Applied Soft Computing 40 (2016) 1–9

Contents lists available at ScienceDirect

Applied Soft Computing

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

Enhanced algorithm for high-dimensional data classification

Xiaoming Wang [a,*], Shitong Wang [b]

[a] School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[b] School of Digital Media, Jiangnan University, Wuxi 214122, China

ARTICLE INFO

Article history:
Received 25 January 2014
Received in revised form
21 November 2014
Accepted 24 October 2015
Available online 27 November 2015

Keywords:
Machine learning
Supervised learning
Kernel methods
Support vector machine

ABSTRACT

Minimum class variance support vector machine (MCVSVM) and large margin linear projection (LMLP) classifier, in contrast with traditional support vector machine (SVM), take the distribution information of the data into consideration and can obtain better performance. However, in the case of the singularity of the within-class scatter matrix, both MCVSVM and LMLP only exploit the discriminant information in a single subspace of the within-class scatter matrix and discard the discriminant information in the other subspace. In this paper, a so-called twin-space support vector machine (TSSVM) algorithm is proposed to deal with the high-dimensional data classification task where the within-class scatter matrix is singular. TSSVM is rooted in both the non-null space and the null space of the within-class scatter matrix, takes full advantage of the discriminant information in the two subspaces, and so can achieve better classification accuracy. In the paper, we first discuss the linear case of TSSVM, and then develop the nonlinear TSSVM. Experimental results on real datasets validate the effectiveness of TSSVM and indicate its superior performance over MCVSVM and LMLP.

1. Introduction

In the past decade, kernel methods [1] have been widely studied and applied [2–4]. Support vector machine (SVM), as the most well-known representative of kernel methods, is a powerful machine learning method based on Vapnik's Statistical Learning Theory [5] and draws lots of researchers' attentions [6–10]. Different from other methods which usually attempt to minimize the misclassification errors on the training set (empirical risk minimization), SVM minimizes the structural risk, which is the probability of misclassifying a previously unseen sample [5,11]. The essential point of SVM is to find a separating hyperplane which achieves the maximal margin among two different classes of data. The basic idea of SVM has also been used to deal with regression problems and the corresponding method is called support vector regression (SVR) [12], which is a very effective regression method. The regression problem of SVR can actually be viewed as a classification problem of SVM in the dual space. Therefore, SVR can be seen as the most common application form of SVM and they share many same properties, for example the structural risk minimization principle. The one-class SVM algorithm [13] also stems from SVM and is designed as a data description method. One-class SVM shows appealing performance in the field of the outlier detection. It employs the strategy of constructing a hyperplane in the feature space in order to distinguish the normal data points from the given dataset which possibly contains the abnormal ones. Actually, the hyperplane separates the normal data points from the origin with maximum margin. Obviously, one-class SVM embodies the basic idea of SVM.

However, SVM does not take the class distribution into consideration and may result in a non-robust solution [14], i.e., the solution of SVM is easily impacted by outliers or noises. This means that its generalization performance or ability to correctly classify unknown samples may be impaired. In order to overcome the drawback of SVM, a modified class of SVM called minimum class variance support vector machine (MCVSVM) was presented in [14] which is inspired from the optimization of Fisher's discriminant analysis (FDA) [15,16]. Unlike SVM, in which only the samples in the boundaries are taken into consideration, the solution of MCVSVM takes full advantage of both the samples in the boundaries and the distribution of the classes and gives a robust solution. In [17], MCVSVM were further extended to directly solve multiclass classification problems and a novel class of multiclass classifiers called minimum within-class variance multiclass classifiers (MWCVMC) were introduced. Similarly to MCVSVM, MWCVMC is also inspired by multiclass FDA and SVM. Moreover, the authors investigated and solved MWCVMC based on indefinite kernels and dissimilarity measures via pseudo-Euclidean embedding in [17]. In [18], the authors introduced the intrinsic manifold structure of the data space into MCVSVM and a novel method called minimum class locality preserving variance support vector machine MCLPVSVM is proposed. Following the way that MCVSVM incorporates the distribution of the classes by using the within-class covariance matrix, MCLPVSVM incorporates the intrinsic geometry of the data and local structure by using the locality preserving within-class covariance matrix when we define the distance for the sample point to the decision hyperplane. MCLPVSVM are very closely related to MCVSVM and share some properties with the SVM and the MCVSVM.

Similarly to FDA, MCVSVM encounters the singularity of the within-class scatter matrix in the high-dimensional data classification task, e.g., in the small sample size (SSS) problem [19–21]. Here the singularity problem is that the within-class scatter matrix is not invertible but its inverse is necessary during solving the optimization problem of MCVSVM. In order to tackle this problem, a good method is to first perform eigenanalysis to the within-class scatter matrix and then implement MCVSVM only in the non-null space [22,23] which is spanned by the eigenvectors corresponding to non-zero eigenvalues [14]. This method essentially removes the null space which is spanned by the eigenvectors corresponding to zero eigenvalues and dodges the singularity problem. However, as was pointed out in [24], the null space generally contains the most discriminative information. Obviously, it can be observed that MCVSVM is just carried out in a single subspace, which leads to a loss of some significant discriminant information in the high-dimensional data space.

* Corresponding author. Tel.: +86 18782299958.
E-mail address: wangxmwm@gmail.com (X. Wang).

http://dx.doi.org/10.1016/j.asoc.2015.10.051
1568-4946/

On the other hand, the large margin linear projection (LMLP) classifier [25] is rooted only in the null space. LMLP takes full advantage of the singularity of the within-class scatter matrix, and classifies projected points in one-dimensional space by itself. Experimental results indicated the effectiveness of LMLP for the high-dimensional classification problems such as face recognition and image classification. However, LMLP obviously ignore the discriminant information in the non-null space.

So, in the case of the high-dimensional data classification where the singularity of the within-class scatter matrix occurs, both MCVSVM and LMLP only exploit the discriminant information in a single subspace. Thus, when employing MCVSVM or LMLP, it is obvious that certain discriminative information resides in the other subspace which is discarded. So, MCVSVM and LMLP are opposite extremes to each other, and lose the discriminant information residing in the complementary subspace of the within-class scatter matrix since the non-null space and the null space are orthogonal and complementary with respect to discriminative power.

