Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

A mathematical programming approach to SVM-based classification with label noise[☆]

Víctor Blanco[a,b,*], Alberto Japón[c,d], Justo Puerto[c,d]

[a] Institute of Mathematics (IMAG), Universidad de Granada, Spain
[b] Dpt. Quantitative Methods for Economics & Business, Universidad de Granada, Spain
[c] Institute of Mathematics (IMUS), Universidad de Sevilla, Spain
[d] Dpt. Stats & OR, Universidad de Sevilla, Spain

ARTICLE INFO

Keywords:
Supervised classification
SVM
Mixed integer non linear programming
Label noise

ABSTRACT

In this paper we propose novel methodologies to optimally construct Support Vector Machine-based classifiers that take into account that label noise occur in the training sample. We propose different alternatives based on solving Mixed Integer Linear and Non Linear models by incorporating decisions on relabeling some of the observations in the training dataset. The first method incorporates relabeling directly in the SVM model while a second family of methods combines clustering with classification at the same time, giving rise to a model that applies simultaneously similarity measures and SVM. Extensive computational experiments are reported based on a battery of standard datasets taken from UCI Machine Learning repository, showing the effectiveness of the proposed approaches.

1. Introduction

The primary goal of supervised classification is to find patterns from a training sample of labeled data in order to predict the labels of out-of-sample data, in case the possible number of labels is finite. Among the most relevant applications of classification methods are those related with security, as in spam filtering or intrusion detection. The main difference of these applications with respect to other uses of classification approaches is that malicious adversaries can adaptively manipulate their data to mislead the outcome of an automatic analysis. For instance, spammers often modify their emails by obfuscating words which typically appear in known spam or by adding words which are likely to appear in legitimate emails. Also, as stated in Weerasinghe, Erfani, Alpcan, and Leckie (2019), when machine learning algorithms utilized in safety–critical environments are compromised by adversaries, it could even result in loss of human lives. Note that, doubting on the reliability of the labels on the target variable is usual when having suspicions about the possibility of an intentional flip among these labels. However, it is not by far the only case in which one must think about this possibility. Nowadays, it is commonly said that data scientists spend a large percentage of their time dealing with collecting and preprocessing data, meanwhile the remainder is used to model and extract information from databases. Mistakes converted into wrong label assignments are very likely to happen. For instance, data can be wrongly identified at the very beginning of the data collection phase, or code errors can occur when preprocessing a database, leading to a dataset with label noise. Then, one has to, not only derive a classification rule from a training sample, able to adequately classify out-of-sample data, but also to take into account that some of the labels might be incorrect.

The goal of this paper is to analyze the power of using mathematical programming tools for the label noise detection when constructing a Support Vector Machine (SVM) classifier. As pointed out in Ganapathiraju and Picone (2000), among all the available optimization-based classifiers, SVMs particularly suffer the effect of noisy labels because their reliance on support vectors and the feature interdependence assumption. This is the reason to analyze only this baseline model here. Although it would have been possible to extend the analysis to other methodologies (for instance, to classification trees), it would require

[☆] The authors of this research acknowledge financial support by the Spanish Ministerio de Ciencia y Tecnologia, Agencia Estatal de Investigacion and Fondos Europeos de Desarrollo Regional (FEDER) via project PID2020114594GB-C21. The authors also acknowledge partial support from projects FEDER-US-1256951, Junta de Andalucía P18-FR-1422, CEI-3-FQM331, NetmeetData: Ayudas Fundación BBVA a equipos de investigación científica 2019. The first author was also supported by projects P18-FR-2369 (Junta de Andalucía) and IMAG-Maria de Maeztu grant CEX2020-001105-M /AEI /10.13039/501100011033. (Spanish Ministerio de Ciencia y Tecnologia).

[*] Corresponding author.
E-mail addresses: vblanco@ugr.es (V. Blanco), ajapon1@us.es (A. Japón), puerto@us.es (J. Puerto).

https://doi.org/10.1016/j.cie.2022.108611
Received 7 September 2021; Received in revised form 10 May 2022; Accepted 23 August 2022
Available online 30 August 2022
0360-8352/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

to include extra mathematical programming formulations as well as further considerations on the proposed models which would loose the focus of our contribution and would decrease the readability of the paper. Needless to say, that a similar approach can be followed with other baseline methods but this is beyond the scope of this paper. The interested reader is referred to Blanco, Japón, and Puerto (2020b) for further details of this methodology applied to the construction of classification trees.

Related works

Analyzing the vulnerabilities of classifiers and their robustness against attacks, to better understand how their security may be improved, has recently received growing interest from the scientific community. Bi and Zhang propose in Bi and Zhang (2005) robust alternatives when the features of the training sample observations are corrupted. On the other hand, Biggio et al. provide in Biggio, Nelson, and Laskov (2011) an algorithmic approach to handle adversarial modifications of the labels, in case the labels are independently flipped with the same probability, by correcting the kernel matrix. According to Nalepa and Kawulok (2018), three main groups of approaches for dealing with noisy datasets have been already proposed in the literature: (1) Design of algorithms which filter noisy and/or mislabeled vectors from the input data (as in Ekambaram et al., 2016; Ghoggali & Melgani, 2009; Han & Chang, 2013); (2) Construction of robust classifiers against noisy labeling (see Duan and Wu (2018) and Natarajan, Dhillon, Ravikumar, and Tewari (2017)); and (3) Use of noise models in parallel with the obtention of the classifier, which are finally coupled for a higher-quality classification (see Bertsimas, Dunn, Pawlowski, and Zhuo (2019), Ganapathiraju and Picone (2000), Weerasinghe et al. (2019), Xiao et al. (2015) and Xu, Crammer, and Schuurmans (2006)). Further details on the different approaches to deal with datasets containing mislabeled observations can be found in the recent survey in Frénay and Verleysen (2013).

Most recent methodologies to deal with noisy datasets are sequential. Thus, loosing the optimal performance obtained by one shot methods based on mathematical programming approaches. For instance, in the recent method presented in Northcutt, Jiang, and Chuang (2021), based in the so-called Support Vector Machine with *Confident Learning (SVM-CL) approach,* the authors propose a probabilistic method in three sequential phases: (1) estimate the transition matrix of class-conditional label noise, (2) filter out noisy examples, and (3) train the dataset once noisy data are removed via Co-Teaching. Analogously, in de França and Coelho (2015) it is proposed a novel method in which first the training sample is biclustered (see e.g., Cheng & Church, 2000) trying to capture correlation between features and observations, next the training sample is modified according to the biclusters, and then the classification is performed on the modified dataset. Furthermore, there are some *globally optimal* methods that have been proposed in the literature. In particular, in Bertsimas et al. (2019), the authors present different robust adaptations of classical classification methods to deal with uncertainty in labels and/or features in the training sample.

In contrast to those methods that have been already proposed to deal with classification and noisy labels, our approach simultaneously construct a SVM-based classifier and re-labels observations, leading to an optimal method. In addition, this approach allows one to get separating hyperplanes that would have been impossible to obtain throughout standard SVM and that report better results for many different problems.

Although the method proposed in Bertsimas et al. (2019) also optimally constructs the classifier under the presence of noisy labels, it is thought to be robust against the worse possible situation. On the contrary, our method builds the classifier always on the convenience of finding good classifiers and not to be protected against the worst possible flip of labels which results in better classifiers in most scenarios.

Moving away from the main focus of our paper, one can also find in the literature different techniques to handle data with noisy labels, as for instance, deep-learning classification models (see e.g., Chen, Liao, Chen, and Zhang (2019), Liu, Niles-Weed, Razavian and Fernandez-Granda (2020), Tanaka, Ikami, Yamasaki, and Aizawa (2018) and Yu et al. (2019)) or Classification Trees (Blanco et al., 2020b).

