Information Sciences 279 (2014) 163–175

Contents lists available at ScienceDirect

Information Sciences

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

Feature selection for Support Vector Machines via Mixed Integer Linear Programming

Sebastián Maldonado [a,*], Juan Pérez [a], Richard Weber [b], Martine Labbé [c]

[a] Universidad de los Andes, Mons. Álvaro del Portillo 12455, Las Condes, Santiago, Chile
[b] Department of Industrial Engineering, Universidad de Chile, República 701, Santiago, Chile
[c] Computer Science Department, Université Libre de Bruxelles, Boulevard du Triomphe, B-1050 Brussels, Belgium

ARTICLE INFO

Article history:
Received 4 August 2013
Received in revised form 5 February 2014
Accepted 26 March 2014
Available online 2 April 2014

Keywords:
Feature selection
Support Vector Machine
Mixed Integer Linear Programming

ABSTRACT

The performance of classification methods, such as Support Vector Machines, depends heavily on the proper choice of the feature set used to construct the classifier. Feature selection is an NP-hard problem that has been studied extensively in the literature. Most strategies propose the elimination of features independently of classifier construction by exploiting statistical properties of each of the variables, or via greedy search. All such strategies are heuristic by nature. In this work we propose two different Mixed Integer Linear Programming formulations based on extensions of Support Vector Machines to overcome these shortcomings. The proposed approaches perform variable selection simultaneously with classifier construction using optimization models. We ran experiments on real-world benchmark datasets, comparing our approaches with well-known feature selection techniques and obtained better predictions with consistently fewer relevant features.

1. Introduction

Support Vector Machines (SVMs) has been shown to be a very powerful machine learning method. For classification tasks and based on the structural risk minimization principle [18], this method attempts to find the separating hyperplane which has the largest distance from the nearest training data points of any class. SVM provides several advantages such as adequate generalization to new objects, a flexible non-linear decision boundary, absence of local minima, and a representation that depends on only a few parameters [18,21].

Feature selection is one of the most important steps within classification. An appropriate selection of the most relevant features reduces the risk of overfitting, thus improving model generalization by decreasing the model's complexity [7]. This is particularly important in small-sized high-dimensional datasets, where the *curse of dimensionality* is present and a significant gain in terms of performance can be achieved with a small subset of features [9,13]. Additionally, low-dimensional representation allows better interpretation of the resulting classifier. This is particularly important in applications in fields such as business analytics, since many machine learning approaches are considered to be *black boxes* by practitioners who therefore tend to be hesitant to use these techniques [6]. A better understanding of the process that generates the data by identifying the most relevant features is also of crucial importance in life sciences, where we want to identify, for example,

* Corresponding author. Tel.: +56 2 26181874.
E-mail address: smaldonado@uandes.cl (S. Maldonado).

http://dx.doi.org/10.1016/j.ins.2014.03.110
0020-0255/

those genes that best explain the presence of a particular type of cancer, and therefore could improve cancer incidence prediction.

Since the selection of the best feature subset is considered to be an NP-hard combinatorial problem, many heuristic approaches for feature selection have been presented to date [7]. With the two Mixed Integer Linear Programs for simultaneous feature selection and classification that we introduce in this paper we show that integer programming has become a competitive approach using state-of-the-art hardware and solvers.

In particular, we propose two novel SVM-based formulations for embedded feature selection, which simultaneously select relevant features during classifier construction by introducing indicator variables and constraining their selection via a budget constraint. The first approach studies an adaptation of the l_1-SVM formulation [4], while the second one extends the LP-SVM method presented in [22]. Our experiments show that the proposed methods are capable of selecting a few relevant features in all datasets used, leading to highly accurate classifiers within reasonable computational time.

Section 2 of this paper introduces Support Vector Machines for binary classification, including recent developments for feature selection using SVMs. The proposed feature selection approaches are presented in Section 3. Section 4 provides experimental results using real-world datasets. A summary of this paper can be found in Section 5, where we provide its main conclusions and address future developments.

2. Prior work on support vector classification

The mathematical derivation of the standard l_2-SVM formulation [18], the l_1-SVM formulation [4], and the LP-SVM method [22] are described in this section. The latter two linear classification methods constitute the basis for our proposed feature selection algorithms.

2.1. l_2-Support Vector Machine

Considering training examples $\mathbf{x}_i \in \Re^n$ with their respective labels $y_i \in \{-1, +1\}, i = 1, \ldots, m$, SVM determines a hyperplane $f(\mathbf{x}, \alpha)$ to separate the training examples optimally according to their labels, where $\alpha \in \Lambda$, is the set of possible model parameters.

This optimal split is based on Statistical Learning Theory [18], which provides a general measure of complexity (the VC dimension), and estimates a bound for the expected risk $R(\alpha)$ as a function of the empirical risk $R_{emp}(\alpha)$. According to this theory, the following inequality holds with probability $1 - \eta$ [22]:

$$R(\alpha) \leqslant R_{emp}(\alpha) + \frac{\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha)}{\varepsilon}}\right), \tag{1}$$

where ε is a function of the VC dimension h (a measure of complexity defined by the largest number of points that can be *shattered* by members of $f(\mathbf{x}, \alpha)$), η, and the number of instances m ($\epsilon = 4(h(ln(2m/h) + 1) - ln\ \eta)/m$). We observe that minimizing the expected risk is equivalent to simultaneously minimizing the two terms on the right-hand side of Eq. (1).

Considering a linear hyperplane of the form $f(\mathbf{x}) = \mathbf{w}^\top \cdot \mathbf{x} + b$, the SVM hyperplane then minimizes the classification errors and at the same time maximizes the *margin*, which is computed as the sum of the distances to one of the closest positive and one of the closest negative training examples, and is linked to Statistical Learning Theory since maximizing the margin is similar to minimizing the VC dimension [22].

To maximize the margin, we need to classify the training vectors $\mathbf{x}_i$ correctly into the two different classes, using the smallest norm of coefficients $\mathbf{w} \in \Re^n$ [18]. The primal SVM formulation balances the minimization of $\|\mathbf{w}\|_2^2$ (structural risk) and of the misclassification errors (empirical risk) by introducing an additional set of slack variables $\xi_i, i = 1, \ldots, m$ and a penalty parameter, C, that controls the trade-off between both objectives:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, m, \\ & \xi_i \geqslant 0, \qquad\qquad\qquad\quad i = 1, \ldots, m. \end{aligned} \tag{2}$$

2.2. l_1-Support Vector Machine

Bradley and Mangasarian [4] proposed a variation of SVM, reducing the model's complexity by using the l_1-norm (also known as LASSO penalty) instead of the Euclidean norm. This norm provides a strategy for suppressing redundant and/or irrelevant features automatically, i.e. components of the vector $\mathbf{w}$, while converting the quadratic programming problem studied by l_2-SVM (Formulation (2)) into a linear one. This formulation follows: