Neurocomputing 130 (2014) 3–9

Contents lists available at ScienceDirect

Neurocomputing

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

Recursive least squares projection twin support vector machines for nonlinear classification

Shifei Ding [a,b,*], Xiaopeng Hua [a,c]

[a] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China
[b] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Science, Beijing 100190, China
[c] School of Information Engineering, Yancheng Institute of Technology, Yancheng, Jiangsu 224001, China

ARTICLE INFO

Article history:
Received 23 June 2012
Received in revised form
20 February 2013
Accepted 26 February 2013
Available online 21 August 2013

Keywords:
Projection twin support vector machine
Least squares
Recursive learning
Kernel trick

ABSTRACT

During the last few years, multiple surface classification (MSC) algorithms, such as projection twin support vector machine (PTSVM), and least squares PTSVM (LSPTSVM), have attracted much attention. However, there are not any modifications of them that have been presented to handle nonlinear classification. This motivates the rush towards new classifiers. In this paper, we formulate a nonlinear version of the recently proposed LSPTSVM for binary nonlinear classification by introducing nonlinear kernel into LSPTSVM. This formulation leads to a novel nonlinear algorithm, called nonlinear LSPTSVM (NLSPTSVM). Additionally, in order to promote its generalization capability, we also extend the recursive leaning method, used for further boosting the performance of PTSVM and LSPTSVM, to the nonlinear case. Experimental results on synthetic datasets, UCI datasets and NDC datasets show that NLSPTSVM has better classification capability.

1. Introduction

Support vector machine (SVM) [1,2], as an effective kernel based learning algorithm, has been successfully applied to pattern classification and regression estimation like face detection [3], drug discovery [4] and time series prediction [5]. Instead of employing empirical risk minimization principle like conventional artificial neural networks, SVM implements structural risk minimization principle to achieve better generalization ability [6]. The central idea of SVM is to find an optimal separating hyperplane which is defined as the one giving maximum margin between the positive and negative training samples that are closest to the hyperplane [6,7].

Multisurface proximal support vector machine via generalized eigenvalues (GEPSVM) [8], as a novel variant of SVM, does binary classification by generating two nonparallel hyperplanes, one for each class. In this approach, each hyperplane, which is closest to the samples of its own class and furthest from the samples of the other class, is determined by solving the eigenvector corresponding to a smallest eigenvalue of a generalized eigenvalue problem. The new datapoints are assigned to a class based on its proximity to one of the two hyperplanes.

Twin support vector machine (TSVM) [9], similar in spirit to GEPSVM, seeks two nonparallel hyperplanes by solving two dual quadratic programming problems (QPPs) of smaller size rather than solving single dual QPP with all training samples in SVM. Experimental results show the effectiveness of TSVM over SVM and GEPSVM [9,10].

Multi-weight vector projection support vector machine (MVSVM) [11], different from TSVM which improves GEPSVM by using SVM-type formulation, was proposed to enhance the performance of GEPSVM by seeking two projection weight vectors instead of two hyperplanes for each class. In this approach, the projection weight vectors can be found by solving a pair of eigenvalue problems, such that the samples of one class are closest to its class mean while the samples of different class are separated as far as possible [11].

Based on MVSVM and TSVM, projection twin support vector machine (PTSVM) is proposed in [12]. It seeks a projection axis for each class by solving an associated SVM-type QPP, such that the projected samples are well separated from those of the other class in its respective subspace. Additionally, to further boost its performance, the authors propose a recursive procedure to generate more than one axis for each class. Experimental results show the effectiveness of PTSVM over TSVM and MVSVM [12]. In order to dealing with large datasets efficiently, a least squares PTSVM (LSPTSVM) was proposed in [13]. The solution of LSPTSVM follows directly from solving two systems of linear equations as opposed to solving two QPPs in PTSVM. This makes LSPTSVM be able to solve large datasets accurately without any external optimizers [13].

* Corresponding author at: China University of Mining and Technology, School of Computer Science and Technology, Xuzhou, Jiangsu 221116, China
E-mail addresses: dingsf@cumt.edu.cn (S. Ding), xp_hua@163.com (X. Hua).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.02.046

However, the authors only proposed linear algorithms both in [12] and [13], and did not extended them to solve nonlinear classification. In this paper, we first extend LSPTSVM to handle nonlinear kernels whose solution also leads to systems of linear equations. We call this method the nonlinear LSPTSVM (NLSPTSVM). Then, in order to further boost its performance, we extend the recursive leaning method, used for further promoting the performance of PTSVM and LSPTSVM, to the nonlinear case.

The rest of this paper is organized as follows. In Section 2, we give a short summary of PTSVM and LSPTSVM. NLSPTSVM is presented and discussed in Section 3. Section 4 deals with experimental results and Section 5 contains concluding remarks.

