# Clustering Categories in Support Vector Machines *

Emilio Carrizosa[1], Amaya Nogales-Gómez[1],
and Dolores Romero Morales[2]

[1] Departamento de Estadística e Investigación Operativa
Facultad de Matemáticas
Universidad de Sevilla
Spain
{ecarrizosa,amayanogales}@us.es

[2] Saïd Business School
University of Oxford
United Kingdom
dolores.romero-morales@sbs.ox.ac.uk

**Abstract**

Support Vector Machines (SVM) is the state-of-the-art in Supervised Classification. In this paper the Cluster Support Vector Machines (CLSVM) methodology is proposed with the aim to reduce the complexity of the SVM classifier in the presence of categorical features. The CLSVM methodology lets categories cluster around their peers and builds an SVM classifier using the clustered dataset. Four strategies for building the CLSVM classifier are presented based on solving: the original SVM formulation, a Quadratically Constrained Quadratic Programming formulation, and a Mixed Integer Quadratic Programming formulation as well as its continuous relaxation.

The computational study illustrates the performance of the CLSVM classifier using two clusters. In the tested datasets our methodology achieves comparable accuracy to that of the SVM with original data but with a dramatic decrease in complexity.

**Keywords:** Support vector machines, Categorical features, Classifier complexity, Clustering, Quadratically constrained programming, 0-1 programming

*This work has been partially supported by projects MTM2012-36163 of Ministerio de Economía y Competitividad, Spain, P11-FQM-7603 and FQM-329 of Junta de Andalucía, Spain.

# 1 Introduction

In Supervised Classification, [1, 14, 28], we are given a set of objects $\Omega$ partitioned, in its simplest setting, into two classes, and the aim is to classify new objects. Given an object $i \in \Omega$, it is represented by a vector $(x_i, x_i', y_i)$. The feature vector $x_i$ is associated with $J$ categorical features, that can be binarized by splitting each feature into a series of 0-1 dummy features, one for each category, and takes values on a set $X \subseteq \{0,1\}^{\sum_{j=1}^{J} K_j}$, where $K_j$ is the number of categories of feature $j$. The feature vector $x_i'$ is associated with $J'$ continuous features and takes values on a set $X' \subseteq \mathbb{R}^{J'}$. Finally, $y_i \in \{-1, +1\}$ is the class membership of object $i$. Information about objects is only available in the so-called *training sample*, with $n$ objects.

In many applications of Supervised Classification datasets are composed by a large number of features and/or objects, making it hard to both build the classifier and interpret the results. In this case, it is desirable to obtain a less complex classifier, which may make classification easier to handle and interpret, less prone to overfitting and computationally cheaper when classifying new objects. The most popular strategy proposed in the literature to achieve this goal is feature selection [12, 13, 26], which aims at selecting the subset of most relevant features for classification while maintaining or improving accuracy and preventing the risk of overfitting. Feature selection reduces the number of features by means of all-or-nothing procedure. For categorical features, binarized as explained above, it simply ignores some categories of some features, and does not give valuable insight on the relationship between feature categories. These issues may imply a significant loss of information.

A state-of-the-art method in Supervised Classification is Support Vector Machines (SVM). The SVM aims at separating both classes by means of a classifier, $(\omega)^\top x + (\omega')^\top x' + b = 0$, $(\omega, \omega')$ being the so-called score vector, where $\omega$ is associated with the categorical features and $\omega'$ is associated with the continuous features. Given an object $i$, it is classified in the positive or the negative class, according to the sign of the score function, $sign((\omega)^\top x_i + (\omega')^\top x_i' + b)$, while for the case $(\omega)^\top x_i + (\omega')^\top x_i' + b = 0$, the object is classified randomly. See [4, 9, 13, 17, 21] for successful applications of the SVM and [8] for a recent review on Mathematical Optimization and the SVM. In this paper, a methodology to reduce the complexity of the Support Vector Machines (SVM) classifier for datasets composed by categorical features, sometimes containing many categories, and eventually continuous features, is proposed. This is done by clustering the different categories of each categorical feature into a given number of clusters, and then obtaining an SVM-type classifier for the clustered dataset. We call this the Cluster Support Vector Machines (CLSVM) methodology and we will refer to the CLSVM classifier.

As an illustration, let us consider the well-known German credit dataset, `german`, which is one of the datasets from the UCI repository, [3], used in our computational tests. This is a credit scoring dataset, with *good* customers defining the positive class ($y = +1$) and bad customers defining the negative class ($y = -1$), and has been used in the context of Supervised Classification, such as in [2]. In this dataset each object is composed by 20 features: 11 categorical features, binarized into 52 dummies, and 9 continuous features. For this dataset, the SVM formulation with original data, hereafter denoted by $\text{SVM}^{\text{O}}$, gives a classifier leading to a classification

accuracy of 76.67% and whose categorical score subvector $\omega'$ has 50 relevant features, i.e., $card(\{\omega'_j \neq 0\}) = 50$. However, using the CLSVM methodology described in this paper, where the categories of each categorical feature are grouped just into two clusters, the classification accuracy is increased to 80.00% while the CLSVM classifier uses $2 \times 11 = 22$ relevant dummies. In other words, the methodology proposed here allows one to obtain a much simpler classifier with an accuracy even higher than the original one. The clustering is shown in Figure 8, where we can see each categorical feature separated by a discontinuous line and each category from each categorical feature represented by a circle. The two clusters are distinguished by the coloring with dark grey and light grey circles. For instance, the categorical feature "Property" originally had four categories, namely, "real estate", "building society savings agreement/life insurance", "car or other" and "unknown/no property". As we will see later, the three first categories, colored in dark grey, are those indicating *good* customers, and will be grouped into one single cluster, against the category indicating *bad* customers, namely "unknown/no property".

In this paper, four strategies to build the CLSVM classifier are proposed using different mathematical optimization formulations. The first strategy proposed solves the SVM$^{\mathrm{O}}$ as initial step. Then, categories are clustered using the SVM$^{\mathrm{O}}$ scores and the CLSVM classifier consists of building an SVM classifier using the clustered values. For the second strategy a Mixed Integer Nonlinear Programming (MINLP) formulation of the same type as the SVM formulation is proposed, but in this case defining a score for each cluster of each categorical feature. The second strategy is based on solving the continuous relaxation of this MINLP formulation, a Quadratically Constrained Quadratic Programming (QCQP) formulation to find a clustering, and the CLSVM classifier consists of building again an SVM classifier using the clustered dataset. The third and fourth strategies are based on a Mixed Integer Quadratic Programming (MIQP) formulation derived from the MINLP formulation using the *big M* modeling trick to reformulate the nonlinear terms in the feasible region. The third strategy works similarly to the second one, but solves the continuous relaxation of the MIQP. The fourth strategy solves the MIQP formulation itself and obtains the clustering and the classifier at once.