Our contribution

In this paper, we propose a novel mathematical programming based methodology to construct an optimal classification rule by means of an *ad hoc* adaptation of a Support Vector Machine (SVM) classifier that incorporates the detection and correction of label noise in the dataset. Support Vector Machine (SVM) is a widely-used methodology in supervised binary classification, firstly proposed in Cortes and Vapnik (1995). Given a number of observations with their corresponding labels, the SVM technique consists, in its simplest form, of finding an hyperplane in the feature space so that each class belongs to a different half-space maximizing the separation between classes (in a training sample) and minimizing some measure of the misclassifying errors. This problem can be cast within the class of convex optimization and its dual has very good properties that allow one to extend the methodology to construct also nonlinear classifiers. Most of the SVM literature concentrates on binary classification where several extensions are available. One can use different measures for the separation between classes (Blanco, Puerto and Rodríguez-Chía, 2020; Ikeda & Murata, 2005a, 2005b), aggregation strategies (Maldonado, Merigó, & Miranda, 2018), select important features (Labbé, Martínez-Merino, & Rodríguez-Chía, 2018), apply regularization strategies (López, Maldonado, & Carrasco, 2018; Peng, Xu, Kong, & Chen, 2016), use twin (non parallel) separators (Peng & Chen, 2018), extensions to multiclass classification (Blanco, Japón, & Puerto, 2020a; Liu, Martín-Barragán, & Prieto, 2021), one-class classification (Kang, Kim, & Cho, 2019; Shin, Eom, & Kim, 2005) and control charts pattern recognition (Ünlü, 2021), incorporation of margin distributions (Liu, Chu, Gong and Peng, 2020), or extensions of the hyperplane location problem to other supervised learning problems (Blanco, Japón, Ponce, & Puerto, 2021; Blanco, Puerto, & Salmerón, 2018), etc.

One of the main reasons of the success of SVM tools in classification, may be that one can project the original data onto a higher dimensional space where the separation of the classes can be more adequately performed, and still with the same computational effort that was required in the original problem. This property is the so-called *kernel trick*, and very likely this is one of the reasons that has motivated the successful use of this tool in a wide range of applications (see e.g., Bahlmann, Haasdonk, and Burkhardt (2002), Kašćelan, Kašćelan, and Novović Burić (2016), Majid, Ali, Iqbal, and Kausar (2014), Okwuashi and Ndehedehe (2020) and Radhimeenakshi (2016), among many others).

The construction of SVM-based classifiers that simultaneously relabel observations has many advantages when dealing with label noise datasets, but also when working on problems in which false positives and false negatives have different misclassifying costs. Also, in problems with unbalanced classes (as for instance in datasets on fraud with credit card transactions in which around a 99.9% of the observations are not fraudulent transactions Federal Trade Commission, 2017; Maldonado, Bravo, López, & Pérez, 2017 or in the number of claims in non-life insurances Boucher, Denuit, & Guillen, 2009). In Fig. 1 we illustrate this situation. One can observe in the left picture the projection on the plane of a set of observations labeled by fraudulent (red) and non fraudulent (green) transactions. Linear separators seems to be impossible to construct for this instance, but also non linear classifiers will result in overfitting. However, as shown in the right picture, if one allows a few of the labels to be changed, one can obtain better classifiers. Note that in this case, false positives are more costly than false negatives (since asking for a little more of information via

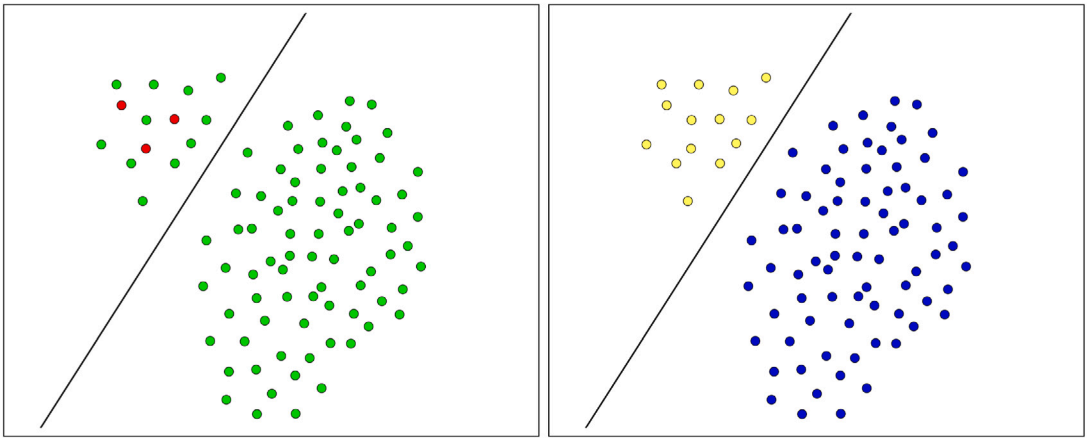

Fig. 1. Original data (left) and optimal hyperplane separating re-labeled classes with our method (right).

text message on the phone normally solves this true negative cases). It is also important to remark that this separating hyperplane could not have been obtained through standard SVM since all the support vectors belong to the same class (green points).

In this paper we propose two different approaches. We present a model in which re-labeling observations depends on the errors of the SVM-based method itself searching for a compromise between the gain obtained in misclassification error and margin and the penalty paid for each change of labels. On the other hand, we will also introduce two models in which re-labeled observations will come from similarity measures on the data. Our method is distribution-free so that it does not assume any distribution on the dataset and the detection of mislabeled observations and the construction of the classifier is optimal based on solving an add-hoc mathematical program.

To assess the validity of these methods we have performed a battery of computational experiments on 7 different real datasets. For these datasets we have repeated the experiments for 5 different scenarios, by randomly flipping a 0%, 20%, 30%, 40% or 50% of the labels in the original data. When comparing our method with respect to classical SVM, and with SVM-CL from Northcutt et al. (2021), we can see that ours gets better results on noisy label datasets.

Summarizing, the main contributions of this work are the following:

- We provide different mathematical programming models to construct classification rules from a training sample by deciding, simultaneously, those observations with label noise.
- The mathematical programming formulations are based on adapting adequately different Support Vector Machine models to integrate them the possibility of relabeling observations with two different families of approaches: One based on relabeling by means of minimizing misclassification errors and other based on different unsupervised learning tools.
- The proposed approaches do not assume any distribution on the dataset and the detection of mislabeled observations and the construction of the classifier is optimal based on solving an add-hoc optimization program.
- The results of testing our algorithms on different real-world datasets indicate that our proposals are more robust under attacks than the classical SVM classifier but also than the recent SVM-CL (Northcutt et al., 2021) approach which is specifically taylored to this end.

The rest of the paper is organized as follows. In Section 2 we set up and describe the elements of the problem to be considered. Afterward, in Section 3 we introduce the different formulations of our models, to end up in Section 4 presenting our computational experiments. Finally, we finish this article in Section 5 with some conclusions and an outline of our future work.

2. Preliminaries

In this section we introduce the problem under study and set the notation used through this paper.

