible non-dominated solutions shown for each dataset, the user could choose according to her interest and to what she is willing to lose in the less critical classes.

Finally, to illustrate the computational cost of the optimization algorithm depending on the number of instances and features, we simulated data following (Witten et al. 2014) with {500, 1000, 3000, 5000, 10,000, 15,000, 20,000} instances and $p \in \{10, 50, 100, 300, 500, 700, 900, 1000\}$. Figures 1 and 2 report the logarithm of the user times (in seconds) when the SCNB is run on an Intel(R) Core(TM) i7-7500U CPU at 2.70 GHz 2.90 GHz with 8.0 GB of RAM, and the number of evaluations for the algorithm `auglag` is 100. The X-axis of Fig. 1 shows the number of instances whereas each line represents the number of variables of the dataset ($p$). Figure 2 is the opposite. Overall, running time grows linearly respect to the number of instances, but not so smooth when $p$ increases.

## 4 Conclusions and extensions

In this paper a new version of the NB classifier is proposed with the aim of controlling misclassification rates in the different classes, avoiding the use of precise values of misclassification costs, which may be hard to choose. In order to achieve this goal, performance constraints are included into the optimization problem which estimates the involved parameters. The approach results in a novel method (SCNB) not reported in the literature previously, up to our knowledge. Unlike the classic NB, which is based on a two-step approach, the (SCNB) integrates the performance rates in the parameters' estimation step. In fact, this novel approach allows the user to impose thresholds to assure the achievement in the measures of efficiency (in this case, the *Recall* values). The proposed methodology has been tested on eight real datasets with

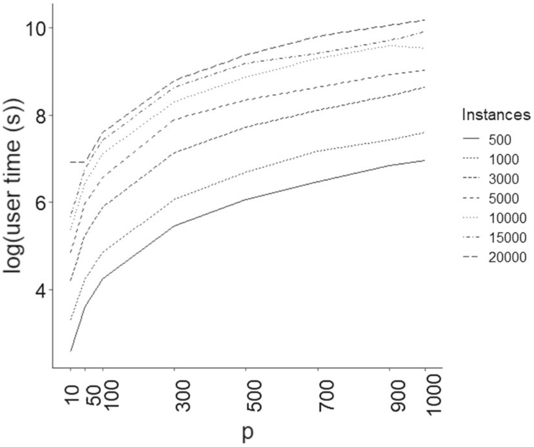

**Fig. 2** Scalability: X-axis represents the number of features (with range from 10 to 1000) whereas each line the number of instances (with range from 500 to 20,000)

different sampling properties. The numerical results show that not only the classification rates of interest can be controlled and improved, but also similar or even better overall results, comparing with those of the classic NB, are obtained. The former is of great interest in some medical, credit scoring or social contexts where some classes are more critical than others.

A possible extension to this work is to consider non parametric estimation for the density function for continuous attributes via kernel density estimation. Also, one anonymous referee suggested to measure the efficiency of the approach via statistical tests in the same spirit as in Demšar (2006). Work of these issues is underway.

**Acknowledgements** This research is partially supported by research grants and projects MTM2015-65915-R (Ministerio de Economía y Competitividad, Spain) and PID2019-110886RB-I00 (Ministerio de Ciencia, Innovación y Universidades, Spain), FQM-329 and P18-FR-2369 (Junta de Andalucía, Spain), PR2019-029 (Universidad de Cádiz, Spain) and EC H2020 MSCA RISE NeEDS Project (Grant Agreement ID: 822214). This support is gratefully acknowledged.

**Funding** Open Access funding provided by the IReL Consortium.

## References

Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. J Mult-Valued Logic Soft Comput 17:255–287

Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. Soft Computing 13(3):307–318

Benítez-Peña S, Blanquero R, Carrizosa E, Ramírez-Cobo P (2019) On support vector machines under a multiple-cost scenario. Advances in Data Analysis and Classification 13(3):663–682

Bermejo P, Gámez JA, Puerta JM (2011) Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. Expert Systems with Applications 38(3):2072–2080

Birgin E, Martínez J (2008) Improving ultimate convergence of an augmented Llagrangian method. Optim Methods Softw 23(2):177–195

Blanquero R, Carrizosa E, Molero-Río C, Romero Morales D (2021) Optimal randomized classification trees. Computers & Operations Research 132:105281

Blanquero R, Carrizosa E, Ramírez-Cobo P, Sillero-Denamiel MR (2021) A cost-sensitive constrained lasso. Advances in Data Analysis and Classification 15:121–158

Boullé M (2007) Compression-based Averaging of Selective Naive Bayes Classifiers. Journal of Machine Learning Research 8:1659–1685

Bradford JP, Kunz C, Kohavi R, Brunk C, Brodley CE (1998) Pruning decision trees with misclassification costs. In: Nédellec C, Rouveirol C (eds) Machine learning: ECML-98. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 131–136

Cao P, Zhao D, Zaïane OR (2013) A PSO-based cost-sensitive neural network for imbalanced data classification. In: Li J, Cao L, Wang C, Tan KC, Liu B, Pei J, Tseng VS (eds) Trends and applications in knowledge discovery and data mining. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 452–463