Aiming at the drawbacks of MCVSVM and LMLP, in the paper, we propose a novel classification algorithm called twin-space support vector machine (TSSVM) to deal with the high-dimensional data classification task. Formally, the optimization problem of TSSVM is a combination of the optimizations of MCVSVM and LMLP. In essence, however, TSSVM is different from MCVSVM and LMLP. The key difference between TSSVM and the other two methods is that the former explicitly exploits the information of both the null space and the non-null space of the within-class scatter matrix, whereas the latter two are only rooted in a single subspace. In the paper, we discuss the linear case of TSSVM. After that, we first present a novel alternative version of the nonlinear MCVSVM and develop a kernelization algorithm of LMLP since in [25] the authors did not present the nonlinear LMLP, and then propose the nonlinear TSSVM. Experimental results on real datasets validate the effectiveness of TSSVM and indicate its superior performance over MCVSVM and LMLP.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. In Section 3, the linear case of TSSVM is discussed. In Section 4, an alternative version of the nonlinear MCVSVM and the nonlinear LMLP are first proposed, and then the nonlinear TSSVM is defined and solved. The experimental results are reported in Section 5. Finally, conclusions are drawn in Section 6.

2. Related work

In this paper, we suppose a training dataset which contains two classes of N samples, represented by $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, with training samples $\mathbf{x}_i \in R^M$ and corresponding labels $y_i \in \{1, -1\}$, where $i = 1, \ldots, N$ and M is the dimension of the sample space. Note, in the paper we focus on the classification problem in the high-dimensional sample space where the singularity of the within-class scatter matrix occurs.

2.1. Fundamentals

For the above given training dataset, the within-class scatter matrix $\mathbf{S}_W$ and the between-class scatter matrix $\mathbf{S}_B$ are respectively defined as [24]

$$\mathbf{S}_W = \sum_{k=1}^{2} \sum_{\mathbf{x} \in \mathbf{X}_k} (\mathbf{x} - \mathbf{u}_k)(\mathbf{x} - \mathbf{u}_k)^T \tag{1}$$

$$\mathbf{S}_B = (\mathbf{u}_1 - \mathbf{u}_2)(\mathbf{u}_1 - \mathbf{u}_2)^T \tag{2}$$

where $\mathbf{X}_1 = \{\mathbf{x}_i | y_i = 1, i = 1, \ldots, N\}$, $\mathbf{X}_2 = \{\mathbf{x}_i | y_i = -1, i = 1, \ldots, N\}$, $\mathbf{u}_k = \frac{1}{N_k} \sum_{\mathbf{x} \in \mathbf{X}_k} \mathbf{x}$ and T denotes vector transpose. Here, N_k is the cardinality of $\mathbf{X}_k$.

Definition 1 *([26])*. Suppose $\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M\}$ are M orthonormal eigenvectors of $\mathbf{S}_W$ and the first r ($r = rank\ \mathbf{S}_W$) ones $\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_r\}$ are corresponding to non-zero eigenvalues, then we define the non-null space $\boldsymbol{\Omega}$ and the null space $\boldsymbol{\Psi}$ of $\mathbf{S}_W$ as $\boldsymbol{\Omega} = span\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_r\}$ and $\boldsymbol{\Psi} = span\{\boldsymbol{\beta}_{r+1}, \ldots, \boldsymbol{\beta}_M\}$, respectively.

Lemma 2 ([26]). *For the non-null space $\boldsymbol{\Omega}$ and the null space $\boldsymbol{\Psi}$ of $\mathbf{S}_W$ as defined above, $\boldsymbol{\Psi}$ is orthogonal complement of $\boldsymbol{\Omega}$. Also, any arbitrary $\mathbf{w} \in \mathbf{R}^M$ can be denoted by $\mathbf{w} = \boldsymbol{\zeta} + \boldsymbol{\eta}$, where $\boldsymbol{\zeta} \in \boldsymbol{\Omega}$ and $\boldsymbol{\eta} \in \boldsymbol{\Psi}$.*

2.2. Minimum class variance support vector machine

In order to overcome the drawback of SVM, a modified class of SVM called MCVSVM was proposed in [14], which takes into consideration both the samples in the boundaries and the distribution of the classes and gives a robust solution. In the case where the training samples are not linearly separable the optimization problem of MCVSVM is defined as

$$\begin{aligned} \min_{\mathbf{w},\ b} \quad & \mathbf{w}^T \mathbf{S}_W \mathbf{w} + C \sum_{i=1}^{N} \xi_i \\ s.t. \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \xi_i \geq 0, \quad i = 1, \ldots, N \end{aligned} \tag{3}$$

If the within-class scatter matrix $\mathbf{S}_W$ is nonsingular, similarly to SVM, the optimization problem (3) of MCVSVM could be efficiently solved by switching to its Wolfe dual problem using a Lagrangian formulation of the problem [27]. However, when encountering the singularity of $\mathbf{S}_W$, e.g., in the case of the SSS problem, ones cannot directly use this way to tackle the optimization problem (3). In Section 3.1 we will discuss the case in detail.

2.3. Large margin linear projection (LMLP) classifier

Unlike MCVSVM and various LDA methods which try hard to avoid the singularity of $\mathbf{S}_W$, LMLP takes full advantage of the characteristic. If $\mathbf{S}_W$ is singular, there is at least one non-zero vector $\boldsymbol{\eta}$ such that $\boldsymbol{\eta}^T \mathbf{S}_W \boldsymbol{\eta} = 0$. In this case, LMLP defines its optimization problem as follows [25]

$$\begin{aligned} \max_{\boldsymbol{\eta}} \quad & \boldsymbol{\eta}^T \mathbf{S}_B \boldsymbol{\eta} \\ s.t. \quad & \boldsymbol{\eta}^T \mathbf{S}_W \boldsymbol{\eta} = 0, \quad \boldsymbol{\eta}^T \boldsymbol{\eta} = 1 \end{aligned} \tag{4}$$

Note, $\boldsymbol{\eta}^T \mathbf{S}_W \boldsymbol{\eta} = 0$ implies $\boldsymbol{\eta} \in \boldsymbol{\Psi}$. Thus, the above optimization problem (4) can be rewritten as

$$\begin{aligned} \max_{\boldsymbol{\eta}} \quad & \boldsymbol{\eta}^T \mathbf{S}_B \boldsymbol{\eta} \\ s.t. \quad & \boldsymbol{\eta} \in \boldsymbol{\Psi}, \quad \boldsymbol{\eta}^T \boldsymbol{\eta} = 1 \end{aligned} \tag{5}$$