Given a training sample $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq \mathbb{R}^p \times \{+1, -1\}$, the goal of linear SVM (see e.g., Cortes and Vapnik (1995) and Mangasarian (1999)) is to obtain a hyperplane separating the data ($x \in \mathbb{R}^p$) into their two different classes ($y \in \{+1, -1\}$). Among all possible hyperplanes that can obtain such a separation between the classes, SVM looks for the one with maximum margin (maximum distance from classes to the separating hyperplane) while minimizing the misclassification errors. Let us denote by $\mathcal{H}$ a hyperplane in $\mathbb{R}^p$ in the form $\mathcal{H} = \{z \in \mathbb{R}^p \ : \ \omega^t z + \omega_0 = 0\}$ for some $\omega \in \mathbb{R}^p$ and $\omega_0 \in \mathbb{R}$ (the vector v^t is the result of the transpose operator applied to the vector $v \in \mathbb{R}^p$). This hyperplane will induce a subdivision of the data space $\mathbb{R}^p$ into three regions: the $+1$ (positive) half-space $\mathcal{H}^+ = \{z \ : \ \omega^t z + \omega_0 > 1\}$, the -1 (negative) half-space $\mathcal{H}^- = \{z \ : \ \omega^t z + \omega_0 < -1\}$ and the strip $\mathcal{S} = \{z \ : -1 \leq \ \omega^t z + \omega_0 \leq 1\}$. In the SVM model, positive-class observations ($y = +1$) will be forced to lie on the positive half-space, and the same constraint will be imposed for the negative-class ($y = -1$) observations on the negative half-space. When these constraints are violated for an observation, a penalization error is accounted for in the optimization problem. The separation (margin) between classes is computed as the width of the strip $\mathcal{S}$. As mentioned before, the SVM separating hyperplane will be obtained from an equilibrium of maximizing the separation between classes and minimizing these penalization errors. Denoting by $e_i \in \mathbb{R}^+$ the misclassification error of observation i, and by C the constant of penalization of these errors, the SVM can be formulated as the following Non Linear Problem (NLP):

$$
\begin{aligned}
\min \ & \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{n} e_i \\
\text{s.t.} \ & y_i(\omega^t x_i + \omega_0) \geq 1 - e_i, && \forall i = 1, \ldots, n \\
& \omega \in \mathbb{R}^p, \ \omega_0 \in \mathbb{R}, \\
& e_i \in \mathbb{R}_+, && \forall i = 1, \ldots, n.
\end{aligned}
$$

In Fig. 2 we can see a set of points belonging to two different, blue and green, classes (left picture) and its SVM optimal solution for a given parameter C (right picture). The black line is the separating hyperplane while the other two parallel lines are delimiting the strip, $\mathcal{S}$, between classes. The points that lie on these parallel lines, the boundary of the strip, are the so called support vectors, and they verify that $|\omega^t x_i + \omega_0| = 1$. Finally, we represent in red color the magnitude of the errors induced by margin violations.

If we further analyze the above dataset, we can see that there are four blue observations at the very right of the dataset, and two green observations on the left that have a strong impact when building the

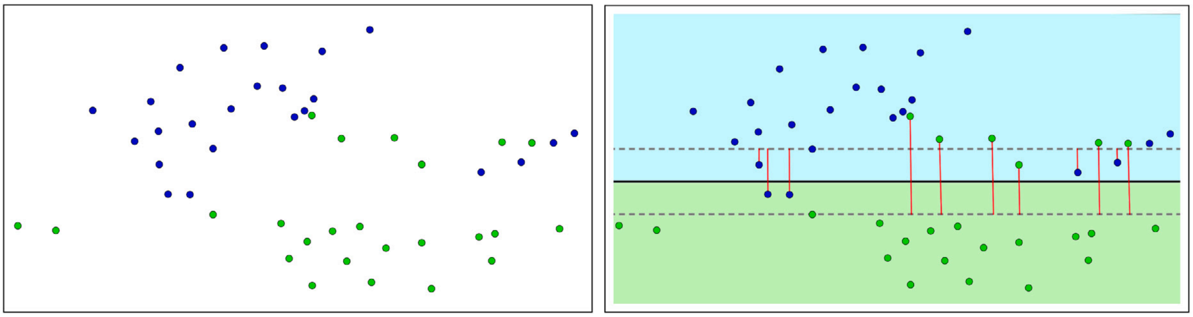

Fig. 2. Original set of points (left) and optimal SVM solution on these points (right).

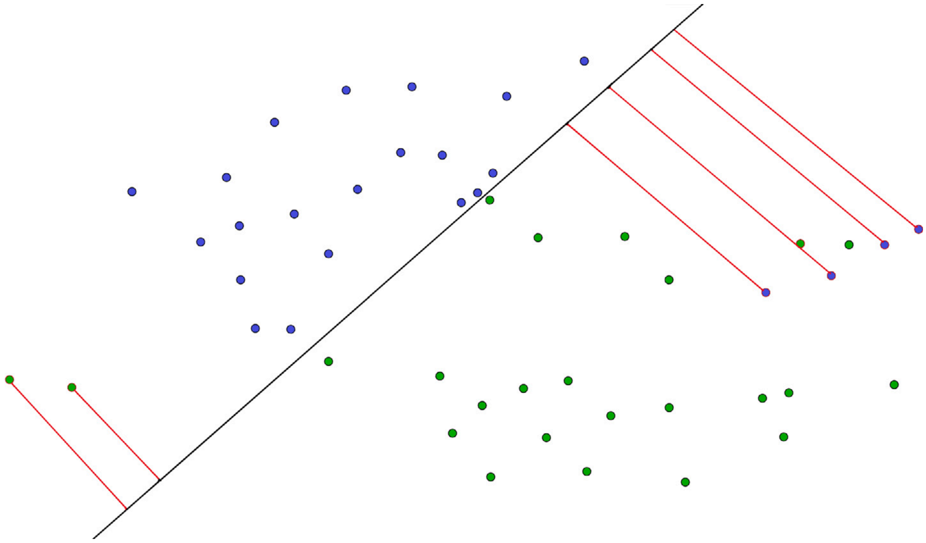

Fig. 3. Not optimal solution on the SVM problem.

classifier. These observations do not allow one to construct a SVM separator of the dataset as the one we can see in Fig. 3, since that would lead to very big misclassification errors with a very tiny margin.

Moreover, there are another two green observations, besides the two on the left, that are closer to the blue cloud of points than to the green one. Hence, if we could consider that these four green points and the four blue ones on the right were wrongly labeled (because of their closeness to the rest of points), we might consider a separating hyperplane with a slope like the one presented on the left of Fig. 4 as a better classifier. However, this separating hyperplane would be impossible to obtain with the SVM model since all the support vectors belong to the same class and to avoid huge misclassification errors the model would forbid such a slope.

Motivated by the above kind of configurations, we have studied different models in which a separating hyperplane is obtained not only based on the original labels but also on the possibility of relabeling some of the original observations of the training sample at a given penalty cost. We say that an observation is relabeled if one of the following assumptions occurs:

$y_i = \pm 1$ but our model considers that $y_i = \mp 1$.

We will use the notation $\hat{y}_i$ to represent the class that the model is considering for observation i. Hence, an observation is said to be relabeled if $y_i \neq \hat{y}_i$.

Following the example shown in Figs. 2 and 3, we can see on the right of Fig. 4 the solution of our model, with a separating hyperplane with the desired slope. Considering the original classes (blue and green), purple points represent the points that the model considers to be blue (despite of their actual label), and orange points represent the points that the model considers to be green. This separating hyperplane is optimal in our problem, the model considers that support points belong to different classes (even thought that is not true regarding to the original values) and no misclassification errors appear in the solution (which is also not true for the original labels). The underlying idea in these models is that based on the geometry of the problem, relabeling some observations can lead to more robust/accurate classifiers. These classifiers can be very useful when dealing with datasets with outliers, and also in datasets in which some noise is known to be added to the data labels.

