Omega 66 (2017) 28–37

Contents lists available at ScienceDirect

Omega

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

Clustering categories in support vector machines ☆, ☆☆

Emilio Carrizosa [a], Amaya Nogales-Gómez [b,*,1], Dolores Romero Morales [c]

[a] Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla, Spain
[b] Mathematical and Algorithmic Sciences Lab, Huawei France R&D, France
[c] Department of Economics, Copenhagen Business School, Denmark

ARTICLE INFO

Article history:
Received 7 July 2014
Accepted 20 January 2016
Available online 3 February 2016

Keywords:
Support vector machine
Categorical features
Classifier sparsity
Clustering
Quadratically constrained programming
0-1 programming

ABSTRACT

The support vector machine (SVM) is a state-of-the-art method in supervised classification. In this paper the Cluster Support Vector Machine (CLSVM) methodology is proposed with the aim to increase the sparsity of the SVM classifier in the presence of categorical features, leading to a gain in interpretability. The CLSVM methodology clusters categories and builds the SVM classifier in the clustered feature space. Four strategies for building the CLSVM classifier are presented based on solving: the SVM formulation in the original feature space, 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 in the original feature space, with a dramatic increase in sparsity.

1. Introduction

In supervised classification [2,18,37], we are given a set of objects Ω 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. Thus, $x_i = (x_{i,j,k})$, 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. 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 [26], making it hard to both build the classifier and interpret the results. In this case, it is desirable to obtain a sparser 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 [14,15,17,35], 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 an 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 the support vector machine (SVM). The SVM aims at separating both classes by means of a classifier, $(\omega)^\top x + (\omega')^\top x' + b = 0$, (ω, ω') being the so-called score vector, where ω is associated with the categorical features and ω' 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 [5,11,17,24,29] for successful applications of the SVM and [10] for a recent review on Mathematical Optimization and the SVM.

In this paper, a methodology to increase the sparsity of the support vector machine (SVM) classifier for datasets composed by categorical features, sometimes containing many categories, and eventually continuous features, is proposed. This is done by

☆This manuscript was processed by Associate Editor Snyder.
☆☆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.
* Corresponding author.
E-mail addresses: ecarrizosa@us.es (E. Carrizosa), amaya.nogales.gomez@huawei.com (A. Nogales-Gómez), drm.eco@cbs.dk (D. Romero Morales).
[1] Most of this work was done when the author was at Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla.

http://dx.doi.org/10.1016/j.omega.2016.01.008
0305-0483/

clustering the different categories of each categorical feature into a given number of clusters, and then obtaining an SVM-type classifier in the clustered feature space. We call this the Cluster Support Vector Machine (CLSVM) methodology and we will refer to the CLSVM classifier. Note that we apply a clustering methodology to the feature space, while other papers in the literature such as [16] apply clustering to the set of records.

Sparsity is used as a surrogate of interpretability, since in sparse classifiers only the most valuable information is retained. As an illustration, let us consider the well-known German credit dataset, `german`, which is one of the datasets from the UCI repository, [4], 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 [3]. 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 in the original feature space, hereafter denoted by SVM^{O}, gives a classifier leading to a classification accuracy of 76.67% and whose categorical score subvector ω 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 without compromising accuracy (in this case, accuracy is even higher than the original one). The clustering of categories for `german` is shown in Fig. 6, 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, against the category indicating *bad* customers, namely "unknown/no property". This is a further gain in interpretability of the methodology proposed here when categories are grouped into two clusters, by detecting which clusters point towards the positive class.

In this paper, four strategies to build the CLSVM classifier are proposed using different mathematical optimization formulations. The first strategy proposed solves the SVM^{O} as initial step. Then, categories are clustered using a partition of the SVM^{O} scores and the CLSVM classifier consists of building an SVM classifier in the clustered feature space. 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 in the clustered feature space. 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^{O} in twelve real-life datasets using two performance criteria, namely accuracy and sparsity of the classifier for the categorical features. We conclude from our experiments that the CLSVM achieves a comparable or even better accuracy than the SVM^{O} in eleven of the twelve datasets tested. In addition, the CLSVM methodology shows an outstanding performance in terms of sparsity of the classifier for the categorical features, with SVM^{O} using many more dummy features than each of the strategies in ten of the twelve datasets.

The remainder of this paper is organized as follows. In Section 2, the CLSVM methodology is introduced together with two mathematical optimization formulations. Two theoretical results on relevance of features and interpretability 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^{O} classifier are compared using twelve datasets. Finally, Section 5 contains a brief summary, 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. 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. Two theoretical results on relevance of features and interpretability are shown for both formulations.