2. Brief review of PTSVM and LSPTSVM

Consider a binary classification problem in the n-dimension real space R^n. The set of m training samples is represented by $\{(\boldsymbol{x}_j^{(i)}, y_j) | i=1,2;\ j=1,2,...,m_i\}$ where $\boldsymbol{x}_j^{(i)}$ denotes the jth training sample corresponding to class i and $m_1+m_2=m$, $y_j\in\{-1,1\}$. We further organize the m_1 samples of class 1 by a $m_1\times n$ matrix $\boldsymbol{A}$ and m_2 samples of class 2 by a $m_2\times n$ matrix $\boldsymbol{B}$.

2.1. Projection twin support vector machine

The central idea in PTSVM [12] is to find a projection axis for each class, such that within-class variance of the projected samples of its own class is minimized meanwhile the projected samples of the other class scatter away as far as possible. This leads to the following two optimization problems [12]

$$\min_{\omega_1,\xi}\frac{1}{2}\sum_{i=1}^{m_1}\left(\omega_1^T\boldsymbol{x}_i^{(1)}-\omega_1^T\frac{1}{m_1}\sum_{j=1}^{m_1}\boldsymbol{x}_j^{(1)}\right)^2+c_1\sum_{k=1}^{m_2}\xi_k$$
$$\text{s.t.}\quad \omega_1^T\boldsymbol{x}_k^{(2)}-\omega_1^T\frac{1}{m_1}\sum_{j=1}^{m_1}\boldsymbol{x}_j^{(1)}+\xi_k\ge 1,\xi_k\ge 0, \tag{1}$$

and

$$\min_{\omega_2,\eta}\frac{1}{2}\sum_{i=1}^{m_2}\left(\omega_2^T\boldsymbol{x}_i^{(2)}-\omega_2^T\frac{1}{m_2}\sum_{j=1}^{m_2}\boldsymbol{x}_j^{(2)}\right)^2+c_2\sum_{k=1}^{m_1}\eta_k$$
$$\text{s.t.}\quad -(\omega_2^T\boldsymbol{x}_k^{(1)}-\omega_2^T\frac{1}{m_2}\sum_{j=1}^{m_2}\boldsymbol{x}_j^{(2)})+\eta_k\ge 1,\eta_k\ge 0, \tag{2}$$

where c_1 and c_2 are both trade-off constraints, and ξ_k and η_k are all nonnegative slack variables.

In order to simplify the above formulations, two within-class scatter matrixes $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, corresponding to class 1 and class 2, respectively, are defined as follows:

$$\boldsymbol{S}_1=\sum_{i=1}^{m_1}(\boldsymbol{x}_i^{(1)}-\frac{1}{m_1}\sum_{k=1}^{m_1}\boldsymbol{x}_k^{(1)})(\boldsymbol{x}_i^{(1)}-\frac{1}{m_1}\sum_{k=1}^{m_1}\boldsymbol{x}_k^{(1)})^T, \tag{3}$$

and

$$\boldsymbol{S}_2=\sum_{i=1}^{m_2}(\boldsymbol{x}_i^{(2)}-\frac{1}{m_2}\sum_{k=1}^{m_2}\boldsymbol{x}_k^{(2)})(\boldsymbol{x}_i^{(2)}-\frac{1}{m_2}\sum_{k=1}^{m_2}\boldsymbol{x}_k^{(2)})^T. \tag{4}$$

Then Eqs. (1) and (2) can be converted to their equivalent formulations shown as follows:

$$\min_{\omega_1,\xi}\frac{1}{2}\omega_1^T\boldsymbol{S}_1\omega_1+c_1\sum_{k=1}^{m_2}\xi_k$$
$$\text{s.t.}\quad \omega_1^T\boldsymbol{x}_i^{(2)}-\omega_1^T\frac{1}{m_1}\sum_{j=1}^{m_1}\boldsymbol{x}_j^{(1)}+\xi_k\ge 1,\xi_k\ge 0, \tag{5}$$

and

$$\min_{\omega_2,\eta}\frac{1}{2}\omega_2^T\boldsymbol{S}_2\omega_2+c_2\sum_{k=1}^{m_1}\eta_k$$
$$\text{s.t.}\quad -(\omega_2^T\boldsymbol{x}_i^{(1)}-\omega_2^T\frac{1}{m_2}\sum_{j=1}^{m_2}\boldsymbol{x}_j^{(2)})+\eta_k\ge 1,\eta_k\ge 0. \tag{6}$$

The Wolf dual problems of Eqs. (5) and (6) have been shown in [12] to be Eqs. (7) and (8) in terms of the Lagrangian multipliers $\boldsymbol{\alpha}\in R^{m_2}$ and $\boldsymbol{\gamma}\in R^{m_1}$, respectively.

