Neurocomputing 101 (2013) 59–67

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Sphere Support Vector Machines for large classification tasks

Robert Strack*, Vojislav Kecman, Beata Strack, Qi Li

Virginia Commonwealth University, Computer Science Department, Richmond, VA 23284-3019, United States

ARTICLE INFO

Article history:
Received 9 March 2012
Received in revised form
1 July 2012
Accepted 7 July 2012
Communicated by K.J. Cios
Available online 25 August 2012

Keywords:
Support Vector Machines
Core Vector Machines
Minimum enclosing ball
Large datasets
Classification

ABSTRACT

This paper introduces Sphere Support Vector Machines (SVMs) as the new fast classification algorithm based on combining a minimal enclosing ball approach, state of the art nearest point problem solvers and probabilistic techniques. The blending of the three significantly speeds up the training phase of SVMs and also attains practically the same accuracy as the other classification models over several large real datasets within the strict validation frame of a double (nested) cross-validation. The results shown are promoting SphereSVM as outstanding alternatives for handling large and ultra-large datasets in a reasonable time without switching to various parallelization schemes for SVM algorithms recently proposed.

1. Introduction

Support Vector Machines are considered to be among the best classification tools available today. Many experimental results achieved on a variety of classification (and regression) tasks complement the highly appreciated theoretical properties of SVMs. However, there is one property of SVM learning algorithm that has required, and still requires, special attention. This is the fact that the learning phase of SVMs scales with the number of training data points. The time complexity of the general purpose, state-of-the-art SVM implementations is somewhere between $O(n)$ and $O(n^{2.3})$. Hence, with an increase of datasets sizes, the learning phase can be a quite slow process. Some successful attempts to deal with this matter include the decomposition approaches that have led to several efficient pieces of software, the most popular being SVMlight [1] and LIBSVM [2]. However, none of these algorithms obtained linear complexity, and the ever increasing size of datasets has driven the SVMs training time beyond acceptable limits. The two remedial avenues for overcoming the issues of large datasets employed during the last decade include various parallelization attempts (including the newest GPU embedded implementations [3,4]) and the use of geometric approaches. The latter include solving SVMs learning problem by employing both convex hulls and enclosing ball approaches [5,6]. The most recent and advanced method, known as the Ball Vector Machines [7], has shown a high capacity for handling large datasets.

The Sphere Support Vector Machine (SphereSVM) proposed here combines the two techniques (convex hulls and enclosing balls approaches). While keeping the level of accuracy, it achieves a significant speedup with respect to all three L1 and L2 LIBSVM and BallVMs.

Although the most popular SVM solvers (such as Platt's SMO [8]) are based on the Lagrange multipliers method and search for solutions in the dual space, there is substantial research conducted towards finding efficient algorithms that work directly in the feature space. These algorithms are mostly based on the geometric interpretation of the maximum margin classifiers.

The geometric properties of hard margin SVM classifiers have been known for a long time [9]. Recently, Keerthi et al. [10] and Franc [11] proposed algorithms based on the geometric interpretation of the SVM algorithm for solving cases with separable classes. Their approach treats the problem of finding the maximum margin between two classes as a problem of finding two closest points belonging to convex polytopes covering the classes. Crisp et al. analyzed the geometric properties of the ν-SVM algorithm [12] and, based on this work, Mavroforakis introduced the reduced convex hulls (RCH) [13]. This idea allowed using a geometric approach to solve SVM problems for overlapping classes. Reduced convex hulls enabled a shrink of overlapping convex polytopes covering each class. This reduction creates the margin between two overlapping classes and permits separating them (previously unfeasible with Keerthi's or Franc's algorithms).

Another field of research involves algorithms based on the minimal enclosing ball (MEB) problem. Tsang et al. [5,14] formulated

* Corresponding author.
E-mail address: strackr@vcu.edu (R. Strack).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.07.025

the SVM problem as the MEB problem and proposed Core Vector Machines (CVM) algorithm as an approach suitable for very large SVM training. Their algorithm is an application of Badoiu and Clarkson's work [15] that investigates the use of coresets in finding an approximation of MEB. Additional speedup is obtained by using the "probabilistic speedup" approach proposed by Smola and Schölkopf [16]. CVM is further generalized [17] by allowing the use of any kernel function (and not only the normalized ones as previously required). Furthermore, Tsang et al. [7] have improved the idea of Core Vector Machines by introducing a new algorithm not requiring QP solver—Ball Vector Machines (BVM). Moreover, Asharaf et al. [18] have proposed another extension of CVM that is capable of handling multi-class problems.

In this paper, we propose a new algorithm that improves the BVM by applying ideas previously used in SVM learning based on RCH. The original SVM solver involving RCH was improved by López et al. [19] by replacing the SK algorithm [20] that was used in searching for the closest points with a faster MDM algorithm introduced by Michell et al. in [21]. Our work, similar to López's, introduces an algorithm originating within an MDM solver as the technique for finding an enclosing ball (EB) surrounding the data points. This novel EB algorithm is successfully applied in solving the EB problem that originates after transforming an original L2 SVM problem as the EB task.