First, we present the standard SVM formulation [10,12,32,33]. 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 [10]. 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)

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

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

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

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

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

where (ξ_i) denotes the vector of deviation variables and the parameter denoted by C is a nonnegative regularization parameter that calls for tuning [7,10]. 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).

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

Given a dataset Ω:

Step 1. Find the assignment vector $z^* \in \{0,1\}^{\sum_{j=1}^{J} L_j K_j}$, defining the clustering of categories for the categorical features.

Step 2. For each object $i \in \Omega$, cluster the categories according to z^*, i.e.,

- consider (x_i, x_i', y_i), $x_i \in \{0,1\}^{\sum_{j=1}^{J} K_j}$, $x_i' \in \mathbb{R}^{J'}$,
- transform x_i into $\bar{x}_i$, with $\bar{x}_i \in \{0,1\}^{\sum_{j=1}^{J} L_j}$ and $\bar{x}_{i,j,\ell} = \sum_{k=1}^{K_j} z^*_{j,k,\ell} x_{i,j,k}$, and
- derive $(\bar{x}_i, x_i', y_i)$.

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

Fig. 1. Pseudocode for the CLSVM methodology.

each dummy feature, and thus to each value of the categorical feature. Instead, the CLSVM methodology clusters dummies and builds an SVM classifier in the clustered feature space, which may reduce the number of relevant features. The pseudocode of the CLSVM methodology can be found in Fig. 1. We denote by L_j the number of clusters in which the K_j dummies of categorical feature j are clustered, and $\overline{\omega}_{j,\ell}$ the score for the ℓ-th cluster of categorical feature j. In the first step, the CLSVM finds 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 ℓ-th cluster and 0 otherwise, for $j=1,\ldots,J$, $k=1,\ldots,K_j$, $\ell=1,\ldots,L_j$. In the second step, for $i \in \Omega$, and using z^*, x_i is transformed into $\bar{x}_i$. In the third step, an SVM-type classifier, $(\overline{\omega})^\top \bar{x} + (\omega')^\top x' + b = 0$, is constructed in the clustered feature space. To avoid symmetry between clustering solutions, the first category of each categorical feature is always assigned to its first cluster.

Note that in this paper we illustrate the CLSVM methodology with the standard SVM, but that ours is applicable to other SVM-type formulations, with *loss* functions others than the *hinge loss* (such as the ramp loss [5,8]) and regularization terms others than the l_2-norm (such as the l_1-norm [21,23]).

2.1. Formulations for the CLSVM

In this section two different mathematical optimization formulations are proposed for the CLSVM methodology, an MINLP formulation and an MIQP one. 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:

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

s.t. (CL)

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

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

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

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

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

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

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

This formulation resembles the SVM formulation (1)–(6), and we will discuss their main differences. Here we have a score associated with each categorical feature and each cluster, $\overline{\omega}_{j,\ell}$, 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.

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 ℓ^* such that $z_{j^*,k,\ell^*} = 1$ $\forall k = 1,\ldots,K_{j^*}$, then $\overline{\omega}_{j^*,\ell} = 0$ $\forall \ell = 1,\ldots,L_{j^*}$.*

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

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

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

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

Fig. 6 of dataset `german`, mentioned in Section 1, illustrates the applicability of Proposition 2.2. We have assigned a dark gray coloring to clusters in which $\overline{\omega}_{j,\ell} > 0$ in the CLSVM classifier, and therefore, those clusters point towards *good* customers; similarly, a light gray coloring is assigned to clusters in which $\overline{\omega}_{j,\ell} < 0$ in the CLSVM classifier, and therefore, those clusters point towards *bad* customers. For example, 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. [31]. 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 $\overline{\omega}_{j,\ell} \sum_{k=1}^{K_j} z_{j,k,\ell} x_{i,j,k}$ 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

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

s.t. (CL − bigM)

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

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

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

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

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

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

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

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

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

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

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

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

We now compare the CL-bigM and the CL formulations. Both objective functions are exactly the same. The difference between the two formulations comes from the constraints, and the addition of $\sum_{j=1}^{J} L_j K_j$ new continuous variables. Constraint (17) is as 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 categorical 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 [36]. Then, constraint (8) is reformulated as linear constraint (17), and $4 \cdot \sum_{j=1}^{J} L_j K_j$ additional constraints are needed for the reformulation, (19)–(22), the so-called *big M* constraints.