In the linear case, similarly to SVM, LMLP wants to find a linear decision function or hyperplane to classify the samples. In LMLP, however, the threshold is set to 0. Thus, suppose $\boldsymbol{\eta}^*$ is the optimal solution of the optimization problem (5) of LMLP, then the decision function of LMLP is given as

$$f(\mathbf{x}) = \mathrm{sgn}((\boldsymbol{\eta}^*)^T \mathbf{x}) \tag{6}$$

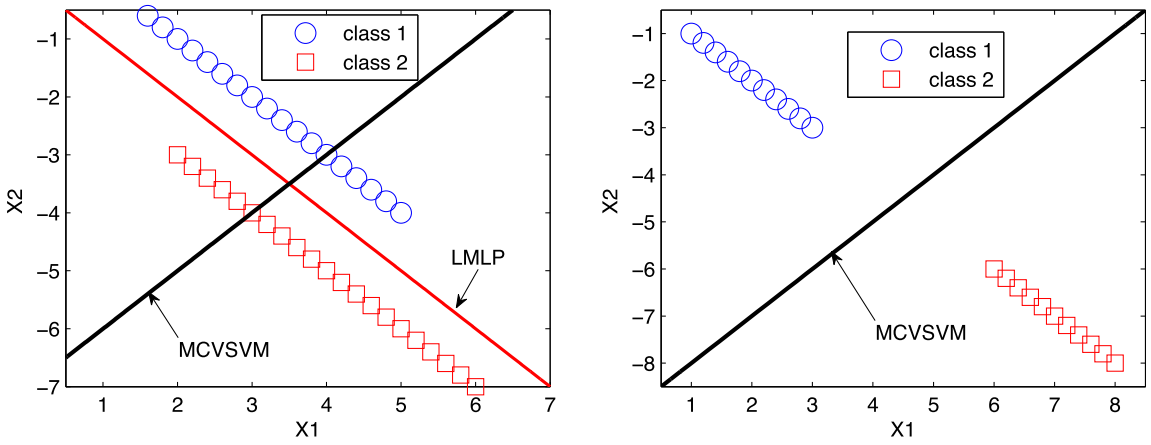
As pointed out in [25], LMLP is a special kind of SVM which assumes that the samples are linearly separable. Thus, LMLP inherits the excellent characteristics of SVM to some extent.

3. Twin-space support vector machine

In this section, we first analyze and illustrate the problems with MCVSVM and LMLP in the case of the singularity of $\mathbf{S}_W$, then define the optimization problem of our method TSSVM. Finally, how to solve the optimization problem is discussed.

3.1. The problems with MCVSVM and LMLP

In the case where $\mathbf{S}_W$ is nonsingular, ones can solve the optimization problem (3) of MCVSVM by its dual optimization problem. When $\mathbf{S}_W$ is singular, however, this way cannot be directly employed to solve (3) because during switching (3) into its dual problem the inverse of $\mathbf{S}_W$ is necessary but it cannot be directly obtained. A method to deal with this case is to employ principal component analysis (PCA) [28] to transform the samples into a low-enough-dimensional subspace where the new within-class scatter matrix is nonsingular, and then the MCVSVM algorithm is carried out in the new subspace. This method needs to predefine the dimensionality of the low-dimensional subspace.



(a) The decision hyperplanes of MCVSVM and LMLP on an artificial dataset. Note, here the decision hyperplane of MCVSVM cannot separate the dataset although it is obviously separable.

(b) The decision hyperplane of MCVSVM on an artificial dataset. Note, here LMLP cannot construct the decision hyperplane although the dataset is obviously separable.

Fig. 1. Illustration of the drawbacks of MCVSVM and LMLP. (a) The decision hyperplanes of MCVSVM and LMLP on an artificial dataset. Note, here the decision hyperplane of MCVSVM cannot separate the dataset although it is obviously separable. (b) The decision hyperplane of MCVSVM on an artificial dataset. Note, here LMLP cannot construct the decision hyperplane although the dataset is obviously separable.

However, we cannot know in advance the dimensionality of the new subspace which can guarantee that the new within-class scatter matrix is nonsingular. The good alternative method is to perform eigenanalysis to the singular matrix $\mathbf{S}_W$ and to remove the eigenvector that corresponds to null eigenvalue [14], i.e., MCVSVM is carried out only in the non-null space $\boldsymbol{\Omega}$ of $\mathbf{S}_W$. In this case the optimization problem (3) of MCVSVM can be reformulated as follows

$$\begin{aligned} &\min_{\boldsymbol{\zeta},\ b,\ \boldsymbol{\xi}} \quad \boldsymbol{\zeta}^T\mathbf{S}_W\boldsymbol{\zeta} + C\sum_{i=1}^{N}\xi_i \\ &s.t. \qquad \boldsymbol{\zeta} \in \boldsymbol{\Omega},\quad y_i(\boldsymbol{\zeta}^T\mathbf{x}_i + b) \geq 1 - \xi_i,\quad \xi_i \geq 0,\quad i = 1, \ldots, N \end{aligned} \tag{7}$$

Obviously, here MCVSVM discards the information in the null space $\boldsymbol{\Psi}$. Therefore, although the singularity problem is dodged in this way, MCVSVM potentially loses some significant discriminant information since the null space $\boldsymbol{\Psi}$ generally contains the most discriminative information [24].

On the other hand, different from MCVSVM, LMLP, which is rooted in the null-space $\boldsymbol{\Psi}$, takes full advantage of the singularity of $\mathbf{S}_W$. In fact, LMLP is the opposite extreme to MCVSVM. However, the problem with LMLP is that it discards the useful discriminant information in the non-null space $\boldsymbol{\Omega}$ of $\mathbf{S}_W$.

Fig. 1 gives an illustration for MCVSVM and LMLP. Fig. 1(a) describes the decision hyperplanes of MCVSVM and LMLP on an artificial dataset. Note, here the artificial dataset is not small size, but the within-class scatter matrix $\mathbf{S}_W$ is singular. So, MCVSVM is carried out in the non-null space $\boldsymbol{\Omega}$ of $\mathbf{S}_W$. As can be seen in Fig. 1(a), the decision hyperplane of MCVSVM cannot separate the two-class dataset although it is obviously separately. This example clearly reveals the drawback of MCVSVM in the case of the singularity of $\mathbf{S}_W$. Here LMLP can appropriately separate the dataset. Fig. 1(b) denotes the drawback of LMLP. Here, the two-class dataset is obviously separable. However, LMLP cannot construct the decision hyperplane. The reason is that in the null space $\boldsymbol{\Omega}$ of $\mathbf{S}_W$ the two-class samples is not separable at all although in the original sample space it are obviously separable. Note, in this example MCVSVM can separate the dataset.

From the above illustration, it can be observed that both MCVSVM and LMLP are carried out only in a single subspace, which leads to a loss of some significant discriminant information since different subspace is complementary with respect to discriminative power.