$$\min_{\alpha}\frac{1}{2}\boldsymbol{\alpha}^T H\boldsymbol{\alpha}-\boldsymbol{e}_2^T\boldsymbol{\alpha}$$
$$\text{s.t.}\quad 0\le\boldsymbol{\alpha}\le c_1\boldsymbol{e}_2\,, \tag{7}$$

and

$$\min_{\gamma}\frac{1}{2}\gamma^T G\gamma-\boldsymbol{e}_1^T\gamma$$
$$\text{s.t.}\quad 0\le\gamma\le c_2\boldsymbol{e}_1\,, \tag{8}$$

where

$$\boldsymbol{e}_1=(1,...,1)^T\in R^{m_1},\ \boldsymbol{e}_2=(1,\dots,1)^T\in R^{m_2},$$
$$\boldsymbol{H}=(h_{ij})_{m_2\times m_2},\ h_{ij}=(\boldsymbol{x}_i^{(2)}-\frac{1}{m_1}\sum_{k=1}^{m_1}\boldsymbol{x}_k^{(1)})^T\boldsymbol{S}_1^{-1}(\boldsymbol{x}_j^{(2)}-\frac{1}{m_1}\sum_{k=1}^{m_1}\boldsymbol{x}_k^{(1)}),$$
$$\boldsymbol{G}=(g_{ij})_{m_1\times m_1},\ g_{ij}=(\boldsymbol{x}_i^{(1)}-\frac{1}{m_2}\sum_{k=1}^{m_2}\boldsymbol{x}_k^{(2)})^T\boldsymbol{S}_2^{-1}(\boldsymbol{x}_j^{(1)}-\frac{1}{m_2}\sum_{k=1}^{m_2}\boldsymbol{x}_k^{(2)}).$$

By solving the above two dual problems, Lagrangian multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ can be got, and then the two projection axes corresponding to class 1 and class 2 can be attained by Eqs. (9) and (10), respectively.

$$\omega_1=\boldsymbol{S}_1^{-1}\sum_{i=1}^{m_2}\alpha_i(\boldsymbol{x}_i^{(2)}-\frac{1}{m_1}\sum_{k=1}^{m_1}\boldsymbol{x}_k^{(1)}), \tag{9}$$

and

$$\omega_2=-\boldsymbol{S}_2^{-1}\sum_{i=1}^{m_1}\gamma_i(\boldsymbol{x}_i^{(1)}-\frac{1}{m_2}\sum_{k=1}^{m_2}\boldsymbol{x}_k^{(2)}) \tag{10}$$

for testing, the label of a new coming sample $\boldsymbol{x}$ is determined depending on the distance between the projection of $\boldsymbol{x}$ and the projected class mean which is expressed as

$$\text{label}(\boldsymbol{x})=\underset{i=1,2}{\text{argmin}}\{d_i\}=\begin{cases} d_1\Rightarrow & \boldsymbol{x}\in\text{class}\quad 1,\\ d_2\Rightarrow & \boldsymbol{x}\in\text{class}\quad 2,\end{cases} \tag{11}$$

where

$$d_i=\left|\omega_i^T\boldsymbol{x}-\omega_i^T\frac{1}{m_i}\sum_{k=1}^{m_i}\boldsymbol{x}_k^{(i)}\right|\quad\text{and}\quad|\cdot|$$

is the absolute value.

2.2. Least squares projection twin support vector machine

Different from PTSVM, the decision function of LSPTSVM is obtained from the primal problems directly. The primal problems of LSPTSVM are modified versions of the primal problems Eqs. (1) and (2) of PTSVM in least squares sense and constructed following the idea of PSVM proposed in [14]. Different from the primal problems Eqs. (1) and (2) with the inequality constraints, The primal problems of LSPTSVM have only equality constraints as follows:

$$\min_{\omega_1,\xi}\frac{1}{2}\sum_{i=1}^{m_1}\left(\omega_1^T\boldsymbol{x}_i^{(1)}-\omega_1^T\frac{1}{m_1}\sum_{j=1}^{m_1}\boldsymbol{x}_j^{(1)}\right)^2+\frac{c_1}{2}\sum_{k=1}^{m_2}\xi_k^2+\frac{c_3}{2}||\omega_1||^2$$
$$\text{s.t.}\quad \omega_1^T\boldsymbol{x}_k^{(2)}-\omega_1^T\frac{1}{m_1}\sum_{j=1}^{m_1}\boldsymbol{x}_j^{(1)}+\xi_k=1, \tag{12}$$

and

$$\min_{\omega_2,\eta}\frac{1}{2}\sum_{i=1}^{m_2}\left(\omega_2^T\boldsymbol{x}_i^{(2)}-\omega_2^T\frac{1}{m_2}\sum_{j=1}^{m_2}\boldsymbol{x}_j^{(2)}\right)^2+\frac{c_2}{2}\sum_{k=1}^{m_1}\eta_k^2+\frac{c_4}{2}||\omega_2||^2$$