Note that Propositions 2.1 and 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 SVM in the original feature space, the SVM^{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 $(\overline{\omega}, \omega', b)$, which determines the CLSVM classifier, and the assignment vector z^*, defining the clustering for the categorical features and thus the clustered feature space, as shown in Fig. 1.

The first strategy, the *centroid SVM* (SVM^{C}) Strategy, is based on the SVM^{O} scores. The strategy is as follows. The SVM^{O} classifier is built, the categories of feature j are clustered into L_j clusters finding a partition of the SVM^{O} scores, for each j, and the SVM classifier built in the clustered feature space is returned as the CLSVM classifier. The pseudocode of this strategy can be found in Fig. 2. There, the partition of the SVM^{O} scores is found by solving the minimum sum of squares clustering (MSSC) problem, [19], which is polynomially solvable for one-dimensional data when the number of clusters is fixed [1,20,30]. Given a categorical feature j, the MSSC problem 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 SVM^{C} Strategy can be implemented using other partitions of the SVM^{O} scores instead of the one given by MSSC. For instance, one can use natural values to partition the scores, such as 0, placing the negative scores in the first cluster, the zero ones in the second cluster, and the remaining ones in the third cluster. Other natural values are the median score, yielding a partition into two clusters, or, more generally, percentiles of the scores.

The second strategy, the *CL randomized rounding* (CL^{RR}) Strategy, performs a randomized rounding [27], 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 Fig. 3, where `rand`(p) is a subroutine of random numbers generation, returning the value 1 with probability p and 0 otherwise.

Phase 1: For each C

Step 1. Solve the SVM^{O} and obtain the (partial) optimal solution ω.

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

Step 3. Solve the SVM formulation in the clustered feature space defined by z^*, and return this as the CLSVM classifier.

Phase 2: Choose the best C using the CLSVM classifiers in Phase 1.

Fig. 2. Pseudocode for the SVM^{C} Strategy.

Phase 1: For each C

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

(ii) Set $z^*_{j,k,\ell} = 0 \quad \forall k = 1,\dots,K_j, \forall \ell = 1,\dots,L_j, \forall j = 1,\dots,J$

For $j = 1,\dots,J$

For $k = 1,\dots,K_j$

Set $\ell = 1$

while $(\ell < L_j)$

Set $z^*_{j,k,\ell} = \text{rand}(z_{j,k,\ell})$

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 in the clustered feature space defined by z^*, and return this as the CLSVM classifier.

Phase 2: Choose the best C using the CLSVM classifiers in Phase 1.

Fig. 3. Pseudocode for the CL^{RR} Strategy.

The third strategy, the *CL-bigM randomized rounding* (CLM^{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 CL^{RR} Strategy, but with the difference that it solves the continuous relaxation of the CL-bigM formulation, where constraint (23) is relaxed to $z \in [0,1]^{\sum_{j=1}^{J} L_j K_j}$. The pseudocode of this strategy can be found in Fig. 4.

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. In this case we obtain the clustering and the classifier at once. The pseudocode of this strategy can be found in Fig. 5. 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. We tested the strategy for which the solution gave the clustering and the classifier at once. We also tested another one for which the assignment vector z^* of the 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^{O}, in terms of classification accuracy and sparsity of the classifier for the categorical features. We have chosen $L_j = 2$, for all $j = 1,\dots, J$, and therefore the dimension of the clustered categorical feature space is equal to $\sum_{j=1}^{J} L_j = 2J$.

The classification 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 is sparsity with respect to the original categorical feature space. The sparsity of the SVM^{O} classifier is given by

$$\frac{card(\{\omega_{j,k} = 0\})}{\sum_{j=1}^{J} K_j} \cdot 100\%,$$

which quantifies (in percentage) the fraction of irrelevant dummies of the score vector associated with the categorical features. The sparsity of the CLSVM classifier relative to the original

Phase 1: 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 $z^*_{j,k,\ell} = \text{rand}(z_{j,k,\ell})$
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 in the clustered feature space defined by z^*, and return this as the CLSVM classifier.

Phase 2: Choose the best C using the CLSVM classifiers in Phase 1.

Fig. 4. Pseudocode for the CLM^{RR} Strategy.

Phase 1: For each C

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.

Phase 2: Choose the best C using the CLSVM classifiers in Phase 1.