3.2. The proposed method

From the above discussion, it can easily be seen that there exists deficiency in both MCVSVM and LMLP because they are carried out only in a single subspace of $\mathbf{S}_W$ and discard the discriminant information in the other subspace. So, in order to fully utilize the information of both the null space and non-null space, we define the optimization problem of TSSVM as

$$\begin{cases} \min\limits_{\boldsymbol{\zeta},\ b,\ \boldsymbol{\xi}} \quad \boldsymbol{\zeta}^T\mathbf{S}_W\boldsymbol{\zeta} + C\sum\limits_{i=1}^{N}\xi_i, \\ s.t. \qquad \boldsymbol{\zeta} \in \boldsymbol{\Omega},\quad y_i(\boldsymbol{\zeta}^T\mathbf{x}_i + b) \geq 1 - \xi_i,\quad \xi_i \geq 0,\quad i = 1, \ldots, N \\ \text{and} \\ \max\limits_{\boldsymbol{\eta}} \qquad \boldsymbol{\eta}^T\mathbf{S}_B\boldsymbol{\eta},\quad s.t. \quad \boldsymbol{\eta} \in \boldsymbol{\Psi},\quad \boldsymbol{\eta}^T\boldsymbol{\eta} = 1 \end{cases} \tag{8}$$

Obviously, the above optimization problem consists of the optimization problems of MCVSVM and LMLP. However, the distinguishing feature of TSSVM is that it incorporates the information of the null space $\boldsymbol{\Psi}$ and the non-null space $\boldsymbol{\Omega}$. Nevertheless, MCVSVM only considers the information of the non-null space and LMLP is rooted only in the null space. This is the essential difference between TSSVM and the latter two methods.

Suppose $\{\boldsymbol{\zeta}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\eta}^*\}$ is the optimal solution to the optimization problem (8) of TSSVM, then we define the decision function of

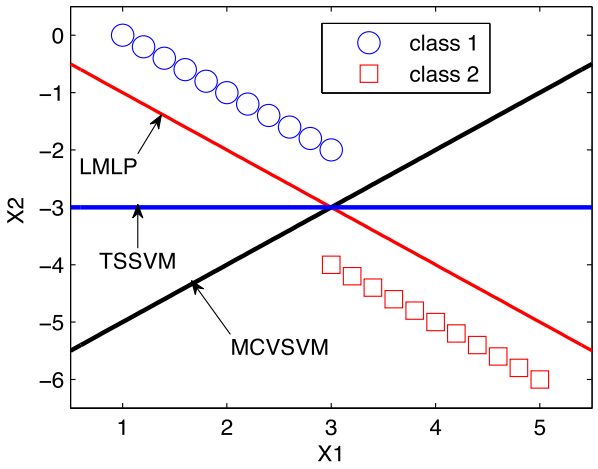


Fig. 2. Illustration of the decision hyperplanes generated by TSSVM, MCVSVM, and LMLP on an artificial dataset.

TSSVM as

$$f(\mathbf{x}) = \mathrm{sgn}(\lambda f_1(\mathbf{x}) + (1-\lambda) f_2(\mathbf{x})) = \mathrm{sgn}(\lambda((\boldsymbol{\zeta}^*)^T\mathbf{x} + b^*) + (1-\lambda)(\boldsymbol{\eta}^*)^T\mathbf{x}) = \mathrm{sgn}((\lambda\boldsymbol{\zeta}^* + (1-\lambda)\boldsymbol{\eta}^*)^T\mathbf{x} + \lambda b^*) \quad (9)$$

where $f_1(\mathbf{x}) = (\boldsymbol{\zeta}^*)^T\mathbf{x} + b^*$ and $f_2(\mathbf{x}) = (\boldsymbol{\eta}^*)^T\mathbf{x}$, which are respectively obtained by the first optimization problem and the second one of (8). Obviously, TSSVM integrates MCVSVM and LMLP and takes into full consideration the information of both the null space and non-null space. Note, here $0 \le \lambda \le 1$. And if $\lambda = 1$, (9) turns into the decision function of MCVSVM. The other extreme case is that (9) becomes the decision function of LMLP when $\lambda = 0$. In Section 5.1, we investigated the influence of λ on the performance of the proposed method TSSVM through conducting experiments. Generally, we can employ cross-validation [29] to find a suitable value of λ. Obviously, this will lead to spending more training time. However, through the experiments, we find that if $\lambda = 0.5$ ones can also obtain good experimental results which may not be the best. Therefore, for simplicity, we can choose $\lambda = 0.5$. The chief advantage of this strategy is that we can avoid the time-consuming cross-validation process.

In TSSVM, actually, we want to find a decision hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ or function $f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T\mathbf{x} + b)$ to separate the dataset. According to Lemma 2, any arbitrary $\mathbf{w} \in \mathbf{R}^M$ can be denoted by $\mathbf{w} = \boldsymbol{\zeta} + \boldsymbol{\eta}$, where $\boldsymbol{\zeta} \in \boldsymbol{\Omega}$ and $\boldsymbol{\eta} \in \boldsymbol{\Psi}$. So, the decision function $f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T\mathbf{x} + b)$ can be rewritten as $f(\mathbf{x}) = \mathrm{sgn}((\boldsymbol{\zeta} + \boldsymbol{\eta})^T\mathbf{x} + b)$. In TSSVM, it can be found that $\boldsymbol{\zeta}$ and b derive from the non-null space, which can be obtained from the first part of (8), and $\boldsymbol{\eta}$ does from the null space, which can be got from the second part of (8). However, MCVSVM simplifies the decision function as $f(\mathbf{x}) = \mathrm{sgn}(\boldsymbol{\zeta}^T\mathbf{x} + b)$ and LMLP does as $f(\mathbf{x}) = \mathrm{sgn}(\boldsymbol{\eta}^T\mathbf{x})$. Therefore, TSSVM constructs the decision function by a more comprehensive way in contrast with MCVSVM and LMLP.

Fig. 2 describes the decision hyperplanes of TSSVM, MCVSVM, and LMLP on an artificial dataset. Here we set $\lambda = 0.5$. As can be seen from the case illustrated in Fig. 2, the decision hyperplane of TSSVM reflects the information in the null space and the non-null space of the within-class scatter matrix $\mathbf{S}_W$ and it shows more reasonable.

3.3. Solving the optimization problem

Obviously, the optimization problem (8) of TSSVM can be efficiently solved by tackling separately each sub-problem which actually denotes the optimization problem of MCVSVM or LMLP.