In the computational results, the four strategies are compared against the SVM$^{\mathrm{O}}$ in ten real-life datasets using two performance criteria, namely accuracy and complexity of the classifier for the categorical data. We conclude from our experiments that the CLSVM achieves a comparable or even better accuracy than the SVM$^{\mathrm{O}}$ in nine of the ten datasets tested. In addition, the CLSVM methodology provides a reduction on the complexity of the classifier for the categorical data, while the SVM$^{\mathrm{O}}$ uses more dummy features for all the strategies and for all ten datasets.

The remainder of this paper is organized as follows. In Section 2 we set up notation and terminology. Then, the CLSVM methodology is introduced together with two mathematical optimization formulations. Two theoretical results relevant to the formulations are presented. In Section 3 the four CLSVM strategies are presented. Section 4 is devoted to the computational experience, where the CLSVM classifier and the SVM$^{\mathrm{O}}$ classifier are compared using ten datasets. Finally, Section 5 contains a brief summary, final conclusions and some lines for future research.

## 2 The CLSVM methodology

In this section the CLSVM methodology is introduced. An MINLP formulation is presented for building the CLSVM classifier and some theoretical results for the formulation are stated and proved. Then, an MIQP formulation is derived from the MINLP one, using the big $M$ modeling trick to reformulate the nonlinear terms in the feasible region. The theoretical results also hold for this formulation.

The CLSVM methodology is based on the SVM formulation, but takes into account the way categorical features are handled in the SVM (and other linear classifiers): splitting each feature into a series of 0-1 dummy features, the classifier assigns one score to each dummy feature, and thus to each value of the categorical feature. Instead, the CLSVM methodology lets dummies cluster around their peers and builds an SVM classifier using the clustered dataset, which may reduce the number of relevant features. We will say that category $k$ from categorical feature $j$ is relevant to the classifier if $\omega_{j,k} \neq 0$. Similarly, if $\omega'_{j'} \neq 0$, then we will say that continuous feature $j'$ is relevant to the classifier. Let us focus now on categorical features. If a category is relevant to the classifier, we will say that category $k$ from feature $j$ points towards the positive class if the score associated to the category is positive, i.e., if $\omega_{j,k} > 0$. Analogously, if $\omega_{j,k} < 0$ we will say that category $k$ from feature $j$ points towards the negative class. The fact that a category points towards the positive (or negative) class means that it contributes to classify objects in the positive (or negative) class respectively, i.e., contributes to make $sign((\omega)^\top x_i + (\omega')^\top x'_i + b)$ equal to $+1$ $(-1)$.

Let us introduce some notation. Given an object $i$ in the *training set*, we let $x_i = (x_{i,j,k})$, where $x_{i,j,k}$ is equal to 1 if the value of categorical feature $j$ in object $i$ is equal to category $k$ and 0 otherwise.

First, we present the standard SVM formulation, [8, 10, 23, 24]. The SVM aims at separating both classes by means of a hyperplane, found by minimizing the so-called *hinge loss* and the squared $l_2$-norm of the score vector, [8]. The SVM classifier is obtained by solving the following Quadratic Programming (QP) formulation with linear constraints:

$$\min_{\omega,\omega',b,\xi} \sum_{j=1}^{J}\sum_{k=1}^{K_j} \frac{(\omega_{j,k})^2}{2} + \sum_{j'=1}^{J'} \frac{(\omega'_{j'})^2}{2} + \frac{C}{n}\sum_{i=1}^{n} \xi_i \tag{1}$$

s.t. (SVM)

$$y_i(\sum_{j=1}^{J}\sum_{k=1}^{K_j} \omega_{j,k}\, x_{i,j,k} + (\omega')^\top x'_i + b) \geq 1 - \xi_i \qquad \forall i = 1,\ldots,n \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1,\ldots,n \tag{3}$$

$$\omega \in \mathbb{R}^{\sum_{j=1}^{J} K_j} \tag{4}$$

$$\omega' \in \mathbb{R}^{J'} \tag{5}$$

$$b \in \mathbb{R}, \tag{6}$$

where $(\xi_i)$ denotes the vector of deviation variables and the parameter denoted by $C$ is a nonnegative regularization parameter that calls for tuning, [6, 8].

The methodology proposed in this paper, the CLSVM, receives as input a dataset containing categorical and eventually continuous features. We will denote by $L_j$ the number of clusters in which the $K_j$ dummies of categorical feature $j$ are clustered. As a first step, the CLSVM performs a clustering for each categorical feature, defined by an assignment vector $z^*$, where $z^*_{j,k,\ell}$ is equal to 1 if category $k$ from feature $j$ is assigned to the $\ell$-$th$ cluster and 0 otherwise, for $j = 1, \ldots, J,\ k = 1, \ldots, K_j,\ \ell = 1, \ldots, L_j$. Then, the dataset is clustered according to $z^*$, see Figure 1, and an SVM-type classifier is constructed for the clustered dataset, given by $(\bar{\omega})^\top \bar{x} + (\omega')^\top x' + b = 0$. For categorical feature $j$, the component $\bar{\omega}_{j,\ell}$ denotes the score for the $\ell$-$th$ cluster, $j = 1, \ldots, J,\ \ell = 1, \ldots, L_j$. The pseudocode of the CLSVM methodology can be found in Figure 2. To avoid symmetry between clustering solutions, the first category of each categorical feature is always assigned to its first cluster.

For each $i \in \Omega$:

**Step 1.** Input:

- original object $(y_i, x_i, x'_i)$, $x_i \in \{0,1\}^{\sum_{j=1}^{J} K_j}$, $x'_i \in \mathbb{R}^{J'}$.
- assignment vector $z^* \in \{0,1\}^{\sum_{j=1}^{J} L_j K_j}$
  where $\sum_{\ell=1}^{L_j} z^*_{j,k,\ell} = 1, \ \forall j = 1, \ldots, J; \forall k = 1, \ldots, K_j$.

**Step 2.** Output:

- clustered object $(y_i, \bar{x}_i, x'_i)$, $\bar{x}_i \in \{0,1\}^{\sum_{j=1}^{J} L_j}$, $x'_i \in \mathbb{R}^{J'}$
  where $\bar{x}_i = (\bar{x}_{i,1,1}, \ldots, \bar{x}_{i,J,L_J})$ with $\bar{x}_{i,j,\ell} = \sum_{k=1}^{K_j} z^*_{j,k,\ell} x_{i,j,k}$.

**Figure 1:** Pseudocode for the clustered dataset defined by the assignment vector $z^*$.

Given a dataset $\Omega$:

**Step 1.** Find the assignment vector $z^*$, defining a clustering for the categorical features.

**Step 2.** Obtain the clustered dataset $\bar{\Omega}$ as in Figure 1.

**Step 3.** Find the *CLSVM* classifier for $\bar{\Omega}$, $(\bar{\omega})^\top \bar{x} + (\omega')^\top x' + b = 0$.

**Figure 2:** Pseudocode for the CLSVM methodology.

## 2.1 Formulations for the CLSVM

In this section two different mathematical optimization formulations are proposed for the CLSVM methodology, an MINLP and an MIQP formulations. The MIQP formulation is derived from the MINLP formulation using the *big M* modeling trick to reformulate the nonlinear terms in the feasible region.

