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/© 2016 Elsevier Ltd. All rights reserved.

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$^{\mathrm{O}}$, 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$^{\mathrm{O}}$ as initial step. Then, categories are clustered using a partition of the SVM$^{\mathrm{O}}$ 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$^{\mathrm{O}}$ 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$^{\mathrm{O}}$ 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$^{\mathrm{O}}$ 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$^{\mathrm{O}}$ 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:

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

$$\text{s.t.} \tag{SVM}$$

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

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

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

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

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

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