3. Mathematical programming models

In this section we present the three mathematical optimization models that we propose to solve the problem consisting in building a hyperplane for binary classification, and, simultaneously, relabeling potential noisy observations. In the first model, relabeling labels on the original observations will be based on the errors with respect to the separating hyperplane. On the other hand, besides considering the errors with respect to the separating hyperplane, the other two models will also take into account information from data based on the geometry of the points through the k-means and the k-medians methods. Nevertheless, despite the fact that some observations are relabeled in our models, in order to make predictions, we will maintain the state for predictions on out of sample data which establishes that observations that lie on the positive half-space of the separating hyperplane will be predicted as positive class observations, meanwhile observations that lie on the negative half-space will be predicted as negative class observations.

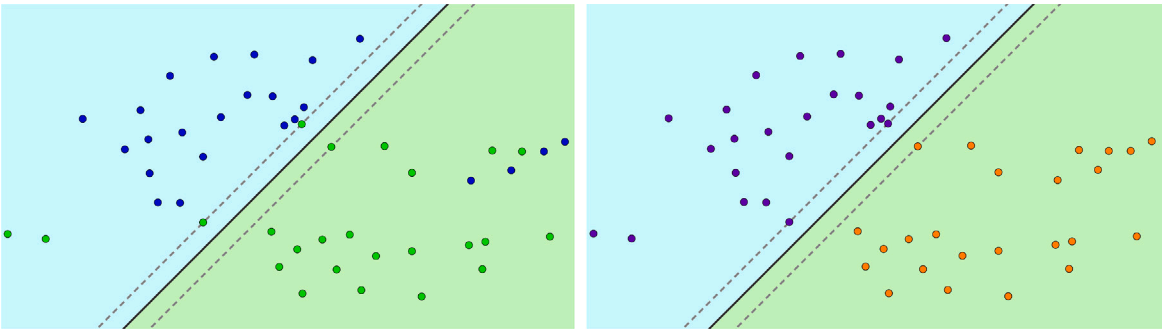

Fig. 4. Optimal solution after re-labeling.

3.1. Model 1: Re-label SVM

The first model that we propose relies on a very basic idea, observations will be relabeled based on the error with respect to the separating hyperplane, i.e., a penalty for each relabeling will be considered and the model will determine whether the cost compensates the global misclassification error. Let $\hat{y}_i$ be the final label for the observation i (after relabeling), for all $i = 1,\dots,n$. Hence, using the notation introduced before, the model can be synthetically summarized in the following way:

$$\begin{aligned}
\min\ & \frac{1}{2}\|\omega\|_2^2 + C_1\sum_{i=1}^n e_i + \text{relabelingCost}(\hat{y}) & \\
\text{s.t.}\ & \hat{y}_i(\omega^t x_i + \omega_0) \ge 1 - e_i, & \forall i = 1,\dots,n \\
& \omega \in \mathbb{R}^p,\ \omega_0 \in \mathbb{R}, & \\
& e_i \in \mathbb{R}^+, & \forall i = 1,\dots,n, \\
& \hat{y}_i \in \{-1,1\}, & \forall i = 1,\dots,n.
\end{aligned}$$

The model above is a SVM model in which observations can be relabeled, and thus, instead of considering y_i on the separability constraint, the relabeled observations $\hat{y}_i$ are used. In what follows we describe how to incorporate the relabeling to the constraints and the objective function. Observe that if no cost is assumed for relabeling, the model will relabel most of the observations to obtain a null misclassification error, resulting in senseless classifiers. Thus, we model this cost with a penalty, so that the model will try to maintain the original labels on data and it will only relabel observations when a strong gain on the margin or a strong minimization on the errors is produced.

Note that classical linear SVM-based methodologies measure the misclassification error for a given training observation (x_i, y_i) by means of the distance from x_i to the *correct* halfspace ($\mathcal{H}^+$ or $\mathcal{H}^-$) with respect to its label y_i. In contrast, in this new model, since the observation may be conveniently re-labeled, the misclassification error, although measured also as the distance from x_i to one of the halfspaces $\mathcal{H}^+$ or $\mathcal{H}^-$, the reference halfspace is determined by the actual label ($\hat{y}_i$) provided by the model instead that by the original one. At the end, misclassification errors are measured exactly in the same way in both models, but in Re-label SVM the originals labels may be modified, paying a penalty cost for relabeling, implying more flexibility when fitting the SVM-based separating hyperplanes.

In order to derive a suitable mathematical programming formulation for the problem, we consider the following set of binary variables to model relabeling:

$$\xi_i = \begin{cases} 1, & if\ \ \hat{y}_i = -y_i, \\ 0, & \text{otherwise.} \end{cases} \quad \text{for } i = 1,\dots,n.$$

With these variables, $\text{relabelingCost}(\hat{y}) = C_2\sum_{i=1}^n \xi_i$, where C_2 is the unitary cost of relabeling. Also, to construct the classifier, we consider the following auxiliary set of continuous variables:

$$\beta_{ij} = \begin{cases} \omega_j, & \text{if observation } i \text{ is relabeled,} \\ 0, & \text{otherwise.} \end{cases} \in \mathbb{R} \ \text{ for } i = 1,\dots,n, \ \text{ for } j = 0,\dots,p,$$

and by $\beta_i = (\beta_{i1},\dots,\beta_{ip}) \in \mathbb{R}^p$.

Observe that, with the above notation,

$$\hat{y}_i(\omega^t x_i + \omega_0) = y_i(\omega^t x_i + \omega_0) - 2y_i(\beta_i^t x_i + \beta_{i0}).$$

Based on the discussion above, our problem can be formulated as follows:

$$\min \frac{1}{2}\|\omega\|_2^2 + C_1\sum_{i=1}^n e_i + C_2\sum_{i=1}^n \xi_i \tag{RE-SVM}$$

$$\text{s.t.}\ \ y_i(\omega^t x_i + \omega_0) - 2y_i(\beta_i^t x_i + \beta_{i0}) \ge 1 - e_i, \quad \forall i = 1,\dots,n, \tag{1}$$

$$\beta_{ij} = \xi_i\omega_j, \quad \forall i = 1,\dots,n, j = 0,\dots,p, \tag{2}$$

$$\omega \in \mathbb{R}^p,\ \omega_0 \in \mathbb{R}, \tag{3}$$

$$\beta_i \in \mathbb{R}^p,\ \beta_{i0} \in \mathbb{R}, \quad \forall i = 1,\dots,n, \tag{4}$$

$$e_i \in \mathbb{R}^+,\ \xi_i \in \{0,1\}, \quad \forall i = 1,\dots,n. \tag{5}$$

In the formulation above, constraints (1) and (2) allow one to model the relabeled observations whereas (3) declares that the coefficients of the hyperplane are continuous variables. Constraint (4) defines a set of variables that will be equal to the coefficients of the hyperplane when an observation is relabeled, and zero otherwise. With these new coefficients, if an observation is not relabeled, constraints (1) coincide with those of the classical SVM, that together with the objective function and (5) allow one modeling the misclassification errors as hinge losses, i.e. $e_i = \max\{0, 1 - \hat{y}_i(\omega^t x_i + \omega_0)\}$ for all $i = 1,\dots,n$.