First, we introduce the Cluster (CL) formulation, an MINLP formulation with nonlinear constraints and 0-1 decision variables. This formulation aims at finding a classifier, but at the same time clustering categorical feature $j$ into $L_j$ clusters, for each $j = 1, \ldots, J$. The CL is formulated as follows:

$$\min_{\bar{\omega},\omega',b,\xi,z} \sum_{j=1}^{J}\sum_{\ell=1}^{L_j} \frac{(\bar{\omega}_{j,\ell})^2}{2} + \sum_{j'=1}^{J'} \frac{(\omega'_{j'})^2}{2} + \frac{C}{n}\sum_{i=1}^{n} \xi_i \tag{7}$$

s.t. (CL)

$$y_i \left( \sum_{j=1}^{J}\sum_{\ell=1}^{L_j} \bar{\omega}_{j,\ell} \sum_{k=1}^{K_j} z_{j,k,\ell}\, x_{i,j,k} + (\omega')^\top x'_i + b \right) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{8}$$

$$\sum_{\ell=1}^{L_j} z_{j,k,\ell} = 1 \qquad \forall j = 1, \ldots, J; \forall k = 1, \ldots, K_j \tag{9}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{10}$$

$$z \in \{0,1\}^{\sum_{j=1}^{J} L_j K_j} \tag{11}$$

$$\bar{\omega} \in \mathbb{R}^{\sum_{j=1}^{J} L_j} \tag{12}$$

$$\omega' \in \mathbb{R}^{J'} \tag{13}$$

$$b \in \mathbb{R}. \tag{14}$$

This formulation resembles the SVM formulation (1)-(6), and we will discuss their main differences. Here we have a score associated with each feature and each cluster, $\bar{\omega}_{j,\ell}$, as opposed to a score for each category, $\omega_{j,k}$. With respect to the decision variables, we have $\sum_{j=1}^{J} L_j K_j$ new 0-1 variables, the number of components of the assignment vector $z$, but the number of continuous features associated with the score vector decreases from $\sum_{j=1}^{J} K_j$ to $\sum_{j=1}^{J} L_j$. Constraint (8) corresponds to constraint (2). Constraint (9) ensures that, given a categorical feature, each category is assigned to a unique cluster, which means that there are $\sum_{j=1}^{J} K_j$ additional constraints to those in the SVM formulation.

We will say that a categorical feature $j$ is irrelevant to the classifier if $\bar{\omega}_{j,\ell} = 0,\ \forall \ell = 1, \ldots, L_j$. On the contrary, if the feature is relevant to the classifier, we will say that cluster $\ell$ from feature $j$ points towards the positive class if the score associated to the cluster is positive, i.e., if $\bar{\omega}_{j,\ell} > 0$. Analogously, if $\bar{\omega}_{j,\ell} < 0$ we will say that cluster $\ell$ from feature $j$ points towards the negative class. The effective use of the clusters by the CL formulation is stated in the following theoretical results.

**Proposition 2.1** *For any optimal solution of CL, given a categorical feature $j^*$, if there exists $\ell^*$ such that $z_{j^*,k,\ell^*} = 1\ \forall k = 1, \ldots, K_{j^*}$, then $\bar{\omega}_{j^*,\ell} = 0\ \forall \ell = 1, \ldots, L_{j^*}$.*

*Proof:* The proposition will be proved by contradiction. Let $(\bar{\omega}, \omega', b, \xi, z)$ be an optimal solution of CL for which the desired property does not hold. For the case $\ell = \ell^*$, if $\bar{\omega}_{j^*,\ell^*} \neq 0$, then $(\bar{\omega}^*, \omega'^*, b^*, \xi^*, z^*)$ obtained by setting $\bar{\omega}^*_{j^*,\ell^*} = 0$ and $b^* = b + \bar{\omega}_{j^*,\ell^*}$ is a feasible solution for (7)-(14) and has a smaller objective value, which contradicts the fact that the solution $(\bar{\omega}, \omega', b, \xi, z)$ is optimal.

Now we analyze the case $\ell \neq \ell^*$. If $\bar{\omega}_{j^*,\ell} \neq 0$, then $(\bar{\omega}^*, \omega'^*, b^*, \xi^*, z^*)$ obtained by setting $\bar{\omega}^*_{j^*,\ell} = 0$ is a feasible solution for (7)-(14) and has a smaller objective value, which contradicts the fact that the solution $(\bar{\omega}, \omega', b, \xi, z)$ is optimal. □

From this proposition, we obtain:

**Corollary 2.1** *Given a categorical feature, if all its categories belong to the same cluster, then the feature is irrelevant to the CLSVM classifier.*

The clustering given in the CL formulation for a categorical feature $j$ with $L_j = 2$, groups the categories into two clusters. It is easy to see that either the feature is irrelevant or one of the clusters of the feature points towards the positive class while the other points towards the negative one.

**Proposition 2.2** *If $L_j = 2$, for a given $j$, for any optimal solution of CL, it holds that:*

$$\bar{\omega}_{j,1} \cdot \bar{\omega}_{j,2} \leq 0. \tag{15}$$

*Proof:* The proposition will be proved by contradiction. Let $(\bar{\omega}, \omega', b, \xi, z)$ be an optimal solution of CL for which the desired property does not hold, i.e., $\bar{\omega}_{j,1} \cdot \bar{\omega}_{j,2} > 0$. Then $(\bar{\omega}^*, \omega'^*, b^*, \xi^*, z^*)$ obtained by setting $\bar{\omega}^*_{j,1} = \frac{\bar{\omega}_{j,1} - \bar{\omega}_{j,2}}{2}$, $\bar{\omega}^*_{j,2} = \frac{\bar{\omega}_{j,2} - \bar{\omega}_{j,1}}{2}$ and $b^* = b + \frac{\bar{\omega}_{j,1} + \bar{\omega}_{j,2}}{2}$ satisfies (15), is a feasible solution for (7)-(14) and has a smaller objective value, which contradicts the fact that the solution $(\bar{\omega}, \omega', b, \xi, z)$ is optimal. □

Figure 8 of dataset `german`, mentioned in Section 1, illustrates the applicability of Proposition 2.2, where the clustering gives the additional information of which cluster points towards the positive class or the negative class. We have assigned a dark gray coloring to clusters in which $\bar{\omega}_{j,\ell} > 0$ in the CLSVM classifier, and therefore, those clusters point towards *good* customers; similarly, a light gray coloring is assigned to clusters in which $\bar{\omega}_{j,\ell} < 0$ in the CLSVM classifier, and therefore, those clusters point towards *bad* customers. For the four categories of feature "Property", the two clusters are given by {"real estate", "building society savings agreement/life insurance", "car or other"} and {"unknown/no property"}. The categories of the first cluster point towards the positive class, i.e., they are likely to be associated with *good* customers, while the category "unknown/no property" points towards the negative class, i.e., *bad* customers.

