Computers & Operations Research 43 (2014) 328–334

Contents lists available at ScienceDirect

Computers & Operations Research

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

A nested heuristic for parameter tuning in Support Vector Machines

Emilio Carrizosa [a], Belén Martín-Barragán [b,*], Dolores Romero Morales [c]

[a] Universidad de Sevilla, Spain
[b] The University of Edinburgh, United Kingdom
[c] University of Oxford, United Kingdom

ARTICLE INFO

Available online 15 October 2013

Keywords:
Supervised classification
Support Vector Machines
Parameter tuning
Nested heuristic
Variable neighborhood search
Multiple kernel learning

ABSTRACT

The default approach for tuning the parameters of a Support Vector Machine (SVM) is a grid search in the parameter space. Different metaheuristics have been recently proposed as a more efficient alternative, but they have only shown to be useful in models with a low number of parameters. Complex models, involving many parameters, can be seen as extensions of simpler and easy-to-tune models, yielding a nested sequence of models of increasing complexity. In this paper we propose an algorithm which successfully exploits this *nested* property, with two main advantages versus the state of the art. First, our framework is general enough to allow one to address, with the very same method, several popular SVM parameter models encountered in the literature. Second, as algorithmic requirements we only need either an SVM library or any routine for the minimization of convex quadratic functions under linear constraints. In the computational study, we address Multiple Kernel Learning tuning problems for which grid search clearly would be infeasible, while our classification accuracy is comparable to that of ad hoc model-dependent benchmark tuning methods.

1. Introduction

Support Vector Machines (SVM) [4,9,15,46,47] is a Supervised Classification technique rooted in Statistical Learning Theory [46,47], whose success is based on the ability of building nonlinear classifiers.

Let Ω denote a data set of n records, each associated with a pair (x^i, y^i), with $x^i \in \mathbb{R}^d$ (the predictor vector of record i) and $y^i \in \{-1, 1\}$ (the label of record i). The SVM classifier will classify records with predictor vectors $x \in \mathbb{R}^d$ by means of a score $s(x)$ of the form

$$s(x) = \sum_{i=1}^{n} \alpha^i y^i K(x, x^i), \tag{1}$$

where $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the so-called SVM *kernel*, see [15,26,27] and references therein, and the coefficients α^i are obtained by solving the following concave quadratic maximization problem with box constraints plus one linear constraint:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \alpha^i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha^i \alpha^j y^i y^j K(x^i, x^j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha^i y^i = 0 \quad \alpha \in [0, C]^n. \end{aligned} \tag{2}$$

Here $C > 0$ is the so-called regularization parameter which bounds the influence of each record i in the score function s. It is well-known that the choice of both the kernel K and the regularization parameter C is crucial to the SVM classification accuracy, [32]. For this reason, *tuning* (i.e., choosing) the SVM parameters becomes a fundamental yet nontrivial issue. Designing *simple* and *effective* tuning procedures will be useful for the wide variety of practitioners using SVM.

In order to formulate the SVM parameter tuning problem, note that, setting $\vartheta^i = \alpha^i / C$ in (2), we obtain the equivalent problem

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \vartheta^i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \vartheta^i \vartheta^j y^i y^j C K(x^i, x^j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \vartheta^i y^i = 0 \quad \vartheta \in [0, 1]^n. \end{aligned} \tag{3}$$

From this formulation it is clear that the classifier obtained using either (2) or (3) depends on C and K through its product CK. Tuning $C > 0$ and K in a given class of kernels $\mathcal{K}_0$ is therefore equivalent to selecting K in the conic hull of $\mathcal{K}_0$, $\mathcal{K} = \{CK : C > 0, K \in \mathcal{K}_0\}$.

Ideally, K should be chosen by maximizing $a(K)$, the probability of correct classification of incoming records if one classifies following the classifier obtained from (1). Since the SVM theory makes no distributional assumptions on the incoming data, $a(\cdot)$ cannot be evaluated, and, instead, an estimate $\hat{a}(\cdot)$ based on the training data set Ω, such as k-fold crossvalidation accuracy [30], is used to guide the choice of K. Now the SVM parameter tuning problem can be formulated as the optimization problem

$$\begin{aligned} \max \quad & \hat{a}(K) \\ \text{s.t.} \quad & K \in \mathcal{K}. \end{aligned} \tag{4}$$

Many classes of kernels have been proposed in the literature. The simplest model for $\mathcal{K}$ is the one in which the kernel is assumed to

* Corresponding author.
E-mail addresses: ecarrizosa@us.es (E. Carrizosa), Belen.Martin@ed.ac.uk (B. Martín-Barragán), dolores.romero-morales@sbs.ox.ac.uk (D. Romero Morales).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.10.002

be proportional to a fixed base kernel K_0, namely

$$\mathcal{K} = \{CK_0 : C > 0\}. \tag{5}$$

As K_0 one can take, for instance, the so-called *linear kernel*,

$$K^{\text{lin}}(x,z) = x^\top z,$$

yielding the standard SVM model [9,15,27,46,47]. A very simple yet extremely powerful is the class of *Radial Basis Function* (RBF) kernels [15,29],

$$\mathcal{K} = \{CK_\sigma^{\text{RBF}} : C > 0, \sigma > 0\},$$
$$K_\sigma^{\text{RBF}}(x,z) = \exp\left(-\sum_{i=1}^{d}(x_i - z_i)^2/\sigma\right), \tag{6}$$

which has been extended by considering the scaling factor σ to be variable-dependent, yielding the *anisotropic* RBF model, see e.g. [12]. An alternative model studied, among others, in [1,12,22,38,37,31,39,45], is the *Multiple Kernel Learning* (MKL) model. MKL is especially suitable when the data set has variables of different nature, calling for the use of different kernel models for the different types of variables involved. In its simplest version, R base kernels, $K_1, \ldots, K_R$, are given, and a conic combination is sought:

$$\mathcal{K} = \left\{\sum_{j=1}^{R}\mu_j K_j : \mu_j \geq 0 \quad \forall j = 1,2,\ldots,R\right\}. \tag{7}$$