Note that (RE-SVM) is a Mixed Integer Nonlinear Problem due to its objective function, because even though constraints (2) are written in a nonlinear way, they can be linearized as follows:

$$\begin{aligned}
\omega_j - M(1-\xi_i) \le \beta_{ij} \le \omega_j + M(1-\xi_i),\ \forall i = 1,\dots,n, j = 0,\dots,p, \\
-M\xi_i \le \beta_{ij} \le M\xi_i,\ \forall i = 1,\dots,n, j = 0,\dots,p,
\end{aligned}$$

for $M \gg 0$ a big enough constant. Observe that one can always assume that the coefficients of the hyperplane are normalized and that $\|(\omega,\omega_0)\|_\infty \le 1$, and then, the value of M can be fixed to one.

With the above considerations, (RE-SVM) can be reformulated as a Quadratic Mixed Integer Programming problem with linear constraints (MIQP), which can be solved by the available off-the-shelf solvers (Gurobi, CPLEX, XPRESS, …), which use a non-linear branch and bound approach (Gupta & Ravindran, 1985) whose continuous subproblems are efficiently solved using interior-point algorithms.

Remark 3.1. In the same manner that we formulate the problem above using a hinge-loss point of view for the misclassification errors, it can be easily adapted to other loss functions as the ramp loss (Huang, Shi, & Suykens, 2014). This latter case results in the following mathematical programming model:

$$\min \frac{1}{2}\|\omega\|_2^2 + C\left(\sum_{i=1}^n e_i + 2\sum_{i=1}^n \xi_i\right) \tag{RL-SVM}$$

$$\text{s.t.}\ \ y_i(\omega^t x_i + \omega_0) \ge 1 - e_i - M\xi_i, \qquad \forall i = 1,\dots,n$$

$$
\begin{aligned}
& 0 \le e_i \le 2, && \forall i = 1, \dots, n\\
& \xi_i \in \{0,1\}, && \forall i = 1, \dots, n\\
& \omega \in \mathbb{R}^p, \omega_0 \in \mathbb{R}.
\end{aligned}
$$

Here, the observations that lie outside the margin in the wrong side of the separating hyperplane are equally penalized in the objective function regardless of the misclassification distance.

3.2. Cluster-SVM models

The second family of models that we propose for detecting label noise in the data are based on using similarity measures on the observations. These models will be called *Cluster-SVM methods* since they perform, simultaneously, two tasks: clustering and classification by SVM. On the one hand, the *cluster* phase of these methods will induce relabeling based on heterogeneity of the information, whereas the SVM phase computes the classifier after relabeling. We present here two different alternatives for clustering data into two groups and its linkage to a classification system: the 2-median and the 2-mean problems.

The goal of these methods is to find two clusters for a given set of observations, considering that an observation will belong to exactly one cluster. These clusters are built by finding two *reference points (centroids or medians)* representing each of the two groups determined by the observations closer to them, in a way that the overall sum of distances from points to their respective reference points is minimum. We distinguish two models under these settings by using two different distance measures: the ℓ_1 and the ℓ_2 norms.

Let us denote by $K_+ \in \mathbb{R}^p$ and $K_- \in \mathbb{R}^p$ the two (unknown) reference points, and $d_i = \min\{\|x_i - K_+\|, \|x_i - K_-\|\}$, the distance from the observation i to its closest reference points, for $i = 1, \dots, n$ (here $\|\cdot\|$ will represent either the ℓ_1 or the ℓ_2-norm). The representation of such a closest distance to the reference points will be incorporated to the mathematical programming model using the following set of binary variables:

$$\theta_i = \begin{cases} 1, & \text{if observation } i \text{ is assigned to cluster } +,\\ 0, & \text{if observation } i \text{ is assigned to cluster } -, \end{cases} \quad \text{for } i = 1, \dots, n.$$

These clusters represent *similar* observations and will help the SVM methodology, together with the relabeling, to find more accurate classifiers.

Combining the ideas presented on RE-SVM with the clustering based methods, we can derive a new family of models, that assign observations to two groups based on the clusters obtained by minimizing the overall sum of the norm-based distances from the data points to their corresponding reference points. Moreover, it also tries to separate as much as possible these two clusters by means of a hyperplane. Each one of the clusters is assigned to one of the differentiated classes in our classification problem. Finally, this hyperplane will induce a subdivision of the data space in a way that the decision rule of the classification problem for out-of-sample data is the same that is used in standard SVM. We present below the MIP formulation for this problem. Let $M_1, M_2, M_3 \gg 0$ be big enough positive constants and $\|\cdot\|$ representing either the ℓ_1 or the ℓ_2-norm.

$$\min \ \frac{1}{2}\|\omega\| + C_1 \sum_{i=1}^{n} e_i + C_2 \sum_{i=1}^{n} \xi_i + C_3 \sum_{i=1}^{n} d_i \tag{Cluster-SVM}$$

$$
\begin{aligned}
\text{s.t.}\ & y_i(\omega^t x_i + \omega_0) \ge -M_1 \xi_i, && \forall i = 1, \dots, n, && (6)\\
& d_i \ge \|x_i - K_+\| - M_2(1-\theta_i), && \forall i = 1, \dots, n, && (7)\\
& d_i \ge \|x_i - K_-\| - M_2\theta_i, && \forall i = 1, \dots, n, && (8)\\
& \omega^t x_i + \omega_0 \ge 1 - e_i - M_3(1-\theta_i), && \forall i = 1, \dots, n, && (9)\\
& \omega^t x_i + \omega_0 \le -1 + e_i + M_3\theta_i, && \forall i = 1, \dots, n, && (10)\\
& \theta_i,\ \xi_i \in \{0,1\}, && \forall i = 1, \dots, n, && (11)\\
& e_i, d_i \in \mathbb{R}_+, && \forall i = 1, \dots, n, && (12)\\
& K_+, K_- \in \mathbb{R}^p, && && (13)\\
& \omega \in \mathbb{R}^p,\ \omega_0 \in \mathbb{R}. && && (14)
\end{aligned}
$$

Note that the constants M_1, M_2, M_3 in the formulation above must be chosen such that $M_1 > \max_{i=1,\dots,n}\left\{y_i\left(\sum_{j=1,\dots,d} x_{ij} + 1\right)\right\}$ (considering w.l.o.g. that the coefficients are taken so that $\|(\omega,\omega_0)\|_\infty \le 1$), $M_2 > \max\{\|x_i - x_j\| : i,j = 1,\dots,n\}$ and $M_3 > \sum_{j=1,\dots,d} x_{ij} + 2 + \max\{\|x_i - x_j\|_2 : i,j = 1,\dots,n\}$. Tightened values for this constants could be calculated using similar ideas than those elaborated in Baldomero-Naranjo, Martínez-Merino, and Rodríguez-Chía (2020).

The objective function of (Cluster-SVM) aggregates the following four elements to be simultaneously optimized:

- The margin (measured with the ℓ_1 or ℓ_2 norm) has to be maximized.
- The errors of classification with respect to the separating hyperplane have to be minimized.
- Relabeled observations have to be penalized.
- Distances from observations to their reference points have to be minimized.

The aggregation of these four terms leads to define a hyperplane with a good margin, separating two *homogeneous* clusters with respect to distances and classes. Constraint (6) enforces the positive (resp. negative) class observations to be located on the positive (resp. negative) half-space of the separating hyperplane when no relabeling is applied. Each relabeled observation is penalized by C_2 units, not allowing a large number of relabeling unless it compensates large misclassification errors or unless they lead to a margin gain. This methodology allows us to keep the same decision rule for out-of-sample data as the one used in standard SVM. Constraints (7) and (8) permit to determine the closest centroid to each observation, whereas constraints (9) and (10) enforce the misclassification errors to be computed with respect to the cluster, i.e. the classification is performed with respect to the classes θ_i that have been created based on the similarity of the observations.

The above model results in two different problems depending on the norm-based distances applied.