Nonconvex nonlinear constraints such as (8) are known to be computationally difficult to deal with, e.g. [22]. Therefore, one may want to reformulate constraint (8) from the MINLP formulation in order to obtain an MIQP formulation where the nonlinear term of the product of variables $\bar{\omega}_{j,\ell} \sum_{k=1}^{K_j} z_{j,k,\ell}\, x_{i,j,k}$ in constraint (8) is reformulated by introducing new *big M* constraints. This implies adding $\sum_{j=1}^{J} L_j K_j$ continuous variables, $\tilde{\omega}_{j,k,\ell},\ j = 1, \ldots, J,\ k = 1, \ldots, K_j,\ \ell = 1, \ldots, L_j$, yielding

$$\min_{\bar{\omega}, \tilde{\omega}, \omega', b, \xi, z} \sum_{j=1}^{J} \sum_{\ell=1}^{L_j} \frac{(\bar{\omega}_{j,\ell})^2}{2} + \sum_{j'=1}^{J'} \frac{(\omega'_{j'})^2}{2} + \frac{C}{n} \sum_{i=1}^{n} \xi_i \tag{16}$$

s.t. (CL-bigM)

$$y_i \left( \sum_{j=1}^{J} \sum_{\ell=1}^{L_j} \tilde{\omega}_{j,k(i),\ell} + (\omega')^\top x'_i + b \right) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{17}$$

$$\sum_{\ell=1}^{L_j} z_{j,k,\ell} = 1 \qquad \forall k = 1, \ldots, K_j, \ \forall j = 1, \ldots, J \tag{18}$$

$$\tilde{\omega}_{j,k,\ell} \leq \bar{\omega}_{j,\ell} + M(1 - z_{j,k,\ell}) \qquad \forall k = 1, \ldots, K_j, \ \forall \ell = 1, \ldots, L_j, \ \forall j = 1, \ldots, J \tag{19}$$

$$\tilde{\omega}_{j,k,\ell} \geq \bar{\omega}_{j,\ell} - M(1 - z_{j,k,\ell}) \qquad \forall k = 1, \ldots, K_j, \ \forall \ell = 1, \ldots, L_j, \ \forall j = 1, \ldots, J \tag{20}$$

$$\tilde{\omega}_{j,k,\ell} \leq M \, z_{j,k,\ell} \qquad \forall k = 1, \ldots, K_j, \ \forall \ell = 1, \ldots, L_j, \ \forall j = 1, \ldots, J \tag{21}$$

$$\tilde{\omega}_{j,k,\ell} \geq -M \, z_{j,k,\ell} \qquad \forall k = 1, \ldots, K_j, \ \forall \ell = 1, \ldots, L_j, \ \forall j = 1, \ldots, J \tag{22}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{23}$$

$$z \in \{0,1\}^{\sum_{j=1}^{J} L_j K_j} \tag{24}$$

$$\bar{\omega} \in \mathbb{R}^{\sum_{j=1}^{J} L_j} \tag{25}$$

$$\omega' \in \mathbb{R}^{J'} \tag{26}$$

$$\tilde{\omega} \in \mathbb{R}^{\sum_{j=1}^{J} L_j K_j} \tag{27}$$

$$b \in \mathbb{R}. \tag{28}$$

We will compare this with the CL formulation. Both objective functions are exactly the same. The difference between the two formulations comes from the constraints, and the addition of new variables, $\sum_{j=1}^{J} L_j K_j$ new continuous variables. Constraint (17) is the corresponding to constraint (8). Here, the nonlinear term is replaced with the variable $\tilde{\omega}_{j,k(i),\ell}$, where $k(i)$ identifies the category in which object $i$ falls for feature $j$. In order to reformulate constraint (8) as a collection of linear constraints, it is a very well-known modeling trick to use a 0-1 variable to control if constraint (8) is active or not, see [27]. Then, constraint (8) is reformulated as linear constraint (17), and $4 \cdot \sum_{j=1}^{J} L_j K_j$ more constraints are needed for the reformulation, (19)-(22), the so-called *big M* constraints.

Please note that Proposition 2.1, Proposition 2.2 and Corollary 2.1 also hold for the CL-bigM formulation, as it is a valid reformulation of the CL formulation.

# 3 Strategies for the CLSVM

In this section four different strategies are proposed to obtain the CLSVM classifier. The first, and natural, way to define a CLSVM classifier is by clustering the categories using the scores of the original SVM, the SVM$^{\mathrm{O}}$. This is a cheap strategy but underperforming in some cases in terms of accuracy, as we will see in the computational section. Three alternative strategies are proposed based on the two mathematical optimization formulations introduced in Section 2, the CL and the CL-bigM.

In the remainder of this section, when describing the strategies, we will explain how to obtain the partial solution $(\bar{\omega}, \omega', b)$, which determines the CLSVM classifier, and the assignment vector $z^*$. Then, the assignment vector $z^*$ performs a clustering for the original dataset, obtaining a clustered dataset, as shown in Figure 1.

The first strategy, the *centroid SVM* ($\text{SVM}^{\text{C}}$) Strategy, is based on the $\text{SVM}^{\text{O}}$ scores. As initial step, the $\text{SVM}^{\text{O}}$ classifier is built for the original dataset, then the categories of categorical feature $j$ are clustered into $L_j$ clusters by clustering the $\text{SVM}^{\text{O}}$ scores, for each $j$. This is done by solving the minimum sum of squares clustering problem (MSSC), [15]. Given a categorical feature $j$, MSSC clusters all the categories into $L_j$ clusters such that the sum of the squared distance of the score of a category from the centroid of the cluster is minimized. The pseudocode of the MSSC problem can be found in Figure 3, where the $j$ index has been dropped for the sake of clarity, and calligraphic font is used to denote sets, while regular font for their cardinality. After clustering the dataset, the CLSVM classifier builds an SVM classifier using the clustered dataset. The pseudocode of this strategy can be found in Figure 4.

Given a vector $\nu \in \mathbb{R}^K$ associated with a categorical feature with $K$ categories,

**Step 1.** Sort the values $\nu_k$, $k = 1, \ldots, K$, increasingly, such that:

$$\nu_{(1)} \leq \nu_{(2)} \leq \ldots \leq \nu_{(K)}$$

**Step 2.** Set $\mathcal{I}_{i_0} = \emptyset$
Set $\mathcal{I}_{i_\ell} = \{1, \ldots, i_\ell\}$
Set $\mathcal{I}_{i_L} = \{1, \ldots, K\}$
Set $F(\mathcal{I}_{i_\ell}) = \sum_{i \in \mathcal{I}_{i_\ell}} \left( \nu_{(i)} - \frac{1}{I_{i_\ell}} \sum_{h \in \mathcal{I}_{i_\ell}} \nu_{(h)} \right)^2$
Solve

$$\min_{i_1, \ldots, i_{L-1}} \sum_{\ell=1}^{L} F(\mathcal{I}_{i_\ell} \setminus \mathcal{I}_{i_{\ell-1}})$$

**Step 3.** Return the assignment vector $z^*$ defined by the $L$ clusters, determined by $\mathcal{I}_{i_1}, \ldots, \mathcal{I}_{i_{L-1}}$.