Firstly, we will deal with the first sub-problem of (8). Since $\mathbf{S}_W$ has r non-zero eigenvectors and $\boldsymbol{\Omega}$ is the non-null space which is spanned by the non-zero eigenvectors of $\mathbf{S}_W$, then, by linear algebra theory, $\boldsymbol{\Omega}$ is isomorphic to r-dimensional Euclidean space $\mathbf{R}^r$ [19,26]. Suppose $\boldsymbol{\zeta} \in \boldsymbol{\Omega}$ and $\mathbf{v} \in \mathbf{R}^r$, then $\boldsymbol{\zeta} = \mathbf{P}\mathbf{v}$, where the column vectors of $\mathbf{P}$ are eigenvectors corresponding to non-zero eigenvectors of $\mathbf{S}_W$, i.e., $\mathbf{P} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_r]$. Thus, the first sub-problem of (8) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{v},\ b,\ \boldsymbol{\xi}} \quad \mathbf{v}^T S_W \mathbf{v} + C\sum_{i=1}^{N}\xi_i, \\ &s.t. \qquad y_i(\mathbf{v}^T\mathbf{z}_i + b) \ge 1 - \xi_i, \quad \xi_i \ge 0, \quad i = 1, \ldots, N \end{aligned} \quad (10)$$

where $\mathbf{v} \in \mathbf{R}^r$, $\mathbf{z}_i = \mathbf{P}^T\mathbf{x}_i$ and $\tilde{\mathbf{S}}_W = \mathbf{P}^T\mathbf{S}_W\mathbf{P}$. Note, here $\tilde{\mathbf{S}}_W$ is non-singular, and so we can solve (10) in its dual optimization problem like MCVSVM in the case where the $\mathbf{S}_W$ is non-singular. Secondly, for the second sub-problem of (10), we can easily find and directly give the optimal solution. First, it can be reformulated as [25]

$$\begin{aligned} &\min_{\boldsymbol{\eta}} \quad \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta}, \\ &s.t. \quad y_i\boldsymbol{\eta}^T\mathbf{x}_i = 1, \quad i = 1, \ldots, N \end{aligned} \quad (11)$$

For the optimization above problem, its primal Lagrangian is

$$L = \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta} - \sum_{i=1}^{N}\alpha_i y_i \boldsymbol{\eta}^T\mathbf{x}_i \quad (12)$$

where the vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T (\boldsymbol{\alpha} \in R^N)$ is the Lagrangian multiplier for the constraints in (11). By differentiating with respect to and using the Karush–Kuhn–Tucker (KKT) conditions [27], the following holds

$$\boldsymbol{\eta} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i = \mathrm{X}\mathrm{Y}\alpha \quad (13)$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{Y} = diag(y_1, \ldots, y_N) = \begin{bmatrix} y_1 & & \\ & \ddots & \\ & & y_N \end{bmatrix}$.

Obviously, $\mathbf{Y} = \mathbf{Y}^T = \mathbf{Y}^{-1}$. According to $y_i\boldsymbol{\eta}^T\mathbf{x}_i = 1$, we have $\boldsymbol{\eta}^T\mathbf{X}\mathbf{Y} = \mathbf{e}^T$. Here $\mathbf{e} = [1, \ldots, 1]^T \in \mathbf{R}^N$. So, the following holds by (13)

$$(\mathrm{X}\mathrm{Y}\alpha)^T\mathbf{X}\mathbf{Y} = \mathbf{e}^T \quad (14)$$

This is

$$\mathbf{Y}\mathbf{X}^T\mathrm{X}\mathrm{Y}\alpha = \mathbf{e} \quad (15)$$

If $\mathbf{X}^T\mathbf{X}$ is non-singular, we have

$$\boldsymbol{\alpha} = (\mathbf{Y}\mathbf{X}^T\mathbf{X}\mathbf{Y})^{-1}\mathbf{e} = \mathbf{Y}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{Y}\mathbf{e} \quad (16)$$

Thus, according to (13) and (16), the optimal solution $\boldsymbol{\eta}^*$ of (11) is given as

$$\boldsymbol{\eta}^* = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i = \mathrm{X}\mathrm{Y}\alpha = \mathbf{X}\mathbf{Y}\mathbf{Y}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{Y}\mathbf{e} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{Y}\mathbf{e} \quad (17)$$

Finally, suppose $\{\mathbf{v}^*, b^*, \boldsymbol{\xi}^*\}$ and $\{\boldsymbol{\eta}^*\}$ are respectively the optimal solutions of (10) and (11), then the corresponding decision function of TSSVM is formulated as

$$f(\mathbf{x}) = \mathrm{sgn}(\lambda(\mathbf{v}^*)^T\mathbf{z} + (1-\lambda)(\boldsymbol{\eta}^*)^T\mathbf{x} + \lambda b^*) \quad (18)$$

where $\mathbf{z} = \mathbf{P}^T\mathbf{x}$. Here $\mathbf{P}$ consists of eigenvectors corresponding to non-zero eigenvectors of $\mathbf{S}_W$.

4. The nonlinear case

In the previous discussion, the derived decision function (or hyperplane) of TSSVM is a linear form. In this section, we first propose an alternative method to the nonlinear MCVSVM, then develop the nonlinear LMLP since in [25] the authors discussed only the linear LMLP, finally define and solve the optimization problem of the nonlinear TSSVM.

4.1. An alternative to the nonlinear MCVSVM

In the nonlinear case, ones seek to use the kernelization trick [30] to map the M-dimensional sample space into a high-dimensional feature space, where a linear hyperplane corresponds to a nonlinear hyperplane in the original sample space. Without loss of generality, the optimization problem of MCVSVM in the feature space is defined as

$$\begin{aligned} \min_{\mathbf{w},\ b} \quad & \mathbf{w}^T\mathbf{S}_W^{\varphi}\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \end{aligned} \tag{19}$$

where $\varphi(\mathbf{x}_i)$ $(i = 1, \ldots, N)$ denotes the samples in the feature space and $\mathbf{S}_W^{\varphi}$ is the corresponding within-class scatter matrix.

Note, (19) could not directly be solved because the within-class scatter matrix $\mathbf{S}_W^{\varphi}$ is generally singular in the high-dimensional feature space. In [14], the authors proposed to employ PCA to transform the samples in the feature space into a low-dimensional space where the new within-class scatter matrix is nonsingular, and then the linear MCVSVM is applied in the low-dimensional space. This procedure is in nature to transform the samples in the original space using KPCA [24] into a new space, and then the linear MCVSVM is carried out in the new space [14]. Obviously, analogously to the linear case, here the dimensionality of the new space needs to be predefined, which can guarantee that the new within-class scatter matrix is nonsingular. But, it cannot be known in advance.