2Median SVM Model This model results from (Cluster-SVM) using the norm ℓ_1. It will be referred to as the 2-Median SVM model. The problem turns out to be a mixed integer linear problem and can be solved using any of the off-the-shelf MIP solvers.

2Mean SVM Model This is the version of model (Cluster-SVM) using the ℓ_2. Since we are using a nonlinear norm, the 2-Means SVM results in a Mixed Integer Nonlinear Programming problem, that can be reformulated as a Mixed Integer Second Order Cone Optimization (MISOCO) problem. As for the MIP there are nowadays available off-the-shelf commercial optimization solvers implementing routines for its efficient solution.

Remark 3.2 (*2-ℓ_τ Cluster SVM Model*)**.** One could also consider different ℓ_τ-norms ($\tau \ge 1$) for both the margin measure and the clusters similarity measures. In this case, the problem becomes also a MINLP problem, but based on the results provided by Blanco, Ben Ali, and Puerto (2014), it can also be efficiently reformulated as a MISOCO problem. These type of problems can be solved by the available off-the-shelf solvers (Gurobi, CPLEX, XPRESS, …), which use a branch and bound approach (Gupta & Ravindran, 1985) where continuous problems in the nodes (Second Order Cone optimization) are efficiently solved using interior-point algorithms.

4. Experiments

In this section we report the results of our computational experience. We have studied seven real datasets from UCI Machine

Learning Repository (see Biggio et al., 2011), all of them are binary classification problems that come from different topics. The datasets used are: Statlog-Australian Credit Approval (Australian), Breast Cancer (BreastCancer), Statlog-Heart (Heart), Parkinson Dataset with replicated acoustic features (Parkinson), QSAR biodegradation (QSARbiodeg), Vertebral Column (Vertebral) and Wholesale Customers (Wholesale). The dimensions (n: number of observations, p: number of features) of these datasets is reported in Table 1.

For each of these datasets we have performed five different experiments. The goal in these experiments is to make predictions as accurate as possible on out of sample data. The first experiment consists on making predictions by training the models with the original data. On the other hand, in order to represent attacks in the training data, we have considered four different scenarios in which a random amount of labels, within the set $\{20\%, 30\%, 40\%, 50\%\}$, have been flipped for training data, i.e., four scenarios in which we have added some label-noise on training data.

We have performed a 5-fold cross validation scheme. Thus, data have been split into 5 train-test random partitions. In each of these folds we have trained our models and we have used the other four folds for testing. Moreover, we have repeated this 5-fold cross validation 5 times for each dataset, in order to avoid beneficial starting partitions, and we report the average results obtained. For all the instances we have trained our three models and we have compared them with standard SVM and SVM-CL (Northcutt et al., 2021). We have considered standard SVM as benchmark since, despite the good results provided by SVM-CL for some experiments, standard SVM provided a better performance on average among all the experiments (see Table 1 and Fig. 5). The measure used to evaluate the performance of the models have been the accuracy, in percentage, on out of sample data:

$$ACC = \frac{\#\text{Well Classified Test Observations}}{\#\text{Test Observations}} \cdot 100$$

The parameters that appear in the different methods that we compare are validated as usual, that is, for each of the instances we perform a grid search on the cost parameters and the best result obtained in the validation sample among these parameters is the one reported. More specifically, the grids used in the experiments are the following:

SVM: $C \in \left\{10^i : i = -5, \dots, 5\right\}$.
RE-SVM: $C_1, C_2 \in \left\{10^i : i = -5, \dots, 5\right\}$.
2-medians-SVM: $C_1, C_2 \in \left\{10^i : i = -5, \dots, 5\right\}$, $C_3 \in \left\{10^i : i = -3, \dots, 0\right\}$.
2-means-SVM: $C_1, C_2 \in \left\{10^i : i = -5, \dots, 5\right\}$, $C_3 \in \left\{10^i : i = -3, \dots, 0\right\}$.
SVM-CL: Default tuning parameters (see Northcutt et al., 2021).

Observe that some approaches required more hyperparameters to calibrate than others. Although it implies a clear computational disadvantage in the training phase, it does not imply a benefit of one method over others in terms of quality of the obtained classifier since training and test are evenly performed with all the models (one fold for training and the remainder for validating).

The mathematical programming models were coded in Python 3.6, and solved using Gurobi 7.5.2 on a PC Intel Core i7-7700 processor at 2.81 GHz and 16 GB of RAM. Due to the complexity of the 2-means-SVM, we have helped the solver uploading an initial feasible solution that was obtained in the 2-medians-SVM problem. We have not solved to optimality all the instances, especially those with the 2-means-SVM in which the problem becomes nonlinear, and hence we have established a time limit of 30 s for all the experiments. This training time has sufficed to obtain rather good classifiers. Indeed, as one can observe from the results obtained, this time limit is adequate to construct robust classifiers under noisy labels. Note that not guarantying the optimality of the solutions of our models does not necessarily imply that the classifiers are not adequate.

Table 1
Accuracy results of our computational experiments.

Dataset	Method	Percentage of flipped labels				
		0%	20%	30%	40%	50%
Australian (690,14)	SVM-CL	84.55	82.10	71.12	58.93	49.68
	SVM	86.11	85.43	79.23	68.13	59.47
	RE-SVM	86.42	85.68	83.37	76.97	66.13
	2-medians-SVM	86.08	85.84	84.67	78.95	69.54
	2-means-SVM	85.97	85.74	82.65	77.14	67.70
BreastCancer (683,9)	SVM-CL	95.73	91.87	87.37	78.49	58.36
	SVM	96.49	93.47	89.96	85.94	68.16
	RE-SVM	96.88	96.20	94.97	90.36	77.00
	2-medians-SVM	96.63	95.31	94.46	91.10	87.31
	2-means-SVM	96.96	95.93	95.39	93.11	90.01
Heart (270,13)	SVM-CL	78.70	71.03	60.09	56.01	49.66
	SVM	82.23	76.86	69.68	63.79	56.90
	RE-SVM	82.84	78.38	73.16	68.86	61.25
	2-medians-SVM	82.01	78.75	77.29	75.38	71.99
	2-means-SVM	82.06	78.81	77.40	75.97	72.90
Parkinson (240,40)	SVM-CL	78.18	65.56	59.47	55.58	49.29
	SVM	81.66	74.74	70.17	62.28	57.82
	RE-SVM	82.43	77.64	73.22	67.29	62.97
	2-medians-SVM	80.32	78.62	78.12	77.51	76.28
	2-means-SVM	80.47	79.22	78.78	78.20	77.03
QSARbiodeg (1055,40)	SVM-CL	81.62	78.86	74.07	56.78	46.78
	SVM	82.12	78.07	74.09	63.38	48.97
	RE-SVM	84.53	79.61	75.00	66.42	54.58
	2-medians-SVM	84.08	78.79	74.32	67.87	67.02
	2-means-SVM	83.61	78.55	74.42	67.86	66.81
Vertebral (310,6)	SVM-CL	80.94	72.79	68.54	60.53	50.69
	SVM	84.51	75.43	71.34	66.78	57.47
	RE-SVM	85.10	79.61	74.83	72.33	67.92
	2-medians-SVM	85.31	82.62	80.80	78.30	76.31
	2-means-SVM	86.28	84.32	81.77	79.91	76.76
Wholesale (440,7)	SVM-CL	88.98	85.40	78.03	57.19	45.42
	SVM	90.08	85.30	79.74	72.23	57.73
	RE-SVM	90.39	88.77	85.97	80.12	69.07
	2-medians-SVM	90.58	89.54	87.79	82.78	73.54
	2-means-SVM	91.23	89.56	87.39	85.88	82.92

In Table 1 we report the average accuracy results obtained in all the experiments for the different models and the different levels of label-noise. In such a table we have used the yellow-green color to indicate the results in which we are a $3\% - 5\%$ better than the benchmark, the green color to indicate whether we are a $5\% - 10\%$ better than the benchmark, and the cyan color to highlight the results in which we are at least a 10% above the benchmark. Also, we show in Fig. 5 the accuracy boxplots of the 625 instances per dataset (5 partitions × 5 scenarios × 5 folds × 5 models).