**Figure 3:** Pseudocode for the MSSC problem.

For each $C$,

**Step 1.** Solve the *SVM*$^{\text{O}}$ and obtain the (partial) optimal solution $\omega$.

**Step 2.** For each $j$, cluster the $K_j$ categories of feature $j$ into $L_j$ clusters solving the MSSC problem for $\nu = \omega_{j\cdot}$, obtaining the components from the assignment vector $z^*_{j\cdot\cdot}$.

**Step 3.** Solve the *SVM* formulation with clustered data defined by $z^*$, and return this as the CLSVM classifier.

Choose the best $C$ using the CLSVM classifier.

**Figure 4:** Pseudocode for the $\text{SVM}^{\text{C}}$ Strategy.

The second strategy, the *CL randomized rounding* ($\text{CL}^{\text{RR}}$) Strategy, performs a randomized rounding, [19], to the fractional assignment vector returned by the

continuous relaxation of the CL formulation. This is a QCQP formulation, where constraint (11) is relaxed to $z \in [0,1]^{\sum_{j=1}^{J} L_j K_j}$. The pseudocode of this reduction strategy can be found in Figure 5, where `rand`$(p)$ is a subroutine of random numbers generation, returning the value 1 with probability $p$ and 0 otherwise.

```
For each C,
Step 1. (i) Solve the continuous relaxation of CL and obtain the (partial) optimal
            solution z.
        (ii) Set z*_{j,k,ℓ} = 0  ∀k = 1,...,K_j, ∀ℓ = 1,...,L_j, ∀j = 1,...,J
             For j = 1,...,J
                    For k = 1,...,K_j
                           Set ℓ = 1
                           while (ℓ < L_j)
                                  Set z*_{j,k,ℓ} = rand(z_{j,k,ℓ})
                                  If z*_{j,k,ℓ} = 0, set ℓ = ℓ + 1
                                  Else ℓ = L_j
                           end
                           Set z*_{j,k,L_j} = 1 − Σ_{ℓ=1}^{L_j−1} z*_{j,k,ℓ}
                    end
             end
       (iii) Return the assignment vector z*.
Step 2. Solve the SVM formulation with clustered data defined by z*, and return this
        as the CLSVM classifier.

Choose the best C using the CLSVM classifier
```

**Figure 5:** Pseudocode for the $\text{CL}^{\text{RR}}$ Strategy.

The third strategy, the *CL-bigM randomized rounding* ($\text{CLM}^{\text{RR}}$) Strategy is based on the randomized rounding of the partial solution of the continuous relaxation of the CL-bigM formulation. It is similar to the $\text{CL}^{\text{RR}}$ Strategy, but with the difference that it solves the continuous relaxation of the CL-bigM formulation, where constraint (24) is relaxed to $z \in [0,1]^{\sum_{j=1}^{J} L_j K_j}$. The pseudocode of this strategy can be found in Figure 6.

The last strategy, the *CLM* Strategy, is based on the CL-bigM formulation. Instead of solving the continuous relaxation, this strategy solves the CL-bigM formulation or returns the incumbent solution after a given time limit. In this case the incumbent solution gives the clustering and the classifier at once. The pseudocode of this strategy can be found in Figure 7. This is the most computationally expensive strategy, as it involves solving an MIQP formulation with *big M* constraints. However, the cost of the strategy is balanced with the computational results, as shown in Section 4.

Other strategies are possible and natural, and some were tested. For instance, we tried two strategies based on solving the CL formulation. These strategies solved to optimality the CL formulation or returned the incumbent solution after a given time limit. We tested the strategy for which the incumbent solution gave the clustering

For each $C$,

**Step 1.** (i) Solve the continuous relaxation of *CL-bigM* and obtain the (partial) optimal solution $z$.

(ii) Set $z^*_{j,k,\ell} = 0 \quad \forall k = 1, \ldots, K_j, \forall \ell = 1, \ldots, L_j, \forall j = 1, \ldots, J$
For $j = 1, \ldots, J$
For $k = 1, \ldots, K_j$
Set $\ell = 1$
while $(\ell < L_j)$
Set $z^*_{j,k,\ell} = \texttt{rand}(z_{j,k,\ell})$
If $z^*_{j,k,\ell} = 0$, set $\ell = \ell + 1$
Else $\ell = L_j$
end
Set $z^*_{j,k,L_j} = 1 - \sum_{\ell=1}^{L_j - 1} z^*_{j,k,\ell}$
end
end

(iii) Return the assignment vector $z^*$.

**Step 2.** Solve the *SVM* formulation with clustered data defined by $z^*$, and return this as the CLSVM classifier..

Choose the best $C$ using the CLSVM classifier

**Figure 6:** Pseudocode for the CLM$^{\text{RR}}$ Strategy.

For each $C$,

**Step 1.** Solve the *CL-bigM* and obtain the (partial) solution $(\bar{\omega}, \omega', b, z)$, the assignment vector and the classifier at once, and return this as the CLSVM classifier.

Choose the best $C$ using the CLSVM classifier

**Figure 7:** Pseudocode for the CLM Strategy.

and the classifier at once. We also tested another one for which the assignment vector $z^*$ of the incumbent solution was used to cluster the dataset and an SVM was solved to find the classifier. These strategies are however computationally expensive as they involve solving MINLP formulations. The performance of these strategies is not reported in Section 4 since they were systematically outperformed by the strategies above.

# 4 Computational results

In this section we illustrate the performance of the CLSVM methodology compared to the benchmark procedure, the SVM$^{\text{O}}$, in terms of accuracy and complexity of the classifier associated with the categorical features. The accuracy of a classifier on a given dataset is defined as the percentage of objects correctly classified by the classifier on such dataset. The second criterion, complexity, quantifies (in

percentage) the fraction of relevant dummies of the score vector associated with the categorical features. In other words, the complexity of the SVM$^{\mathrm{O}}$ classifier is given by $\frac{card(\{\omega_j \neq 0\})}{\sum_{j=1}^{J} K_j} \cdot 100\%$ and the complexity of the CLSVM classifier is given by $\frac{card(\{\bar{\omega}_{j,\ell} \neq 0\})}{\sum_{j=1}^{J} K_j} \cdot 100\%$. We will show that the CLSVM classifier is competitive against the SVM$^{\mathrm{O}}$ classifier in terms of accuracy and outperforms the SVM$^{\mathrm{O}}$ classifier in terms of complexity.

Our experiments have been conducted on a PC with an Intel® Core™ i7 processor with 16 Gb of RAM for all strategies except for the CL$^{\mathrm{RR}}$ Strategy, where the Neos Server is used, [11]. We use the optimization engine CPLEX, [16], for solving the SVM formulation, the CL-bigM formulation and its continuous relaxation, and Ipopt, [25, 11], for the continuous relaxation of CL. We have fixed $M$=1000 on the CL-bigM formulation. Although most optimization problems are solved to optimality in a few seconds, for the CL-bigM formulation the time limit is set to 300 seconds.