So, here we will propose a novel nonlinear MCVSVM. First, let $\mathbf{X}^{\varphi} = [\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \ldots, \varphi(\mathbf{x}_N)]$, according to [31] the within-class scatter matrix $\mathbf{S}_W^{\varphi}$ can be rewritten as

$$\mathbf{S}_W^{\varphi} = \mathbf{X}^{\varphi}\mathbf{L}(\mathbf{X}^{\varphi})^T \tag{20}$$

where $\mathbf{L} = \mathbf{I} - \mathbf{W}$. Here $\mathbf{I}$ is a identity matrix, and $\mathbf{W}$ is defined as

$$\mathbf{W}_{ij} = \begin{cases} 1/N_k, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the } k\text{th class} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Then, according to the representation theorem for Reproducing Kernel Hilbert Spaces [1], of which the vector $\mathbf{w}$ can be formulated as

$$\mathbf{w} = \sum_{i=1}^{N} a_i\varphi(\mathbf{x}_i) \tag{22}$$

where $a_i \in \mathbf{R}$. Thus, by using (20) and (22), the optimization problem (19) can be reformulated as

$$\begin{aligned} \min_{\mathbf{a},\ b} \quad & \mathbf{a}^{\mathrm{T}}\mathbf{KLKa} + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(\mathbf{a}^{\mathrm{T}}\mathbf{k}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \end{aligned} \tag{23}$$

where $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{N\times N}$ is the kernel matrix, the vectors $\mathbf{k}_i$ and $\mathbf{a}$ are defined as $\mathbf{k}_i = [k(\mathbf{x}_i, \mathbf{x}_1), k(\mathbf{x}_i, \mathbf{x}_2), \ldots, k(\mathbf{x}_i, \mathbf{x}_N)]^T$ and $\mathbf{a} = [a_1, \ldots, a_N]^T$, respectively. Here $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)$ is a predefined kernel function. Let $\mathbf{H} = \mathbf{KLK}$, the above optimization problem (23) can be further written as

$$\begin{aligned} \min_{\mathbf{a},\ b} \quad & \mathbf{a}^{\mathrm{T}}\mathbf{Ha} + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(\mathbf{a}^{\mathrm{T}}\mathbf{k}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \end{aligned} \tag{24}$$

It should be noted that the above optimization problem (24) actually is a optimization problem defined by the linear MCVSVM since $\mathbf{H} = \mathbf{KLK}$ is the within-class scatter matrix of the samples which consist of $\mathbf{k}_i (i = 1, \ldots, N)$. So, it can be efficiently solved according to the previous discussion even if $\mathbf{H}$ is singular. In essence, our method of dealing with the nonlinear case of MCVSVM is equivalent to transform the nonlinear MCVSVM into a new linear MCVSVM. Here our method has two fold advantages: (1) It does not employ KPCA and may save the time; (2) It does not need to predefine the dimensionality of the new dimensional space.

Suppose $\{\boldsymbol{\alpha}^*, b^*\}$ solves the above optimization problem (24), then we can obtain the corresponding decision function of the nonlinear MCVSVM as follows

$$f(\mathbf{x}) = (\boldsymbol{\alpha}^*)^T\mathbf{k} + b^* \tag{25}$$

where $\mathbf{k} = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x})]^T$.

4.2. The nonlinear LMLP

In [25], the authors did not present the corresponding nonlinear version of LMLP. In this subsection we will develop the nonlinear LMLP. In the high-dimensional feature space, analogously to the linear case of LMLP in the original sample space, we define the optimization problem of nonlinear LMLP as

$$\begin{aligned} \max_{\boldsymbol{\eta}} \quad & \boldsymbol{\eta}^T\mathbf{S}_B^{\varphi}\boldsymbol{\eta} \\ s.t. \quad & \boldsymbol{\eta} \in \boldsymbol{\Psi}, \quad \boldsymbol{\eta}^T\boldsymbol{\eta} = 1 \end{aligned} \tag{26}$$

where $\boldsymbol{\Psi}$ denotes the null space of $\mathbf{S}_W^{\varphi}$ which is the within-class scatter matrix in the feature space. Through a series of transformation, as in the linear case of LMLP, the above optimization problem (26) can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{\eta}} \quad & \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta} \\ s.t. \quad & y_i\boldsymbol{\eta}^T\boldsymbol{\phi}(\mathbf{x}_i) = 1, \quad i = 1, \ldots, N \end{aligned} \tag{27}$$

Similarly, we can obtain the optimal solution of (27). It can be formulated as

$$\boldsymbol{\eta}^* = \sum_{i=1}^{N}\alpha_i y_i\boldsymbol{\phi}(\mathbf{x}_i) = \mathbf{X}^{\boldsymbol{\phi}}((\mathbf{X}^{\boldsymbol{\phi}})^T\mathbf{X}^{\boldsymbol{\phi}})^{-1}\mathbf{Ye} \tag{28}$$

where $\mathbf{X}^{\varphi} = [\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2), \ldots, \varphi(\mathbf{x}_N)]$. Thus, the decision function of the nonlinear LMLP can be written as

$$f(\mathbf{x}) = \mathrm{sgn}((\boldsymbol{\eta}^*)^T\boldsymbol{\phi}(\mathbf{x})) \tag{29}$$

However, if we first find the optimal solution $\boldsymbol{\eta}^*$ according to (28) and then classify a test sample by (29), it may be encountered that $\boldsymbol{\eta}^*$ is infinite dimension since $\varphi(\mathbf{X})$ is perhaps infinite dimension. Actually, we can directly reformulate the decision function of the nonlinear LMLP. Let $\mathbf{K} = (\mathbf{X}^{\boldsymbol{\phi}})^T\mathbf{X}^{\boldsymbol{\phi}} = (k(\mathbf{x}_i, \mathbf{x}_j))_{N\times N}$ and $\mathbf{k} = (\boldsymbol{\phi}^T(\mathbf{x})\mathbf{X}^{\boldsymbol{\phi}})^T = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x})]^T$, thus $(\boldsymbol{\eta}^*)^T\boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\phi}^T(\mathbf{x})\boldsymbol{\eta}^* = \boldsymbol{\phi}^T(\mathbf{x})\mathbf{X}^{\boldsymbol{\phi}}((\mathbf{X}^{\boldsymbol{\phi}})^T\mathbf{X}^{\boldsymbol{\phi}})^{-1}\mathbf{Ye} = \mathbf{k}^T\mathbf{K}^{-1}\mathbf{Ye}$ holds. So, the decision function of the nonlinear LMLP can be rewritten as

$$f(\mathbf{x}) = \mathrm{sgn}((\boldsymbol{\eta}^*)^T\boldsymbol{\phi}(\mathbf{x})) = \mathrm{sgn}(\mathbf{k}^T\mathbf{K}^{-1}\mathbf{Ye}) \tag{30}$$

4.3. The nonlinear TSSVM

For the nonlinear case, we employ a similar way which is done in the linear case and define the optimization problem of nonlinear TSSVM as

$$\begin{cases} \min\limits_{\zeta,\ b,\ \xi} & \frac{1}{2}\zeta^T \mathbf{S}_W^{\varphi}\zeta + C\sum\limits_{i=1}^{N}\xi_i, \\ s.t. & \zeta \in \boldsymbol{\Omega},\quad z_i(\zeta^T\varphi(\mathbf{x}_i)+b) \geq 1-\xi_i,\quad \xi_i \geq 0,\quad i=1,\ldots,N \\ \text{and} & \\ \max\limits_{\eta} & \eta^T\mathbf{S}_B^{\varphi}\eta,\quad s.t.\quad \eta \in \boldsymbol{\Psi},\quad \eta^T\eta = 1 \end{cases} \tag{31}$$

where $\boldsymbol{\Omega}$ and $\boldsymbol{\Psi}$ denote respectively the non-null space and the null space of $\mathbf{S}_W^{\varphi}$.

