Knowledge-Based Systems 000 (2018) 1–13

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

# A safe accelerative approach for pinball support vector machine classifier

Zhiji Yang, Yitian Xu*

*College of Science, China Agricultural University, Beijing, 100083, China*

ARTICLE INFO

*Article history:*
Received 10 October 2017
Revised 1 February 2018
Accepted 4 February 2018
Available online xxx

*Keywords:*
Support vector machine
Pinball loss
Safe screening
Variational inequality

ABSTRACT

Support vector machine (SVM) and its extensions have seen many successes in recent years. As an extension to enhance noise insensitivity of SVM, SVM with pinball loss (PinSVM) has attracted much attention. However, existing solvers for PinSVM still have challenges in dealing with large data. In this paper, we propose a safe screening rule for accelerating PinSVM (SSR-PinSVM) to reduce the computational cost. Our proposed rule could identify most inactive instances, and then removes them before solving optimization problem. It is safe in the sense that it guarantees to achieve the exactly same solution as solving original problem. The SSR-PinSVM covers the change of multiple parameters. The existing DVI-SVM can be regarded as a special case of SSR-PinSVM when the parameter $\tau$ is constant. Moreover, our screening rule is independent from the solver, thus it can be combined with other fast algorithms. We further provide a dual coordinate descent method for PinSVM (DCDM-PinSVM) as an efficient solver in this paper. Numerical experiments on six artificial data sets, twenty-three benchmark data sets, and a real biological data set have demonstrated the feasibility and validity of our proposed method.



## 1. Introduction

Support vector machine (SVM) proposed by Vapnik is one of the most successful classification algorithms [1,2]. Many improvements have been developed to enhance its performance, and they have gained widespread use in recent years [3–6].

The basic idea of SVM is to construct two parallel hyperplanes to separate two classes of instances and maximize the distance between the hyperplanes. The penalty on the training instances is controlled by a hinge loss. However, the hinge loss related to the shortest distance between two classes is sensitive to noise. To address this issue, Huang et al. proposed an SVM classifier with pinball loss (PinSVM) to improve the prediction performance [7]. The pinball loss is related to the quantile distance and is less sensitive to the noise data [8–10]. The major advantage of PinSVM is the noise insensitivity, especially for the feature noise around the decision boundary. The computational complexity of the PinSVM is similar to the hinge loss SVM, but the PinSVM loses the sparsity. Although the improved method in [10] enjoys sparsity to some extent, it does not perform well on computational speed, since its optimization problem is non-convex. To solve this problem, it needs a complex iterative process to transform the non-convex optimization problem into a series of convex ones.

Some efficient algorithms have been presented to improve the computational speed of PinSVM. In [11], the sequential minimal optimization for PinSVM (SMO-PinSVM) was established to solve its optimization problem, which broke large quadratic programming problem (QPP) into a series of QPPs as small as possible and allowed to handle large training sets [12]. In [13], an algorithm was established to find the entire solution path for PinSVM with different $\tau$ values. Additionally, in [14,15], the pinball loss was introduced into twin support vector machines (TSVMs) [16–18] to enhance the robustness. Besides, researchers have proposed some efficient solvers to further improve the computational speed of TSVMs [19,20].

Although the efficient algorithms for PinSVM have faster computational speed, they have not taken full advantage of the characteristic of PinSVM in which most elements of its dual solution $\lambda$ are equal to a constant $-\tau c$ or $c$, and a few elements are in the closed interval $(-\tau c, c)$. For convenience, we define the instances corresponding to $\lambda_i \in (-\tau c, c)$ as active instances, and those instances corresponding to $\lambda_j = -\tau c$ or $c$ as inactive instances. In the training process, if the great mass of inactive instances could be identified in a short time before solving optimization problem, it only needs to solve a smaller-sized problem rather than the whole one. Thus, the training speed will be greatly accelerated. But for the algorithms mentioned above, inactive instances cannot

* Corresponding author.
*E-mail addresses:* yzhiji@cau.edu.cn (Z. Yang), xytshuxue@cau.edu.cn (Y. Xu).

https://doi.org/10.1016/j.knosys.2018.02.010
0950-7051/

be identified before solving QPP. So we still need to solve a large optimization problem consisting of all training instances, which is a hard challenge especially for large-scale data sets. Considering these above, we try to propose an instance reduction strategy for accelerating PinSVM.

There are many existing approaches for instance reduction. In general, they can be grouped in two main categories [21–23], according to their relationship with the training process: 1) as a pre-processing step before training [24,25]; 2) embedded in the training algorithms [26,27]. Most existing approaches belong to the first category. These methods are independent from the training models. For this category, the difficulty is to ensure whether the pre-processor is fit for the training models or not. The common way to verify the performance of the pre-processor is just to use k-nearest neighbors classifier as the learning algorithm. For the second category, the instance reduction is embedded in the specific models. To some extent, this strategy could better guarantee the safety. However, those approaches mentioned above are all based on some prior assumptions, such as clustering assumption. In real applications, if the data do not conform the assumption, it leads to a negative consequence. Furthermore, the extra time and memory are required in the process of instance reduction.

Recently, a promising technique called "safe screening" was presented to handle large-scale data sets for sparse models [28–37]. The screening rule is embedded in the training process. Compared with those reduction methods mentioned above, this screening technique has the following advantages: 1) the instances discarded by the screening technique are mathematically guaranteed to have no influence on the solutions without any prior assumption; 2) the screening process runs in negligible time.