As customary in Supervised Classification, the optimization of the SVM and the CLSVM calls for tuning some parameters, namely the tradeoff parameter $C$, see Figures 4-7. As usually done in the literature, the tuning procedure works as follows, e.g. [6, 8]. The dataset is split into three sets, the so-called training, testing and validation sets. For each value of $C$, the optimization problem is solved on the training set. The different classifiers built in this way are compared according to their accuracy on the testing set. The parameter $C$ with the highest accuracy on the testing set is chosen, and its accuracy on the validation set is reported. Following the usual approach, the parameter $C$ is tuned by inspecting a grid of the form $\frac{C}{n} \in \{10^{-6}, \ldots, 10^{6}\}$, see [8].

The remainder of this section is structured as follows. The datasets used to compare the CLSVM classifier are described in Section 4.1, and the computational results are presented in Section 4.2.

## 4.1 Datasets

The performance in terms of accuracy and complexity of the CLSVM methodology is illustrated using ten real-life datasets from the UCI repository, [3]. Regression datasets are transformed into 2-class classification datasets using the median (`abalone`), and multi-class datasets are transformed into 2-class ones, treating the largest class as the positive class and the remaining ones as the negative class (`careval`, `solar-c`, `molecular`). Recall that categorical features have been transformed by splitting the categories into 0-1 dummy features.

A description of these datasets can be found in Table 1, whose first three columns report the dataset name, full name given in the repository and total size of the dataset ($|\Omega|$). The size of the training set ($n$) is set as the closest $10^2$ multiple to $|\Omega|/3$ setting 5000 as the maximum in order to have running times below reasonable values, see fourth column of Table 1. The remaining records in the dataset are equally split between the testing and validation sets. The fifth column reports the class split in the training set and the sixth and seventh columns show the number of categorical and continuous features, respectively. Finally, the total number of

categories and the number of categories per feature are reported.

To obtain sharp estimates for the accuracy and the complexity, repeated random subsampling is used, where ten instances are run for each dataset. The ten instances differ in the seed used to reshuffle the dataset in order to obtain different training, testing and validation sets.

## 4.2 Results

In this section we compare the performance of the four strategies proposed to build the CLSVM classifier against that of the $SVM^O$ classifier in terms of accuracy and complexity of the classifier. When, for a given criterion, the difference in performance of two classifiers is below 1 percentage point (p.p.), we will say that both classifiers are comparable under such criterion.

Tables 2-5 report the mean validation accuracy as well as the standard deviation and the median across the ten reshuffles for the accuracy and complexity, where for each dataset and each criterion, we underline the best results accross all the strategies and the benchmark procedure. Results for the benchmark procedure, $SVM^O$, are reported in Table 2, for the $SVM^C$ Strategy in Table 3, for the $CL^{RR}$ Strategy in Table 4, and for the $CLM^{RR}$ and the CLM strategies in Table 5. The following conclusions can be drawn from our computational results for the mean values, but similar conclusions are derived if median values are analyzed.

We start with the accuracy. For seven datasets (`census income`, `mushrooms`, `coil 2000`, `abalone`, `molecular`, `solar-c`, `german`), at least one of the strategies is comparable to the $SVM^O$. For two datasets the $SVM^O$ is outperformed, by two strategies in `adult` and by one strategy in `australian`. In `adult`, the $SVM^C$ Strategy and the $CLM^{RR}$ Strategy outperform the $SVM^O$ by 3.65 p.p. and 4.18 p.p. respectively. In `australian`, the CLM Strategy outperforms the $SVM^O$ in 1.26 p.p. For one dataset, `careval`, the $SVM^O$ achieves the best accuracy, where the difference with the CLSVM classifier is between 2.57 p.p., with the CLM Strategy, and 13.94 p.p., with the $SVM^C$ Strategy.

We now focus on the second criterion, namely, complexity. All strategies show a dramatic reduction on complexity of the classifier with respect to the categorical features. The minimum improvement over the $SVM^O$ is for the `coil 2000` dataset, of 8.12 p.p. For the remaining datasets, all strategies proposed for the CLSVM methodology outperform the $SVM^O$ at least by 30 p.p. For the first six datasets, (`census income`, `adult`, `mushrooms`, `coil 2000`, `abalone`, `molecular`), the CLM Strategy achieves the lowest complexity. For the last four datasets (`careval`, `solar-c`, `german`, `australian`), the $CLM^{RR}$ Strategy achieves the lowest complexity, reaching an improvement of 85.25 p.p. over the $SVM^O$.

In summary, the four strategies proposed for the CLSVM methodology are competitive against the $SVM^O$ in terms of accuracy, and clearly dominate in terms of complexity of the classifier. The $SVM^C$ and $CLM^{RR}$ strategies, have a computational cost comparable to that of the benchmark procedure, $SVM^O$, as they only involve solving QP formulations. Then, for a small increase in the computational cost, one can obtain a more stable strategy, the $CL^{RR}$, solving QCQP formulations. Although the CLM Strategy is the most computationally expensive strategy, as it involves solv-

ing difficult MIQP formulations with *big M* constraints, its cost is balanced with the computational results, as it is the strategy performing best accuracy results in three datasets (`careval`, `german`, `australian`) and best complexity results in six datasets (`census income`, `adult`, `mushrooms`, `coil 2000`, `abalone`, `molecular`).

As shown in Table 5, the performance of the CLM Strategy suggests it could be improved for datasets with a large number of categories, such as `molecular`. Recall that to obtain running times below reasonable values, the time limit for this strategy is set to 300 seconds. Increasing the time limit to 3600 seconds for `molecular`, changes the mean accuracy from 51.92% to 93.70% and the median from 51.92% to 93.74%, which makes the CLM comparable to the $\text{SVM}^{\text{O}}$ in terms of accuracy for `molecular`. Therefore, increasing the running time may be an alternative for the CLM Strategy when dealing with a large number of features.

# 5 Conclusions

In this paper the CLSVM methodology is proposed, based on the SVM and performing a clustering for categorical features, letting categories cluster around their peers and building an SVM classifier using the clustered dataset. Four strategies are presented to build the CLSVM classifier by means of QCQP, MIQP and QP formulations. When using two clusters, the CLSVM classifier has a comparable accuracy to the $\text{SVM}^{\text{O}}$ classifier, in seven of the ten benchmark datasets. In the remaining three datasets, the CLSVM classifier outperforms the $\text{SVM}^{\text{O}}$ classifier in two datasets, and is outperformed in the other one. In terms of complexity of the classifier with respect to the categorical features, the CLSVM methodology shows a dramatic improvement over the $\text{SVM}^{\text{O}}$.

There are several interesting directions to extend the CLSVM methodology. First, knowledge domain [7, 18] can be incorporated into the methodology to build a set of comprehensible rules to facilitate interpretability. This can be done by adding new constraints to the formulations. For each categorical feature, the CLSVM creates a given number of clusters, hence, constraints implying that two categories must belong to the same cluster, or fixing the maximum (or minimum) number of categories that compose a cluster, can be easily added. Other natural constraints could contribute to interpretability. For instance, if categories are countries, one may want to impose some countries to be in the same cluster based on their geographic location.