Fig. 5. Pseudocode for the CLM Strategy.

categorical feature space is the summation of two terms. First, we have the theoretical sparsity, i.e., the one gained by clustering K_j categories into L_j ones. Second, we have the sparsity gained by the zero scores in the clustered feature space. Thus, the sparsity of the CLSVM classifier can be written as

$$\left(1 - \frac{\sum_{j=1}^{J} L_j}{\sum_{j=1}^{J} K_j}\right) \cdot 100\% + \frac{card(\{\overline{\omega}_{j,\ell} = 0\})}{\sum_{j=1}^{J} K_j} \cdot 100\%. \tag{29}$$

We will show that the CLSVM classifier is competitive against the SVM^O classifier in terms of classification accuracy and outperforms the SVM^O classifier in terms of sparsity.

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^{RR} Strategy, where the Neos Server is used, [13]. We use the optimization engine CPLEX, [22], for solving the SVM formulation, the CL-bigM formulation and its continuous relaxation, and Ipopt, [34,13], 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 s, and thus the incumbent solution after such time limit is used instead.

As customary in supervised classification, building the SVM and the CLSVM classifiers calls for tuning the tradeoff parameter C, see Figs. 2–5. As usually done in the literature, the tuning procedure works as follows, e.g. [7,10]. 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 classification accuracy on the testing set. The parameter C with the highest classification accuracy on the testing set is chosen, and its classification accuracy on the validation set is reported. Following the usual approach, the parameter C is tuned by inspecting a grid of the form $\frac{C}{n} \in \left\{10^{-6}, \ldots, 10^{6}\right\}$, see [10].

To obtain sharp estimates for the classification accuracy and the sparsity, 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.

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 classification accuracy and sparsity of the CLSVM methodology is illustrated using twelve real-life datasets from the UCI repository [4]. 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 (`nursery`, `covertype`, `molecular`, `careval`, `solar-c`). Recall that categorical features

Table 1
Datasets.

Name	$\lvert\Omega\rvert$	n	Class split	J	J'	K_j	$\sum_{j=1}^{J} K_j$	Theoretical sparsity
census income	95 130	5000	94/6	31	9	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	491	83.37
adult	30 956	5000	24/76	11	3	5,8,5,16,5,7,14,6,5,5,41	117	81.20
nursery	12 960	5000	67/33	8	0	3,5,4,4,3,2,3,3	27	40.74
covertype	11 340	5000	57/43	2	10	4, 40	44	90.91
mushrooms	8124	5000	48/52	17	4	6,4,10,9,4,3,12,4,4,9,9,4,3,8,9,6,7	111	69.37
coil 2000	5822	3900	94/6	5	80	41,6,10,10,10	77	87.01
abalone	4177	2800	50/50	1	7	3	3	33.33
molecular	3190	2200	52/48	60	0	8,8,8,…	480	75.00
careval	1728	1200	30/70	6	0	4,4,4,3,3,3	21	42.86
solar-c	1066	800	83/17	5	5	7,6,4,3,3	23	56.52
german	1000	700	30/70	11	9	4,5,11,5,5,5,3,4,3,3,4	52	57.69
australian	690	500	56/44	4	10	3,14,9,3	29	72.41

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 two columns report the dataset name and total size of the dataset ($|\Omega|$). The size of the training set (n) is set as the closest 10^2 multiple to $\frac{2}{3}|\Omega|$ setting 5000 as the maximum in order to have running times below reasonable values, see third column of Table 1. The remaining records in the dataset are equally split between the testing and validation sets. The fourth column reports the class split in the training set. The next three columns show the number of categorical and continuous features, respectively, and the number of categories per feature. Finally, the last two columns report the total number of categories, i.e., the size of the original feature space related to the categorical variables, $\sum_{j=1}^{J} K_j$, and the theoretical sparsity of the CLSVM classifier, the first term in (29).

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 classification accuracy and sparsity 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 and 3 report the results for the benchmark procedure, SVM^{O}, and for the strategies proposed in this paper. Table 2 reports the mean accuracy in the validation set as well as the standard deviation across the ten reshuffles, and similar information is reported in Table 3 for the sparsity. For each dataset and each criterion, we underline the best results across all the strategies and the benchmark procedure. The following conclusions can be drawn from our computational results for the mean values.

We start with the accuracy, see Fig. 7. For nine of the twelve benchmark datasets (census income, nursery, covertype, 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. This improvement suggests that SVM^{O}, i.e., the SVM in the original feature space, is overfitting. 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, sparsity of the classifier with respect to the categorical features, see Fig. 8. The strategies show an outstanding performance in terms of sparsity. All the strategies and the SVM^{O} achieve the same sparsity for the coil 2000 dataset, namely, 98.70%. All except for the SVM^{C} Strategy outperform the SVM^{O} for the nursery dataset. All except for the SVM^{C} Strategy outperform the SVM^{O} for the covertype dataset in 65 p.p., while the SVM^{C} Strategy outperforms the SVM^{O} in 15 p.p. For the remaining nine datasets, all the strategies outperform the SVM^{O} by at least 30 p.p.