Note, firstly, the first sub-problem of (31) can be rewritten as (24), which is a linear MCVSVM problem and we have discussed how to solve it. Secondly, for the second sub-problem of (31), which is optimization problem of the nonlinear LMLP, we have obtained the corresponding decision function as (30). Thus, suppose $\{\boldsymbol{\alpha}^*, b^*\}$ solves the above optimization problem (24), then we can obtain the decision function of the nonlinear TSSVM as

$$f(\mathbf{x}) = \mathrm{sgn}(\lambda(\mathbf{a}^*)^T\mathbf{k} + \lambda b^* + (1-\lambda)\mathbf{k}^T\mathbf{K}^{-1}\mathbf{Y}\mathbf{e}) \tag{32}$$

where $\mathbf{k} = [k(\mathbf{x}_1,\mathbf{x}), k(\mathbf{x}_2,\mathbf{x}), \ldots, k(\mathbf{x}_N,\mathbf{x})]^T$.

5. Experiments

In this section, the experimental results will be reported. The experimental results in [14,25] have shown the advantages of MCVSVM and LMLP over the standard SVM, so here we will compare our method with MCVSVM and LMLP.

Note, since the paper focuses on the classification problem of the high-dimensional data classification task where the singularity of the within-class scatter matrix is encountered, we conduct the experiments on face dataset and image dataset where the SSS problem often occurs and the within-class scatter matrix is generally singular. In the first experiment we investigate the influence of the parameter λ in (9) on TSSVM performance. In the second experiment, we report the evaluation results of TSSVM in comparison with SVM and MCVSVM on the CMU PIE face database [32]. At last, we report the experimental results on the COIL-20 dataset [33].

In the paper, only the binary classification tasks are discussed. The multi-class case can be dealt with by one-against-all [34].

5.1. Parameter influence on performance of TSSVM

In contrast with LMLP and MCVSVM on the single parameter, TSSVM introduces one additional parameter λ as shown in (9). In order to illustrate the influence of the hyperplane of TSSVM, we first test TSSVM with different parameter value of λ on an artificial dataset which is used in Fig. 3. Note, here the dataset is not high-dimensional, but the matrix $\mathbf{S}_W$ is singular. So, MCVSVM is carried out only in the non-null space. From Fig. 3, it can be found that the choice of λ influences obviously the hyperplane. If $\lambda > 0.5$, the decision hyperplane of TSSVM is closer to one of MCVSVM, and when $\lambda < 0.5$ it is closer to which is constructed by LMLP. This result also shows that TSSVM incorporates the information both in the non-null space and in the null space.

In order to further investigate the influence of the parameter on classification accuracy of TSSVM, we conduct experiments on the Yale face database [35] with different parameter value of λ in the linear case. In [31], the original images were manually aligned (two eyes were aligned at the same position), cropped, and then resized to 64×64 pixels. We choose randomly a subset with Ntr(= 4, 5, 6, 7) images per individual with labels to form the training set, and the rest of the database was considered to be the test set. In the experiments, the regularization C parameter was selected from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ for SVM and TSSVM. Note, no parameter needs to set for LMLP in the linear case. We repeated this process 50 times and computed the mean classification accuracy. The experimental results are shown in Table 1. From the experimental results, it can be found that the choice of λ plays an important role. One maybe considers that it is the most reasonable if setting $\lambda = 0.5$, but in fact the optimal value of λ which gets the best classification accuracy is not always $\lambda = 0.5$ according to the experimental results on the many real datasets. This implies that TSSVM is an independent algorithm although its optimization problem is constituted of ones of MCVSVM and LMLP. However, for simplicity, generally we can set $\lambda = 0.5$ and also obtain the acceptable results.

5.2. CMU PIE face dataset

The CMU PIE face database, which comes from Irvine Repository of machine learning databases of the well-known University of California (UCI) [32], contains 68 individuals with 41, 368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes under varying pose, illumination, and expression. Before the experiments, generally, preprocessing to locate the faces was applied. In [31] the original images were manually aligned (two eyes were aligned at the same position), cropped, and then re-sized to 32×32 pixels, with 256 gray levels per pixel. Each image is represented by a 1024-dimensional vector in image space. More details can be found in [31]. The databases in Matlab format after being preprocessed is available at http://www.cad.zju.edu.cn/home/dengcai/Data/data.html. In our experiments, we first choose the face images of the first ten individuals, and then select randomly 30 ones in each individual. Thus, we get the experimental dataset which contains 10 individuals and each individual has 30 face images. This dataset is high-dimensional and here the singularity of the within-class scatter matrix is encountered.

At present, choosing the parameters for the kernel methods such as SVM is an open problem. In general, the algorithms parameters are manfully set. In order to evaluate the performance, a strategy, as is pointed out and adopted in [34], is that a set of the parameters is first given and then the best mean rate among the set is used to estimate the generalized performance. In this work we adopted this strategy.

In the experiments, the regularization parameter C is selected from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ for SVM and TSSVM. Note, no parameter needs to set for the linear LMLP. In the nonlinear case, the typical kernel used in our experiments is the Gaussian kernel, i.e., $k(\mathbf{u},\mathbf{v}) = \exp(-\|\mathbf{u}-\mathbf{v}\|^2/2\sigma^2)$, where σ is the spread of the Gaussian kernel and called kernel parameter. Note, here the kernel parameter is selected from the set $\{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}\}$ for all algorithms. And for the nonlinear SVM and the nonlinear TSSVM the regularization parameter C is selected from the same set as the linear case.

In the experiments, a random subset with Ntr(= 5, 10, 15, 20) images per individual was taken with labels to form the training set, and the rest of the dataset was considered to be the test set. In each experiment, we calculated the best classification accuracy of each algorithm on different parameter pair. We repeated this process 50 times and computed the mean classification accuracy and the corresponding standard deviation. Note, for the sake of simplicity, here we set $\lambda = 0.5$ for TSSVM.