Second, a sequential methodology could be designed to handle datasets containing a large number of categorical features. This can be done by running a CLSVM model for each feature, fixing a clustering for the feature, and then iteratively repeating the process for the remaining features. Different ways of choosing the order of features for the iterative process require extra analysis; for instance, one can choose the feature for which the CLSVM classifier has the best accuracy.

Third, the CLSVM methodology can be extended to handle continuous features as well. As the CLSVM aims at reducing the complexity of the classifier in the presence of categorical features, we have focused on benchmark datasets composed by categorical features and eventually continuous features. However, for any dataset, a

combined methodology could be performed in order to transform continuous features into categorical ones, by applying the techniques from [5, 20], either binarizing or discretizing continuous features and then applying the CLSVM methodology. This extension deserves further study and testing.

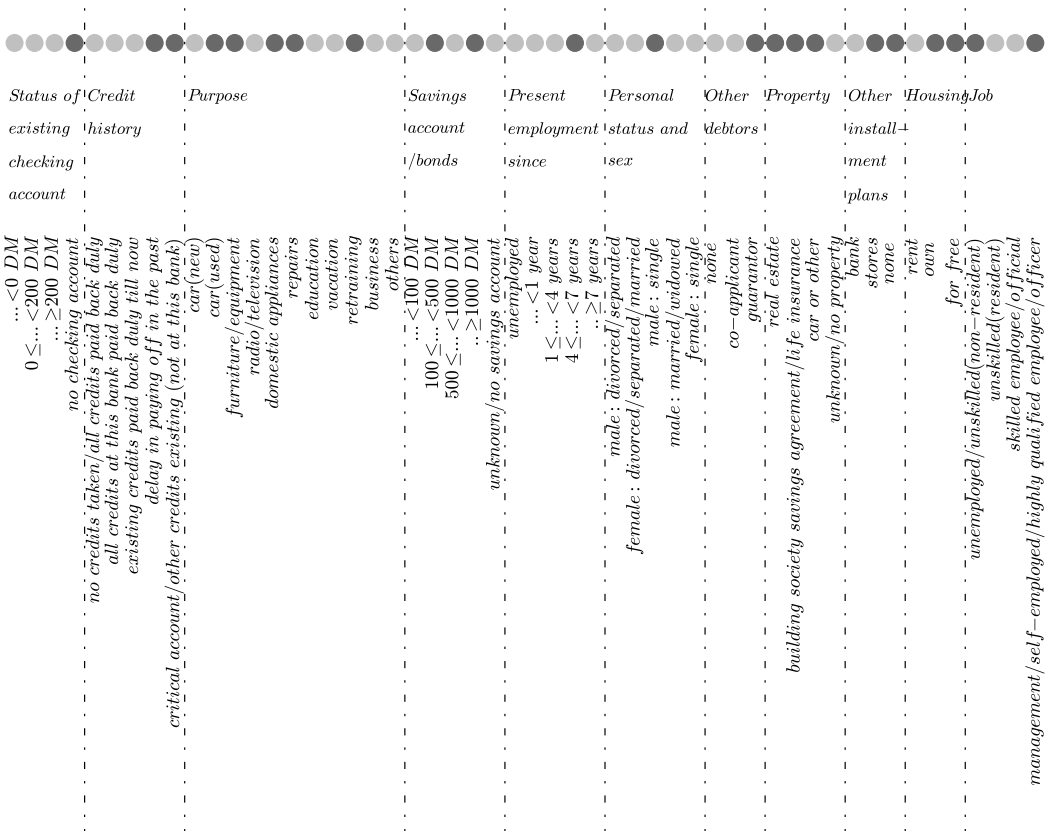


Figure 8: The CLSVM methodology for one instance of the `german` dataset.

Table 1: Datasets.

| Name | Name in Repository | $\|\Omega\|$ | $n$ | Class split (in %) | $J$ | $J'$ | $\sum_{j=1}^{J} K_j$ | $K_j$ |
|---|---|---|---|---|---|---|---|---|
| census income | Census-Income (KDD) Data Set | 95130 | 5000 | 94/6 | 31 | 9 | 491 | 9,52,47,17,3,7,24,15,5,10,3,6,8,6,6, 50,38,8,9,8,9,3,3,5,42,42,42,5,3,3,3 |
| adult | Adult | 30956 | 5000 | 24/76 | 11 | 3 | 117 | 5,8,5,16,5,7,14,6,5,5,41 |
| mushrooms | Mushrooms | 8124 | 5000 | 48/52 | 17 | 4 | 111 | 6,4,10,9,4,3,12,4,4,9,9,4,3,8,9,6,7 |
| coil 2000 | Insurance Company Benchmark (COIL 2000) | 5822 | 3900 | 94/6 | 5 | 80 | 77 | 41,6,10,10,10 |
| abalone | Abalone | 4177 | 2800 | 50/50 | 1 | 7 | 3 | 3 |
| molecular | Molecular Biology (Splice-junction Gene Sequences) | 3190 | 2200 | 52/48 | 60 | 0 | 480 | 8,8,8,... |
| careval | Car Evaluation | 1728 | 1200 | 30/70 | 6 | 0 | 21 | 4,4,4,3,3,3 |
| solar-c | Solar Flare Data Set | 1066 | 800 | 83/17 | 5 | 5 | 23 | 7,6,4,3,3 |
| german | Statlog (German Credit Data) | 1000 | 700 | 30/70 | 11 | 9 | 52 | 4,5,11,5,5,5,3,4,3,3,4 |
| australian | Statlog (Australian Credit Approval) | 690 | 500 | 56/44 | 4 | 10 | 29 | 3,14,9,3 |

Table 2: Accuracy and complexity results in % for the original SVM ($\text{SVM}^{\text{O}}$).

| *Name* | $\text{SVM}^{\text{O}}$ | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | | | Complexity | | |
| | mean | std | med | mean | std | med |
| census income | 94.90 | 0.00 | 94.90 | 63.14 | 0.00 | 63.14 |
| adult | 84.57 | 0.22 | 84.63 | 83.93 | 3.26 | 84.62 |
| mushrooms | 100.00 | 0.00 | 100.00 | 71.17 | 0.00 | 71.17 |
| coil 2000 | 100.00 | 0.00 | 100.00 | 1.30 | 0.00 | 1.30 |
| abalone | 79.87 | 1.18 | 79.72 | 100.00 | 0.00 | 100.00 |
| molecular | 94.22 | 0.80 | 94.04 | 57.04 | 3.12 | 58.54 |
| careval | 96.74 | 1.34 | 96.97 | 99.05 | 2.86 | 100.00 |
| solar-c | 83.53 | 1.23 | 83.46 | 52.17 | 34.51 | 69.57 |
| german | 74.60 | 2.71 | 75.66 | 94.62 | 1.88 | 95.19 |
| australian | 84.11 | 3.17 | 84.73 | 86.90 | 4.83 | 89.66 |

Table 3: Accuracy and complexity results in % for the $SVM^{C}$ Strategy.