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 sparsity 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. Solving QCQP formulations, thus, incurring a small increase in the computational cost, one can obtain the CL^{RR} Strategy. Although the CLM Strategy is the most computationally expensive strategy, as it involves solving 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 six datasets (nursery, mushrooms, coil 2000, careval, german, australian) and best sparsity results in seven datasets (census income, adult, nursery, mushrooms, coil 2000, abalone, molecular).

As shown in Table 2, 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 s. Increasing the time limit to 3600 s for molecular, changes the mean accuracy from 51.92% to 93.70%, which makes the CLM comparable to the SVM^{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 with the linear kernel and performing a clustering for categorical features and building an SVM classifier in the clustered feature space. 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 classification accuracy to the SVM^{O} classifier, in nine of the twelve benchmark datasets. In the remaining three datasets, the CLSVM classifier outperforms the SVM^{O} classifier in two datasets, and is outperformed in the other one. In terms of sparsity of the classifier with respect to the categorical features, the CLSVM methodology shows a dramatic improvement over the SVM^{O}.

Knowledge domain [9,25] can easily be incorporated into the methodology by adding new constraints to the formulations. For instance, must link constraints [19], i.e., constraints implying that two categories must belong to the same cluster, or fixing the

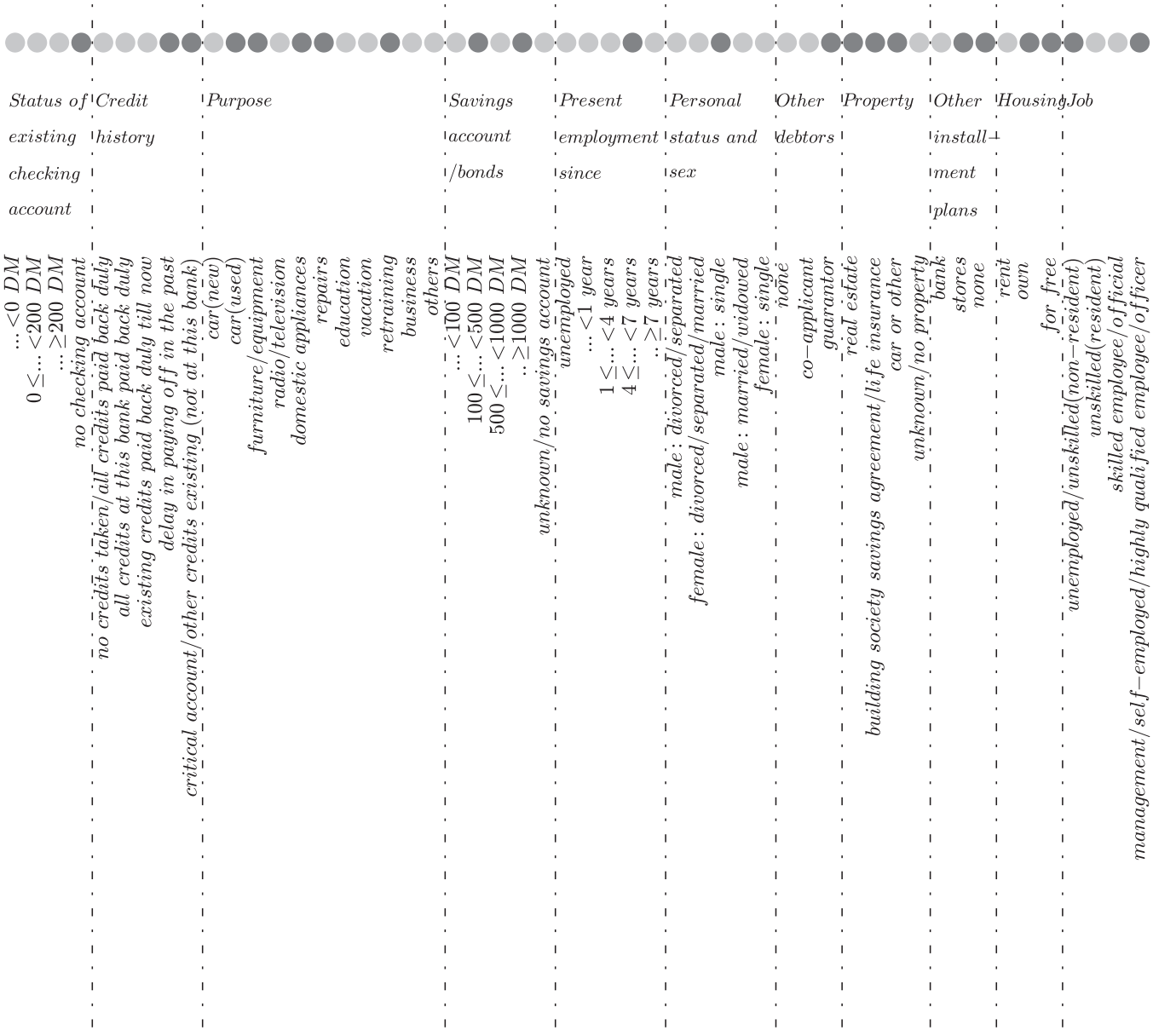

Fig. 6. The CLSVM methodology for one instance of the `german` dataset.

Table 2