Regarding to the results, several conclusions can be pointed out:

- Our three models perform consistently better than classical SVM when the training dataset is corrupted. Besides, the stronger the percentage of flipped labels, the bigger the difference between our models' results and SVM's results. In Fig. 5 one can check how SVM model has lower tails and wider boxes than RE-SVM.
- 2-medians-SVM and 2-means-SVM perform better than RE-SVM for heavy attacks ($40\% - 50\%$ of flipped observations). In contrast, the cluster-based models require more time to be trained than RE-SVM, both because the problems are harder to solve (apart from relabeling, the distances to the centroids and the assignments observations-to-centroids are modeled) and the number of parameters that must be tuned. In Fig. 5 one can easily check that RE-SVM has wider boxes than 2-medians-SVM and 2-means-SVM, which are explained by the behavior of these models against the attacks.
- Our models have a better performance than the rest of approaches even for the original datasets in which no labels are flipped. This

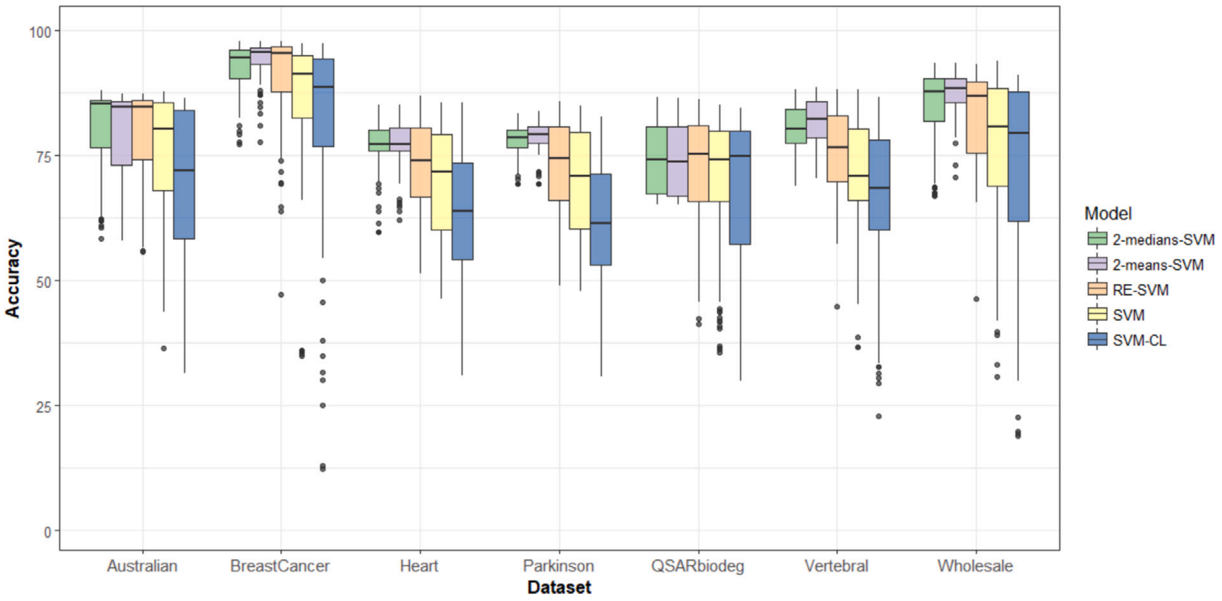

Fig. 5. Accuracy Boxplots of the obtained accuracies.

is due to the flexibility offered by methodologies, because some of the observations are allowed to be relabeled looking for a better classifier. The original datasets may contain outliers that contaminate the sample and so they deteriorate the classifier. This situation is automatically detected and fixed by our methods, by adequately relabeling observations.

- Our methods outperform SVM-CL, which is a specialized method, designed to detect noisy labels. The rationale under these results is that SVM-CL seems to wrongly identify the right distributions of the data. These mistakes propagate to the construction of the classifier since it is built on some incomplete data. This fact also results in worse accuracies than standard SVM that works with the entire dataset without paying attention to the existence of outliers.

Overall, as one may expect and it is confirmed in our computational experiments, it is better to construct the classifier without identifying *incorrectly* the noise labels (as SVM does) than using inadequate flips to build the classifier (as SVM-CL seems to do in the tested datasets). Obviously, the results in the paper also show that it is rather advantageous the correct identification of the wrong labels since it improves significantly the classification rates.

5. Conclusions

This paper presents a methodology to construct a classification rule that at the same time incorporates the detection of label noise in the datasets. Our methodology combines the power of SVM and the features of clustering analysis to simultaneously identify wrong labels to build a separating hyperplane maximizing the margin, minimizing the misclassification errors and penalizing relabeling. The rationale is simple: observations identified as wrongly labeled will be relabeled only if the gain in margin or the decrease in misclassification error compensate the flipping. In spite of its theoretical simplicity we show the exceptional performance of our methodology in a number of databases taken from the UCI repository.

These models are implemented using mathematical programming formulations with some integer variables (MIP). In all cases, they give rise to models that are simple and that enjoy the *quality* of being solvable by nowadays off-the-shelf commercial solvers (Gurobi, CPLEX, XPRESS...)

Our findings are not only of theoretical interest. Its practical performance when applied to databases is remarkable. In all tested cases, our methods are superior to the considered benchmark that in our case is standard SVM. Thus, they are directly applicable to datasets in which flipped labels are suspected, resulting in robust classifiers to noisy labels.

Further research on the topic includes the extension of our models to deal with multiclass instances by modifying the *relabel* -variables to identify the new (non-binary) labels. The strategy should be carefully chosen using a multiclass SVM-based approach (as One versus One, One versus All or any of the unified tools). This extension is not trivial and requires a deeper analysis.

Other lines of research that would extend our methods are the application of alternative clustering strategies, as those based on ordered median objective functions or the twin SVM methodology. Also, the use of kernel tools in our approaches, in order to be able to construct non linear classifiers has to be investigated.

CRediT authorship contribution statement

Víctor Blanco: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Roles/Writing – original draft, Writing – review & editing. **Alberto Japón:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Roles/Writing – original draft, Writing – review & editing. **Justo Puerto:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Roles/Writing – original draft, Writing – review & editing.

References

Bahlmann, C., Haasdonk, B., & Burkhardt, H. (2002). On-line handwriting recognition with support vector machines a kernel approach. In *IWFHR'02, Proceedings of the eighth international workshop on frontiers in handwriting recognition* (pp. 49–54).

Baldomero-Naranjo, M., Martínez-Merino, L. I., & Rodríguez-Chía, A. M. (2020). Tightening big Ms in integer programming formulations for support vector machines with ramp loss. *European Journal of Operational Research*, *286*(1), 84–100.

Bertsimas, D., Dunn, J., Pawlowski, C., & Zhuo, Y. D. (2019). Robust classification. *INFORMS Journal on Optimization*, *1*(1), 2–34.

Bi, J., & Zhang, T. (2005). Support vector classification with input data uncertainty. In *Advances in neural information processing systems* (pp. 161–168).