The screening approach for the traditional SVM "the dual problem of SVM via variational inequalities (DVI-SVM)" [38] has shown its outstanding performance. It significantly reduces the computational cost and memory. Moreover, this method is safe in the sense that the instances discarded by the rule are guaranteed to be non-support vectors.

Although DVI-SVM is an admirable method to reduce the computational cost of traditional SVM, it cannot be directly applied to PinSVM. For different models, the corresponding screening rules differ from each other. In the DVI-SVM, there is only one parameter $c$, and the feasible region of dual solution is invariable. But in PinSVM, there are two parameters $\tau$ and $c$, and the feasible region of solution changes with $\tau$.

In this paper, by introducing the screening idea into PinSVM, we propose a safe screening rule for accelerating PinSVM (SSR-PinSVM) to reduce the computational cost. The SSR-PinSVM is presented by analyzing the dual problem via Karush-Kuhn-Tucker (KKT). By our screening rule, the inactive instances can be substantially reduced. Our method is safe in the sense that it does not sacrifice the optimal solution. Different from the existing DVI-SVM, our method contains multiple parameters which are allowed to change simultaneously. Besides, when the parameter $\tau$ is fixed, our SSR-PinSVM is reduced to the DVI-SVM. Moreover, the SSR is independent from the solver as it is applied before solving the optimization problem. Hence, other efficient solvers can be combined. We propose a dual coordinate descent method (DCDM) as the efficient solver.

The main contributions of our approach are summarized as follows:

(i) We propose an SSR-PinSVM via variational inequalities to efficiently deal with large data, which can speed up the original solver several times and guarantees to achieve the same solution.

(ii) Multiple parameters $\tau$ and $c$ are included in our SSR-PinSVM. It is different from the previous DVI-SVM in which only one parameter $c$ is considered.

(iii) Our rule is independent from the solvers. In this paper, we propose the DCDM as the solving algorithm.

The paper is outlined as follows. Section 2 briefly introduces PinSVM and gives motivations for our method. Our SSR-PinSVM is proposed in Section 3. Some discussions are given in Section 4. Section 5 performs numerical experiments on six artificial data sets, twenty-three benchmark data sets, and a real biological data set to investigate the validity of the proposed algorithm. The last section contains the conclusions.

**Notations:** Throughout this paper, scalars are denoted by italic letters, and vectors by bold face letters. We use $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i \mathbf{x}_i y_i$ to denote the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$, and $\|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle$. For vector $\mathbf{x}$, let $x_i$ be the $i$th component of $\mathbf{x}$. If $X$ is a matrix, $\mathbf{X}_i$ is the $i$th row of $X$ and $X_{i,j}$ is the $(i, j)$th entry of $X$. For a vector $\mathbf{x}$ or a matrix $X$, let $\mathbf{x}_{\mathcal{J}} = (x_{j_1}, x_{j_2}, \cdots, x_{j_k})^T$ and $X_{\mathcal{J}} = (\mathbf{X}_{j_1}, \mathbf{X}_{j_2}, \cdots, \mathbf{X}_{j_k})^T$. In addition, for a feasible solution $\boldsymbol{\theta}$ in an optimization problem, $\boldsymbol{\theta}^*$ denotes its corresponding optimal value.

## 2. Preliminaries

We review the PinSVM, and then give motivations for our methods.

### 2.1. Review on PinSVM

Suppose we have a set of observations $\{\mathbf{x}_i, y_i\}_{i=1}^l$, where the row vector $\mathbf{x}_i \in \mathbb{R}^n$ represents the $i$th data instance, and $y_i \in \mathbb{R}$ is the corresponding response.

The traditional SVM with hinge loss is formulated as:

$$\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{l} L_{\text{hinge}}(1 - y_i f(\mathbf{x}_i))$$

where $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle + b$, $c$ is a trade-off parameter and $L_{\text{hinge}}(u) = \max\{u, 0\}$. $L_{\text{hinge}}$ is called a hinge loss function by which sufficiently rightly classified points are not punished. For this reason, the traditional SVM enjoys good sparsity. However, this feature also leads to the sensitivity of noises, since most of the instances are rightly classified and only a small number of the misclassified points determine the classifier.

To address this issue, Huang et al. [7] introduce pinball loss into the SVM to improve the prediction performance. The pinball loss is given as follows:

$$L_\tau(u) = \begin{cases} u, & \text{if } u \geq 0, \\ -\tau u, & \text{if } u < 0, \end{cases}$$

where $\tau$ is a parameter. It not only gives penalty on the misclassified points, but also takes weight into the sufficiently rightly classified instances. That enhances the noise insensitivity of the classifier, especially for the feature noise around the decision boundary.

The SVM with pinball loss is formulated as the following QPP:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{l} \xi_i \\ \text{s.t.} \ & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \\ & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq 1 + \frac{1}{\tau}\xi_i, \ \ i = 1, 2, \cdots, l. \end{aligned} \tag{1}$$

Once the optimal solutions $\mathbf{w}^*$ and $b^*$ are achieved, we can determine the label of a new testing point $\mathbf{x}$ with the following decision function:

$$D(\mathbf{x}) = \text{sign}(f(\mathbf{x})) = \text{sign}(\langle \mathbf{w}^*, \mathbf{x} \rangle + b^*).$$