Accuracy for the original SVM (SVM^O) and the CLSVM strategies.

Name	SVM^O		SVM^C		CL^{RR}		CLM^{RR}		CLM	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
census income	94.90	0.00	94.85	0.00	94.84	0.04	94.40	0.04	94.37	0.00
adult	84.57	0.22	88.22	2.44	83.44	0.37	88.75	2.96	85.35	3.16
nursery	100.00	0.00	67.98	4.56	96.67	10.00	100.00	0.00	100.00	0.00
covertype	74.42	0.74	73.53	0.99	72.79	1.17	74.48	0.74	74.47	0.74
mushrooms	100.00	0.00	100.00	0.00	100.00	0.00	98.58	0.77	100.00	0.00
coil 2000	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
abalone	79.87	1.18	79.90	1.05	79.86	1.02	79.87	0.96	79.65	1.25
molecular	94.22	0.80	93.94	0.73	93.40	1.28	88.04	1.68	51.92	0.00
careval	96.74	1.34	82.80	5.15	92.23	1.28	83.94	4.91	94.17	2.84
solar-c	83.53	1.23	83.61	1.38	83.83	1.08	83.76	1.02	83.61	1.38
german	74.60	2.71	74.80	2.36	74.60	3.12	72.53	3.77	75.60	3.01
australian	84.11	3.17	84.42	3.32	84.53	3.12	84.53	3.05	85.37	3.28

Table 3
Sparsity for the original SVM (SVM^O) and the CLSVM strategies.

Name	SVM^O		SVM^C		CL^{RR}		CLM^{RR}		CLM	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
census income	36.86	0.00	89.82	0.00	91.69	0.33	91.45	1.62	100.00	0.00
adult	16.07	3.26	86.24	2.21	83.42	1.34	89.83	4.36	90.77	3.64
nursery	88.89	0.00	43.12	18.21	91.85	2.22	92.59	0.00	92.59	0.00
covertype	24.81	1.48	43.57	34.17	91.82	1.82	90.91	0.00	90.91	0.00
mushrooms	28.83	0.00	76.58	0.00	80.72	4.56	78.29	5.87	85.23	1.57
coil 2000	98.70	0.00	98.70	0.00	98.70	0.00	98.70	0.00	98.70	0.00
abalone	0.00	0.00	33.33	0.00	33.33	0.00	33.33	0.00	33.33	0.00
molecular	42.96	3.12	100.00	0.00	75.13	0.19	77.29	0.90	100.00	0.00
careval	0.95	2.86	55.24	8.57	58.10	12.20	70.48	10.82	50.48	3.81
solar-c	47.83	34.51	87.83	10.43	94.78	11.14	99.13	2.61	88.69	7.83
german	5.38	1.88	57.69	0.00	61.73	4.59	63.08	4.28	57.69	0.00
australian	13.10	4.83	80.69	9.66	84.83	11.03	94.48	2.76	76.55	5.52