Such base kernels K_j may be, for instance, RBF kernels with different (but fixed) scaling factors σ_j for each j. While it is frequently claimed that the most relevant parameters to be tuned are the weights in the conic combination of kernels, [22], one may also consider to tune the kernels K_j, choosing them from different kernel sets $\mathcal{K}_j$, [22], yielding

$$\mathcal{K} = \left\{\sum_{j=1}^{R}\mu_j K_j : \mu_j \geq 0, \quad K_j \in \mathcal{K}_j \quad \forall j = 1,\ldots,R\right\}. \tag{8}$$

This ends our review of the most popular kernel models in the literature. At this point, it is important to stress that, the richer the kernel class, the higher the value of the estimate $\hat{a}$, but this does not necessarily imply that the actual classification rate a also improves when the kernel class is enriched, due to the so-called overfitting phenomenon. This explains the variety of models, with different levels of generality, that can be found in the literature, and the need for a tuning method to be able to adapt to them.

To end with the structure of the tuning problem (4), we now discuss its objective function and the challenges when optimizing it. Some papers take as surrogate $\hat{a}(\cdot)$ of the accuracy $a(\cdot)$ a distribution-free, but kernel-specific, bound on the probability of misclassification, see [12,18,48]. While such functions $\hat{a}$ are usually smooth in the parameters, allowing for the use of high-order local search methods, other surrogates, not necessarily differentiable, have also been proposed, [3,21,50]. Most of the papers take as $\hat{a}$ the k-fold crossvalidation accuracy estimate, see [30]. This is also the approach taken in this paper. Note that in this case the cost of evaluating $\hat{a}$ is high. Indeed, evaluating $\hat{a}$ at a given set of parameter values amounts to solving k quadratic problems of the form (3). Also, local-search optimization methods might be not effective because the problem is multimodal, and these methods are challenged by the fact that the objective function is piecewise constant, and hence gradient-type information may be useless.

In this second part of the introduction, we review proposals to solve the resulting optimization problem. For simple kernel models, such as (5) with one single parameter C, the tuning is usually done by a grid search on a sufficiently big interval, say $[2^{-12}, 2^{12}]$. However, and due to the cost of evaluating the objective function, grid search is quite inefficient, becoming infeasible if the dimension of the parameter space is not too small, even if the grid is not too fine. Several heuristic algorithms have been proposed in the literature. Some are ad hoc for a particular kernel model, such as [29], while others are metaheuristics.

An early reference on metaheuristics is [43], where a Pattern Search approach is introduced. An improvement is proposed in [2], where Simulated Annealing is used to screen the neighborhoods in [2].

Since [43], many other metaheuristics have been proposed in the literature. In [13], a genetic algorithm is used for parameter tuning within the RBF kernel model. Since the parameters are real-valued, a 0–1 encoding, of a given precision, is used. Alternative mutation and crossover operators for real-valued parameters are proposed in [36]. In [19], an evolutionary algorithm based on the so-called Covariance Matrix Adaptation Evolution Strategy, [24], is proposed.

In [20], the so-called Efficient Parameter Selection via Global Optimization algorithm is proposed. It is an iterative method based on estimating the objective function given its value in a collection of inspected solutions. This is done using an online Gaussian process, whose parameters are chosen by maximum likelihood. As the authors point out, this method is only competitive when the dimension of the parameter space is low.

Other popular metaheuristic strategies such as Variable Neighborhood Search and Ant Colony Optimization have received perhaps less attention when tuning SVM parameters, [10,51].

Most of the existing approaches in the literature show their performance in the RBF kernel model (6), where only two parameters, C and σ, are to be tuned. An exception is [2], where the anisotropic RBF kernel model [12] is considered. This is a generalization of the RBF kernel model in which parameters C and σ_i ($i=1,\ldots,d$) need to be tuned. As is the case for the anisotropic RBF kernel, complex models, involving many parameters, can be seen as extensions of simpler and easy-to-tune models, yielding a nested sequence of models of increasing complexity.

In this paper we propose an algorithm which successfully exploits this *nested* property of complex methods, i.e., the ability to define a sequence of nested subproblems, with two main advantages. First, our framework is general enough to allow one to address, with the very same method, several popular SVM parameter models encountered in the literature. Indeed, to illustrate the versatility of our algorithm, we present experiments for an array of MKL models. MKL models have attracted a lot of attention and many ad hoc approaches exist, see [22] for a through review on the most successful of these approaches. Second, as algorithmic requirements we only need a black box to train SVMs. In other words, as soon as an SVM package, such as LIBSVM [11], SVMTorch [14] or SVMlight [28], a general-purpose scientific computing, Statistics or Machine Learning package such as MATLAB, R, SAS or WEKA [49], or any routine for the minimization of convex quadratic functions under linear constraints is at hand, our approach is readily applicable. In contrast, some of the specialized MKL techniques we compare with require, for instance, Second-Order Cone Programming (SOCP) solvers, as in [1].

The remainder of the paper is structured as follows. In Section 2, we propose our nested heuristic, which is tested in Section 2.2 against benchmark methods for different kernel models. Concluding remarks and lines of future research are outlined in Section 4.

2. The nested heuristic

In this section we propose a nested heuristic for SVM parameter tuning, where we assume that a nested structure for the kernel model to be tuned and a metaheuristic are at hand. Below we discuss these two ingredients before presenting the algorithm.