Biggio, B., Nelson, B., & Laskov, P. (2011). Support vector machines under adversarial label noise. In *Asian conference on machine learning* (pp. 97–112).

Blanco, V., Ben Ali, S., & Puerto, J. (2014). Revisiting several problems and algorithms in continuous location with l_p norms. *Computational Optimization and Applications*, *58*(3), 563–595.

Blanco, V., Japón, A., Ponce, D., & Puerto, J. (2021). *Computers & Operations Research*, *128*, Article 105124.

Blanco, V., Japón, A., & Puerto, J. (2020a). Optimal arrangements of hyperplanes for multiclass classification. *Advances in Data Analysis and Classification*, *14*, 175–199.

Blanco, V., Japón, A., & Puerto, J. (2020b). Robust optimal classification trees under noisy labels. *Advances in Data Analysis and Classification*, *16*, 155–179.

Blanco, V., Puerto, J., & Rodríguez-Chía, A. M. (2020). On ℓ_p-support vector machines and multidimensional kernels. *Journal of Machine Learning Research*, *21*.

Blanco, V., Puerto, J., & Salmerón, R. (2018). Locating hyperplanes to fitting set of points: A general framework. *Computers & Operations Research*, *95*, 172–193.

Boucher, J.-P., Denuit, M., & Guillen, M. (2009). Number of accidents or number of claims? An approach with zero-inflated Poisson models for panel data. *The Journal of Risk and Insurance*, *76*(4), 821–846.

Chen, P., Liao, B. B., Chen, G., & Zhang, S. (2019). Understanding and utilizing deep neural networks trained with noisy labels. In *International conference on machine learning* (pp. 1062–1070). PMLR.

Cheng, Y., & Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the 8th int. conf. on intelligent systems for molecular biology* (pp. 93–103).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

de França, F. O., & Coelho, A. L. (2015). A biclustering approach for classification with mislabeled data. *Expert Systems with Applications*, *42*(12), 5065–5075.

Duan, Y., & Wu, O. (2018). Learning with auxiliary less-noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(7), 1716–1721.

Ekambaram, R., Fefilatyev, S., Shreve, M., Kramer, K., Hall, L. O., Goldgof, D. B., et al. (2016). Active cleaning of label noise. *Pattern Recognition*, *51*, 463–480.

Federal Trade Commission (2017). *Consumer sentinel network data book for January-December 2016*.

Frénay, B., & Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869.

Ganapathiraju, A., & Picone, J. (2000). Support vector machines for automatic data cleanup. In *Sixth international conference on spoken language processing*.

Ghoggali, N., & Melgani, F. (2009). Automatic ground-truth validation with genetic algorithms for multispectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(7), 2172–2181.

Gupta, O. K., & Ravindran, A. (1985). Branch and bound experiments in convex nonlinear integer programming. *Management Science*, *31*(12), 1533–1546.

Han, X., & Chang, X. (2013). An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms. *Information Sciences*, *218*, 103–118.

Huang, X. L., Shi, L., & Suykens, J. A. K. (2014). Ramp loss linear programming support vector machine. *Journal of Machine Learning Research*, *15*, 2185–2211.

Ikeda, K., & Murata, N. (2005a). Effects of norms on learning properties of support vector machines. In *ICASSP (5)* (pp. 241–244).

Ikeda, K., & Murata, N. (2005b). Geometrical properties of nu support vector machines with different norms. *Neural Computation*, *17*(11), 2508–2529.

Kang, S., Kim, D., & Cho, S. (2019). Approximate training of one-class support vector machines using expected margin. *Computers & Industrial Engineering*, *130*, 772–778.

Kašćelan, V., Kašćelan, L., & Novović Burić, M. (2016). A nonparametric data mining approach for risk prediction in car insurance: a case study from the montenegrin market. *Economic Research-Ekonomska Istraživanja*, *29*(1), 545–558.

Labbé, M., Martínez-Merino, L. I., & Rodríguez-Chía, A. M. (2018). Mixed integer linear programming for feature selection in support vector machine. *Discrete Applied Mathematics*, http://dx.doi.org/10.1016/j.dam.2018.10.025.

Liu, L., Chu, M., Gong, R., & Peng, Y. (2020). Nonparallel support vector machine with large margin distribution for pattern classification. *Pattern Recognition*, *106*, Article 107374.

Liu, L., Martín-Barragán, B., & Prieto, F. J. (2021). A projection multi-objective SVM method for multi-class classification. *Computers & Industrial Engineering*, *158*, Article 107425.

Liu, S., Niles-Weed, J., Razavian, N., & Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. arXiv preprint arXiv:2007.00151.

López, J., Maldonado, S., & Carrasco, M. (2018). Double regularization methods for robust feature selection and SVM classification via DC programming. *Information Sciences*, *429*, 377–389.

Majid, A., Ali, S., Iqbal, M., & Kausar, N. (2014). Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines. *Computer Methods and Programs in Biomedicine*, *113*(3), 792–808.

Maldonado, S., Bravo, C., López, J., & Pérez, J. (2017). Integrated framework for profit-based feature selection and SVM classification in credit scoring. *Decision Support Systems*, *104*, 113–121.

Maldonado, S., Merigó, J., & Miranda, J. (2018). Redefining support vector machines with the ordered weighted average. *Knowledge-Based Systems*, *148*, 41–46.

Mangasarian, O. L. (1999). Arbitrary-norm separating plane. *Operations Research Letters*, *24*(1–2), 15–23.

Nalepa, J., & Kawulok, M. (2018). Selecting training sets for support vector machines: a review. *Artificial Intelligence Review*, 1–44.

Natarajan, N., Dhillon, I. S., Ravikumar, P., & Tewari, A. (2017). Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research*, *18*(1), 5666–5698.

Northcutt, C. G., Jiang, L., & Chuang, I. L. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence*, *70*, 1373–1411.

Okwuashi, O., & Ndehedehe, C. E. (2020). Deep support vector machine for hyperspectral image classification. *Pattern Recognition*, Article 107298.

Peng, X., & Chen, D. (2018). PTSVRs: Regression models via projection twin support vector machine. *Information Sciences*, *435*, 1–14.

Peng, X., Xu, D., Kong, L., & Chen, D. (2016). L1-norm loss based twin support vector machine for data recognition. *Information Sciences*, *340–341*, 86–103.

Radhimeenakshi, S. (2016). Classification and prediction of heart disease risk using data mining techniques of support vector machine and artificial neural network. In *International conference o computing for sustainable global development (INDIACom)* (pp. 3107–3111). IEEE.

Shin, H. J., Eom, D. H., & Kim, S. S. (2005). One-class support vector machines—an application in machine fault detection and classification. *Computers & Industrial Engineering*, *48*(2), 395–408.

Tanaka, D., Ikami, D., Yamasaki, T., & Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5552–5560).

Ünlü, R. (2021). Cost-oriented LSTM methods for possible expansion of control charting signals. *Computers & Industrial Engineering*, *154*, Article 107163.

Weerasinghe, S., Erfani, S. M., Alpcan, T., & Leckie, C. (2019). Support vector machines resilient against training data integrity attacks. *Pattern Recognition*, *96*, 1–14.

Xiao, H., Biggio, B., Nelson, B., Xiao, H., Eckert, C., & Roli, F. (2015). Support vector machines under adversarial label contamination. *Neurocomputing*, *160*, 53–62.

Xu, L., Crammer, K., & Schuurmans, D. (2006). Robust support vector machine training via convex outlier ablation. In *AAAI, Vol. 6* (pp. 536–542).

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., & Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International conference on machine learning* (pp. 7164–7173). PMLR.