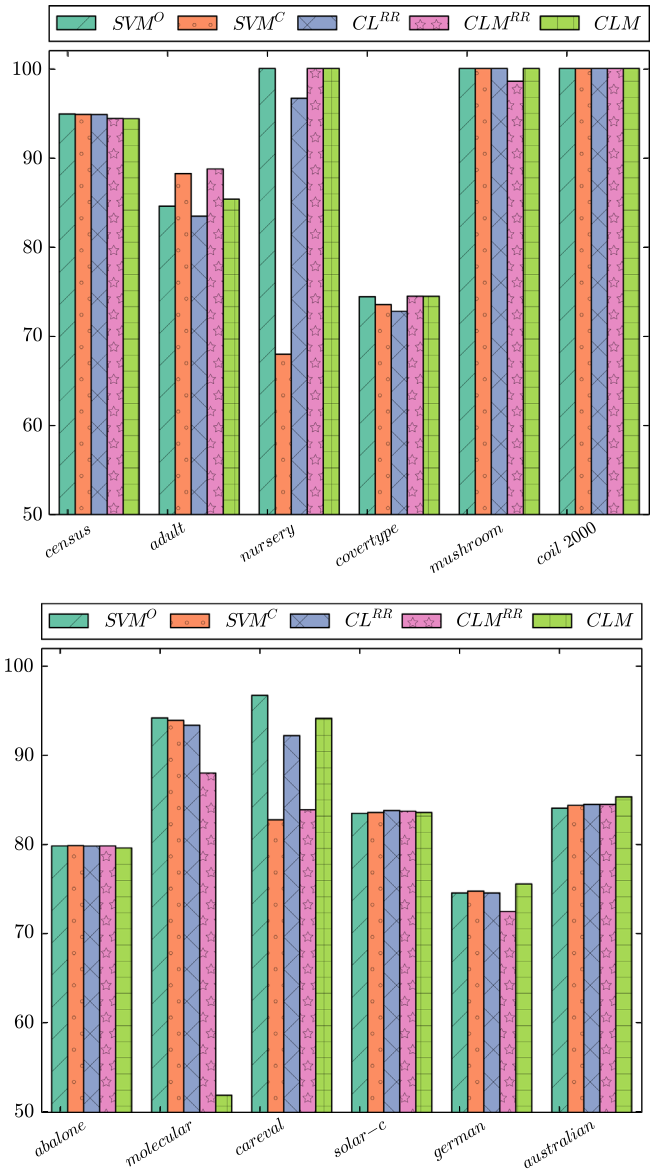

Fig. 7. Visualizing the accuracy in the validation set for the original SVM (SVM^O) and the CLSVM strategies.

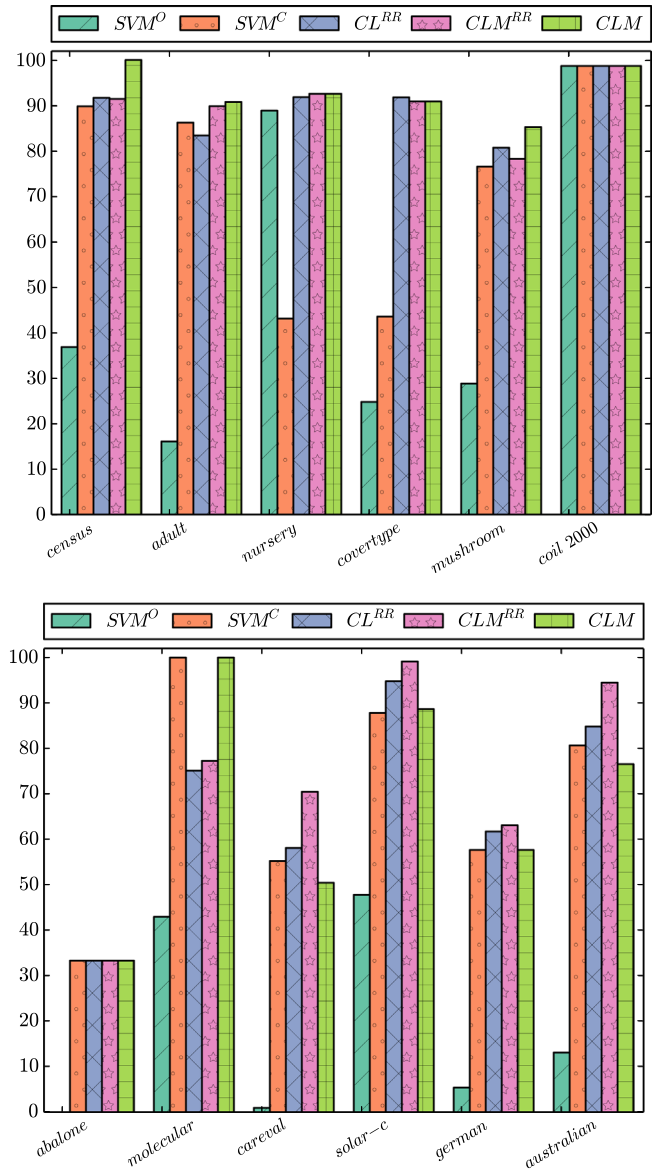

Fig. 8. Visualizing the sparsity in the validation set for the original SVM (SVM^O) and the CLSVM strategies.

maximum (or minimum) number of categories that compose a cluster, can be easily added. The former may be desirable, e.g., if categories represent countries, where one may want to impose that some countries are in the same cluster based on their geographic location. The latter may be desirable to balance the size of the clusters.