| *Name* | $SVM^{C}$ | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | | | Complexity | | |
| | mean | std | med | mean | std | med |
| census income | 94.85 | 0.00 | 94.85 | 10.18 | 0.00 | 10.18 |
| adult | 88.22 | 2.44 | 89.59 | 13.76 | 2.21 | 13.68 |
| mushrooms | 100.00 | 0.00 | 100.00 | 23.42 | 0.00 | 23.42 |
| coil 2000 | 100.00 | 0.00 | 100.00 | 1.30 | 0.00 | 1.30 |
| abalone | 79.90 | 1.05 | 79.44 | 66.67 | 0.00 | 66.67 |
| molecular | 93.94 | 0.73 | 93.84 | 0.00 | 0.00 | 0.00 |
| careval | 82.80 | 5.15 | 82.95 | 44.76 | 8.57 | 38.10 |
| solar-c | 83.61 | 1.38 | 83.46 | 12.17 | 10.43 | 8.70 |
| german | 74.80 | 2.36 | 75.00 | 42.31 | 0.00 | 42.31 |
| australian | 84.42 | 3.32 | 84.73 | 19.31 | 9.66 | 24.14 |



Table 4: Accuracy and complexity results in % for the $CL^{RR}$ Strategy.

| *Name* | $CL^{RR}$ | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | | | Complexity | | |
| | mean | std | med | mean | std | med |
| census income | 94.84 | 0.04 | 94.82 | 8.31 | 0.33 | 8.55 |
| adult | 83.44 | 0.37 | 83.44 | 16.58 | 1.34 | 16.24 |
| mushrooms | 100.00 | 0.00 | 100.00 | 19.28 | 4.56 | 18.02 |
| coil 2000 | 100.00 | 0.00 | 100.00 | 1.30 | 0.00 | 1.30 |
| abalone | 79.86 | 1.02 | 79.44 | 66.67 | 0.00 | 66.67 |
| molecular | 93.40 | 1.28 | 93.53 | 24.87 | 0.19 | 25.00 |
| careval | 92.23 | 1.28 | 92.42 | 41.90 | 12.2 | 38.10 |
| solar-c | 83.83 | 1.08 | 83.46 | 5.22 | 11.14 | 0.00 |
| german | 74.60 | 3.12 | 74.00 | 38.27 | 4.59 | 41.35 |
| australian | 84.53 | 3.12 | 84.73 | 15.17 | 11.03 | 13.79 |

Table 5: Accuracy and complexity results in % for the CLM$^{RR}$ and the CLM strategies.

| *Name* | CLM$^{RR}$ | | | | | | CLM | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | | | Complexity | | | Accuracy | | | Complexity | | |
| | mean | std | med | mean | std | med | mean | std | med | mean | std | med |
| `census income` | 94.40 | 0.04 | 94.37 | 8.55 | 1.62 | 8.35 | 94.37 | 0.00 | 94.37 | 0.00 | 0.00 | 0.00 |
| `adult` | 88.75 | 2.96 | 89.63 | 10.17 | 4.36 | 8.55 | 85.35 | 3.16 | 83.44 | 9.23 | 3.64 | 9.83 |
| `mushrooms` | 98.58 | 0.77 | 98.59 | 21.71 | 5.87 | 22.07 | 100.00 | 0.00 | 100.00 | 14.77 | 1.57 | 14.41 |
| `coil 2000` | 100.00 | 0.00 | 100.00 | 1.30 | 0.00 | 1.30 | 100.00 | 0.00 | 100.00 | 1.30 | 0.00 | 1.30 |
| `abalone` | 79.87 | 0.96 | 79.66 | 66.67 | 0.00 | 66.67 | 79.65 | 1.25 | 79.29 | 66.67 | 0.00 | 66.67 |
| `molecular` | 88.04 | 1.68 | 88.38 | 22.71 | 0.90 | 22.50 | 51.92 | 0.00 | 51.92 | 0.00 | 0.00 | 0.00 |
| `careval` | 83.94 | 4.91 | 85.41 | 29.52 | 10.82 | 28.57 | 94.17 | 2.84 | 93.37 | 49.52 | 3.81 | 47.62 |
| `solar-c` | 83.76 | 1.02 | 83.46 | 0.87 | 2.61 | 0.00 | 83.61 | 1.38 | 83.46 | 11.31 | 7.83 | 8.70 |
| `german` | 72.53 | 3.77 | 71.66 | 36.92 | 4.28 | 36.54 | 75.60 | 3.01 | 76.34 | 42.31 | 0.00 | 42.31 |
| `australian` | 84.53 | 3.05 | 84.73 | 5.52 | 2.76 | 6.90 | 85.37 | 3.28 | 85.26 | 23.45 | 5.52 | 27.59 |

# References


[1] C. Apte. The big (data) dig. *OR/MS Today*, 30(1):24–29, February 2003.

[2] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, 49(3):312–329, 2003.

[3] C.L. Blake and C.J. Merz. UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998. University of California, Irvine, Department of Information and Computer Sciences.

[4] J.P. Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59(2):467–479, 2011.

[5] E. Carrizosa, B. Martín-Barragán, and D. Romero Morales. Binarized support vector machines. *INFORMS Journal on Computing*, 22(1):154–167, 2010.

[6] E. Carrizosa, B. Martín-Barragán, and D. Romero Morales. A nested heuristic for parameter tuning in support vector machines. *Computers and Operations Research*, 43:328–334, 2014.

[7] E. Carrizosa, A. Nogales-Gómez, and D. Romero Morales. Strongly agree or strongly disagree?: Rating features in Support Vector Machines. Working Paper, 2014.

[8] E. Carrizosa and D. Romero Morales. Supervised classification and mathematical optimization. *Computers and Operations Research*, 40:150–165, 2013.

[9] W. A. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo. Novel optimization models for abnormal brain activity classification. *Operations Research*, 56(6):1450–1460, 2008.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[11] J. Czyzyk, M.P. Mesnier, and J.J. More. The neos server. *IEEE Computational Science Engineering*, 5(3):68–75, 1998.

[12] G. Fung and O.L. Mangasarian. A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications*, 28(2):185–202, 2004.

[13] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[14] H. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, 2001.

[15] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, 1997.

[16] IBM-Cplex, v. 12.5. http://www-01.ibm.com/software/integration/optimization/cplex/.

[17] D. Martens, B. Baesens, T.V. Gestel, and J. Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3):1466–1476, 2007.

[18] D. Martens and F. Provost. Explaining data-driven document classifications. *MIS Quarterly*, 38(1):73–99, 2014.

[19] P. Raghavan and C. D. Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[20] D. Romero Morales and J. Wang. A parallel discretization algorithm for cancellation rate forecasting in revenue management. Working Paper, Saïd Business School, University of Oxford, UK, 2009.

[21] D. Romero Morales and J. Wang. Forecasting cancellation rates for services booking revenue management using data mining. *European Journal of Operational Research*, 202(2):554–562, 2010.

[22] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, Boston MA, 2002.

[23] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[24] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[25] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[26] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, pages 668–674, Cambridge, MA, 2000. MIT Press.

[27] H.P. Williams. *Model building in Mathematical Programming*. Wiley, New York, 1985.

[28] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2007.