Carrizosa E, Martín-Barragán B, Romero Morales D (2008) Multi-group support vector machines with measurement costs: A biobjective approach. Discrete Applied Mathematics 156:950–966

Carrizosa E, Romero Morales D (2013) Supervised classification and mathematical optimization. Computers and Operations Research 40(1):150–165

Chandra B, Gupta M (2011) Robust approach for estimating probabilities in Naïve-Bayes classifier for gene expression data. Expert Systems with Applications 38(3):1293–1298

Datta S, Das S (2015) Near–Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs. Neural Netw 70:39–52

Demšar J (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7:1–30

Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29(2–3):103–130

Freitas A, Costa-Pereira A, Brazdil P (2007) Cost-sensitive decision trees applied to medical data. In: Song IY, Eder J, Nguyen TM (eds) Data Warehousing and Knowledge Discovery. Springer, Berlin Heidelberg, pp 303–312

Guan G, Guo J, Wang H (2014) Varying Naïve Bayes Models With Applications to Classification of Chinese Text Documents. Journal of Business & Economic Statistics 32(3):445–456

Hand DJ, Yu K (2001) Idiot's Bayes - Not So Stupid After All? International Statistical Review 69(3):385–398

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer, NY

He H, Yunqian M (2013) Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley, Hoboken

Hogg RV, McKean J, Craig AT (2005) Introduction to Mathematical Statistics. Pearson Education

Jiang L, Wang S, Li C, Zhang L (2016) Structure extended multinomial naive Bayes. Information Sciences 329(Supplement C):346–356

Lee W, Jun CH, Lee JS (2017) Instance categorization by support vector machines to adjust weights in adaboost for imbalanced data classification. Information Sciences 381(Supplement C):92–103

Leevy JL, Khoshgoftaar TM, Bauder RA, Seliya N (2018) A survey on addressing high-class imbalance in big data. J Big Data. https://doi.org/10.1186/s40537-018-0151-6

Lichman, M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml

Ling CX, Yang Q, Wang J, Zhang S (2004) Decision trees with minimal costs. In: Proceedings of the twenty-first international conference on machine learning, ICML '04, p. 69. New York, NY, USA

Mehra N, Gupta S (2013) Survey on multiclass classification methods. International Journal of Computer Science and Information Technologies 4(4):572–576

Menzies T, Greenwald J, Frank A (2007) Data Mining Static Code Attributes to Learn Defect Predictors. IEEE Transactions on Software Engineering 33(1):2–13

Minnier J, Yuan M, Liu JS, Cai T (2015) Risk Classification With an Adaptive Naive Bayes Kernel Machine Model. Journal of the American Statistical Association 110(509):393–404

Parthiban G, Rajesh A, Srivatsa SK (2011) Diagnosis of Heart Disease for Diabetic Patients using Naive Bayes Method. International Journal of Computer Applications 24(3):0975–8887

Peng L, Zhang H, Yang B, Chen Y (2014) A new approach for imbalanced data classification based on data gravitation. Inf Sci 288(Supplement C):347–373

Prati RC, Batista GE, Silva DF (2015) Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. Knowledge and Information Systems 45:247–270

Romei A, Ruggieri S (2014) A multidisciplinary survey on discrimination analysis. The Knowledge Engineering Review 29(5):582–638

Rosen GL, Reichenberger ER, Rosenfeld AM (2010) NBC: the Naïve Bayes Classification tool webserver for taxonomic classification of metagenomic reads. Bioinformatics 27(1):127–129

Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Information Processing & Management 45(4):427–437

Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40(12):3358–3378

Sun Y, Wong AK, Kamel MS (2009) Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23:687–719

Turhan B, Bener A (2009) Analysis of Naive Bayes' assumptions on software fault data: An empirical study. Data & Knowledge Engineering 68(2):278–290

Wei W, Visweswaran S, Cooper GF (2011) The application of naive Bayes model averaging to predict Alzheimer's disease from genome-wide data. Journal of the American Medical Informatics Association 18(4):370–375

Witten DM, Shojaie A, Zhang F (2014) The Cluster Elastic Net for High-Dimensional Regression With Unknown Variable Grouping. Technometrics 56(1):112–122

Wolfson J, Bandyopadhyay S, Elidrisi M, Vazquez-Benitez G, Vock DM, Musgrove D, Adomavicius G, Johnson PE, O'Connor PJ (2015) A Naive Bayes machine learning approach to risk prediction using censored, time-to-event data. Statistics in Medicine 34(21):2941–2957

Wu J, Pan S, Zhu X, Cai Z, Zhang P, Zhang C (2015) Self-adaptive attribute weighting for Naive Bayes classification. Expert Systems with Applications 42(3):1487–1502

Xu QS, Liang YZ (2001) Monte Carlo cross validation. Chemom Intell Lab Syst 56(1):1–11

Yager RR (2006) An extension of the naive Bayesian classifier. Information Sciences 176(5):577–588

Yang Y, Liu X (1999). A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval (SIGIR), pp. 42–49. New York, NY, USA

Zhou Zhi-Hua, Liu Xu-Ying (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng 18(1):63–77

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.