There are several interesting directions to extend the CLSVM methodology.

First, 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 classification accuracy.

Second, the simplified feature space, with fewer categories, generated by our CLSVM methodology can be seen used as input for other classifiers, such as the SVM with nonlinear kernels or classification trees. Alternatively, one can directly model the problem of clustering categories with general kernels, yielding, however, very difficult nonconvex mixed integer optimization problems. Strategies to build this nonlinear classifier deserves further study.

Third, the CLSVM methodology can be extended to handle continuous features as well. As the CLSVM aims at increasing the sparsity 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 [6,28], either binarizing or discretizing continuous features and then applying the CLSVM methodology. This extension deserves further study and testing.

Fourth, our CLSVM methodology can be combined with the strategy in [16] to deal not only with categorical features, but also with datasets with a large number of records, in order to reduce the computational burden of building the CLSVM classifier. Indeed, the goal in [16] is to reduce the computational effort when building SVM classifiers without harming classification accuracy. The records are clustered and an SVM classifier is built for each cluster, where the number of records in each cluster is much smaller than in the original dataset, yielding the desired computational savings, finally the different classifiers are combined into a single one. Merging the two methodologies (feature clustering and record clustering) in a sequential manner or developing a joint approach deserves a thorough testing, which is out of the scope of this paper.

References

[1] Aloise D, Hansen P, Liberti L. An improved column generation algorithm for minimum sum-of-squares clustering. Mathematical Programming 2012;131(1–2):195–220.

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

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

[4] Blake CL, Merz CJ. 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.

[5] Brooks JP. Support vector machines with the ramp loss and the hard margin loss. Operations Research 2011;59(2):467–479.

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

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

[8] Carrizosa E, Nogales-Gómez A, Romero Morales D. Heuristic approaches for support vector machines with the ramp loss. Optimization Letters 2014;8(3):1125–1135.

[9] Carrizosa E, Nogales-Gómez A, Romero Morales D. Strongly agree or strongly disagree?: Rating features in Support Vector Machines. Information Sciences 2016;329:256–273.

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

[11] Chaovalitwongse WA, Fan Y-J, Sachdeo RC. Novel optimization models for abnormal brain activity classification. Operations Research 2008;56(6):1450–1460.

[12] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge: Cambridge University Press; 2000.

[13] Czyzyk J, Mesnier MP, More JJ. The neos server. IEEE Computational Science Engineering 1998;5(3):68–75.

[14] Fountoulakis K, Gondzio J. A second-order method for strongly convex ℓ_1−regularization problems. Mathematical Programming 2015, 1–31, http://dx.doi.org/10.1007/s10107-015-0875-4.

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

[16] Gu Q, Han J. Clustered support vector machines. In: Proceedings of the sixteenth international conference on artificial intelligence and statistics; 2013. p. 307–15.

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

[18] Hand H, Mannila H, Smyth P. Principles of data mining. Cambridge, Massachusetts: MIT Press; 2001.

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

[20] Hassin R, Tamir A. Improved complexity bounds for location problems on the real line. Operations Research Letters 1991;10(7):395–402.

[21] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2nd edition. New York: Springer; 2009.

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

[23] Mangasarian OL, Thompson ME. Massive data classification via unconstrained support vector machines. Journal of Optimization Theory and Applications 2006;131(3):315–325.

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

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

[26] Panagopoulos OP, Pappu V, Xanthopoulos P, Pardalos PM. Constrained subspace classifier for high dimensional datasets. Omega 2016;59:40–46.

[27] Raghavan P, Tompson CD. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica 1987;7(4):365–374.

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

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

[30] Späth H. Cluster analysis algorithms for data reduction and classification of objects. Horwood, Chichester, 1980.

[31] Tawarmalani M, Sahinidis NV. Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications. Boston MA: Kluwer Academic Publishers; 2002.

[32] Vapnik V. The nature of statistical learning theory. New York: Springer-Verlag; 1995.

[33] Vapnik V. Statistical learning theory. New York: Wiley; 1998.

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

[35] Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V. Feature selection for SVMs. In: Leen TK, Dietterich TG, Tresp V, editors. Advances in neural information processing systems. Cambridge, MA: MIT Press; 2000. p. 668–674.

[36] Williams HP. Model building in mathematical programming. New York: Wiley; 1985.

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