2. Sphere Support Vector Machines

It has been shown in [5] that, for normalized kernels,[1] the learning setting of the L2 SVM, defined as

$$\arg\min_{\mathbf{w},b,\zeta,\rho} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{b^2}{2} - \rho + \frac{C}{2}\sum_{i=1}^{m}\zeta_i^2, \tag{1}$$

subject to

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq \rho - \zeta_i, \quad i = 1,\ldots,m \tag{2}$$

can be rewritten as a minimization task equal to solving a problem of the finding minimal enclosing ball

$$\arg\min_{R,\mathbf{c}} R^2, \tag{3}$$

subject to

$$\|\mathbf{c} - \tilde{\mathbf{x}}_i\|^2 \leq R^2, \quad i = 1,\ldots,m \tag{4}$$

in the feature space $\tilde{\Phi}$ defined by kernel

$$\tilde{k}_{ij} = y_i y_j k_{ij} + y_i y_j + \frac{\delta_{ij}}{C}, \tag{5}$$

where $k_{ij} = \varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_j)$ is the kernel used in the original L2 SVM problem, δ_{ij} is Kronecker's delta, and $\tilde{\mathbf{x}}_i = \tilde{\varphi}(\mathbf{x}_i)$ is the image of the vector $\mathbf{x}_i$ in the feature space $\tilde{\Phi}$. In other words, solving the minimal enclosing ball problem will also produce a solution of the L2 SVMs task. This idea becomes the foundation of the CVM and BVM algorithms introduced by Tsang, as well as the new approach presented here.

The minimal enclosing ball problem solved by the CVM algorithm was slightly simplified in the BVM approach. Specifically, Tsang found an accurate estimate of the radius of the enclosing ball. Since the ball's radius could be considered known, the only unknown was the center of the ball. This way, the minimal enclosing ball problem was replaced by an enclosing ball problem. This approach turned out to be very effective and so we decided to apply it to our algorithm. SphereSVM, like its predecessor BVM, is not trying to minimize the radius of the enclosing ball.

2.1. The algorithm

The SphereSVM algorithm is a novel reformulation of the BVM approach. Therefore, some parts of both algorithms are similar. For instance, the initialization procedure, the way the violating vectors[2] are found, and the stopping criterion are all the same. However, there are important differences, the main one being the way updates of the center are performed. In the case of a BVM algorithm, all weights α_i corresponding to vectors $\tilde{\mathbf{x}}_i$ belonging to the coreset are modified in each updating step. In the SphereSVM algorithm proposed here, only two weights α_v and α_u are updated. The first weight α_v corresponds to the vector that is furthest from the ball center while the other weight α_u belongs to the support vector closest to the center. According to the KKT conditions of the MEB problem

$$\alpha_i(\|\mathbf{c} - \tilde{\mathbf{x}}_i\|^2 - R^2) = 0, \tag{6}$$

if the condition $\alpha_i \neq 0$ holds then $\tilde{\mathbf{x}}_i$ lies on the boundary of the minimal enclosing ball. In other words, the vectors lying inside the ball are not support vectors and do not affect the solution. This observation leads to the conclusion that there are two types of violators: the vectors lying outside the enclosing ball and the vectors having nonzero weights that are lying inside the ball. The SphereSVM algorithm aims at eliminating support vectors inside the ball.

The simplified pseudo-code of the SphereSVM algorithm is presented in Algorithm 1.

Algorithm 1. SphereSVM algorithm.

Require: $\varepsilon \in [0,1)$ {the parameter of the stopping criterion}
Require: $N_r \in \mathbb{Z}_+$ {the size of the random subset}
Require: $N_a \in \mathbb{Z}_+$ {the number of draw attempts}
Ensure: $\mathbf{c} = \sum_{i=1}^{m}\alpha_i\tilde{\mathbf{x}}_i$ {the approximation of the EB center}

$\boldsymbol{\alpha} \leftarrow \mathbf{0}, \alpha_0 \leftarrow 1$
$\hat{R} \leftarrow \sqrt{\tau + 1 + \frac{1}{C}}$
$\hat{\varepsilon} \leftarrow \frac{1}{2}$
repeat
  $\hat{\varepsilon} \leftarrow \max\{\hat{\varepsilon}, \varepsilon\}$
  $i \leftarrow 0$
  repeat
    $X_r \leftarrow$ random subset of X of size N_r
    $v \leftarrow \arg\max_{i:\tilde{\mathbf{x}}_i \in X_r \wedge \|\mathbf{c}-\tilde{\mathbf{x}}_i\| > (1+\hat{\varepsilon})\hat{R}} \|\mathbf{c} - \tilde{\mathbf{x}}_i\|$
    $i \leftarrow i + 1$
  until $v \neq \emptyset$ **or** $i > N_a$
  if $v \neq \emptyset$ **then**
    $u \leftarrow \arg\min_{i:\alpha_i > 0} \|\mathbf{c} - \tilde{\mathbf{x}}_i\|$
    $\rho = \frac{(\tilde{\mathbf{x}}_v - \tilde{\mathbf{x}}_u)\cdot(\tilde{\mathbf{x}}_v - \mathbf{c})}{\|\tilde{\mathbf{x}}_v - \tilde{\mathbf{x}}_u\|^2}$
    $\hat{\beta} \leftarrow \rho - \sqrt{\rho^2 - \frac{\|\tilde{\mathbf{x}}_v - \mathbf{c}\|^2 - \hat{R}^2}{\|\tilde{\mathbf{x}}_v - \tilde{\mathbf{x}}_u\|^2}}$
    $\beta \leftarrow \min\{\hat{\beta}, \alpha_u\}$
    $\alpha_v \leftarrow \alpha_v + \beta$
    $\alpha_u \leftarrow \alpha_u - \beta$
  else
    $\hat{\varepsilon} \leftarrow \frac{\hat{\varepsilon}}{2}$

---

[1] Kernels satisfying condition $k_{ii} = \varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_i) = \tau$ is constant, e.g. for a Gaussian kernel $k_{ii} = 1$.

[2] Violating vectors are the data points that violate some predefined conditions. In the case of the BVM algorithm, these are the vectors lying outside the enclosing ball. In SphereSVM, there is also another type of violators, namely vectors that do not satisfy KKT conditions (samples $\tilde{x}_i$ having non-zero weight α_i and not being on the surface of the enclosing ball).