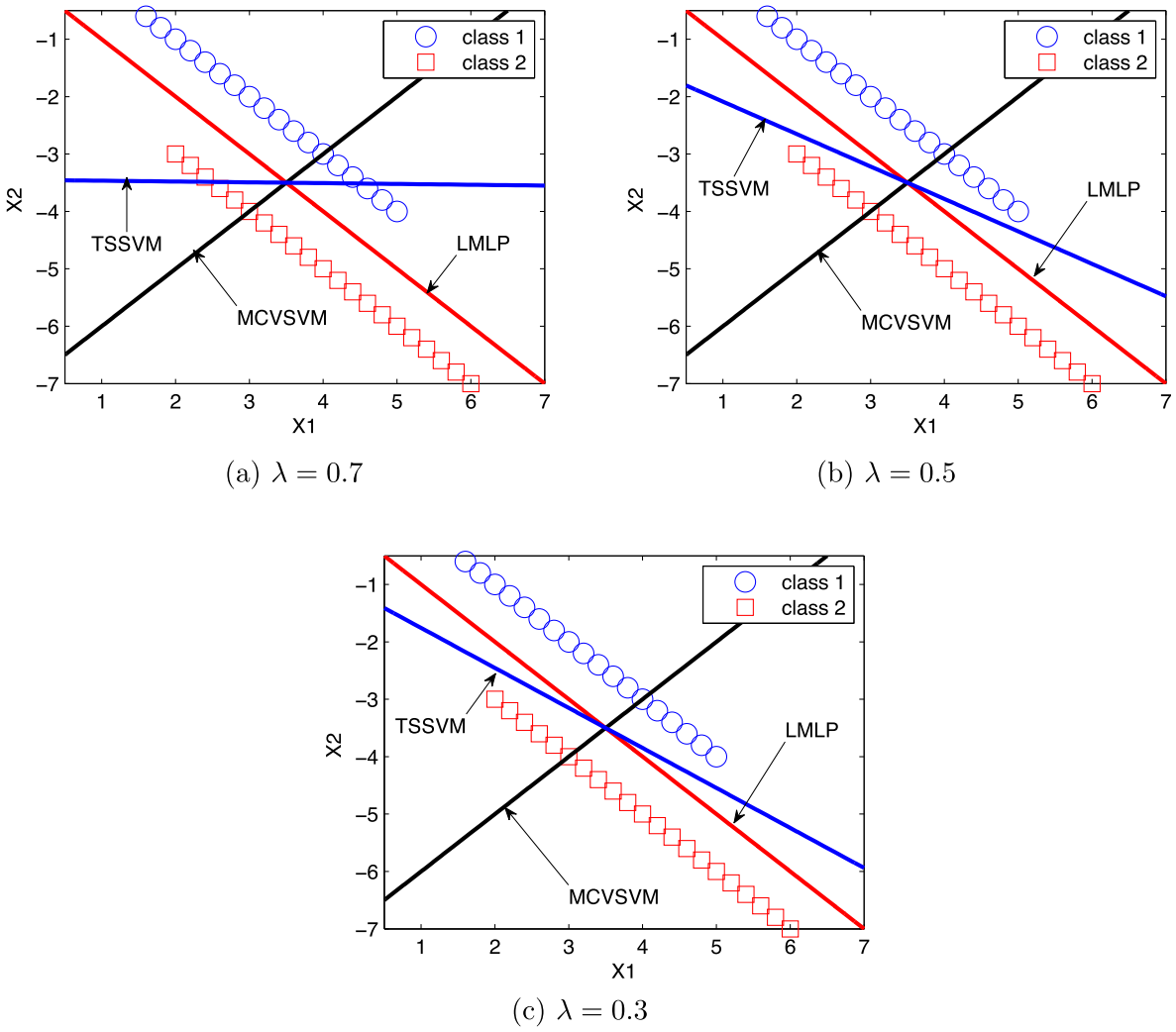


Fig. 3. Illustration of the decision hyperplanes generated by LMLP, MCVSVM, and TSSVM on the different λ.

Table 1
Mean accuracy on the Yale dataset in the linear case.

N_{tr} (number of train samples)	MCVSVM	LMLP	TSSVM		
			λ = 0.2	λ = 0.5	λ = 0.8
4Train	72.16	76.43	73.33	75.26	**78.17**
5Train	73.58	78.29	75.01	**79.36**	76.89
6Train	75.23	76.19	**80.14**	77.14	77.14
7Train	78.88	78.88	79.31	81.52	**82.26**

The better experimental results are highlighted in bold font.

Table 2
Mean accuracy (%) and standard deviation on the CMU PIE face dataset in the linear case.

N_{tr} (number of train samples)	MCVSVM	LMLP	TSSVM
5Train	81.20 ± 0.0180	82.68 ± 0.0328	**85.48** ± 0.0331
10Train	87.60 ± 0.0159	90.25 ± 0.0153	**92.60** ± 0.0144
15Train	91.20 ± 0.0111	91.66 ± 0.0197	**93.20** ± 0.0197
20Train	94.00 ± 0.0260	94.40 ± 0.0200	**96.50** ± 0.0224

The better experimental results are highlighted in bold font.

Table 3
Mean accuracy (%) and standard deviation on the CMU PIE face dataset in the nonlinear case.

N_{tr} (number of train samples)	MCVSVM	LMLP	TSSVM
5Train	83.68 ± 0.0179	82.08 ± 0.0219	**85.32** ± 0.0236
10Train	90.90 ± 0.0152	89.40 ± 0.0128	**92.40** ± 0.0131
15Train	92.33 ± 0.0073	91.60 ± 0.0067	**94.86** ± 0.0049
20Train	95.53 ± 0.0120	94.86 ± 0.0120	**97.45** ± 0.0120

The better experimental results are highlighted in bold font.

Table 4
Mean accuracy (%) and standard deviation on the COIL-20 dataset in the linear case.

N_{tr} (number of train samples)	MCVSVM	LMLP	TSSVM
30Train	86.52 ± 0.0218	85.66 ± 0.0232	**88.16** ± 0.0184
40Train	88.68 ± 0.0043	88.40 ± 0.0094	**90.04** ± 0.0095
50Train	88.02 ± 0.0077	89.59 ± 0.0095	**90.91** ± 0.0117
60Train	89.95 ± 0.0111	90.27 ± 0.0062	**92.57** ± 0.0065

The better experimental results are highlighted in bold font.

Table 5
Mean accuracy (%) and standard deviation on the COIL-20 dataset in the nonlinear case.

N_{tr} (number of train samples)	MCVSVM	LMLP	TSSVM
30Train	97.17 ± 0.0086	95.72 ± 0.0014	**98.96** ± 0.0028
40Train	98.14 ± 0.0017	96.90 ± 0.0011	**98.95** ± 0.0011
50Train	99.07. ± 0.0023	98.06 ± 0.0015	**99.85** ± 0.0039
60Train	99.81 ± 0.0018	98.98 ± 0.0064	**100.00** ± 0.0000

The better experimental results are highlighted in bold font.

Table 6
Mean accuracy (%) and standard deviation on the selected bioinformatics datasets.

Dataset	MCVSVM	LMLP	TSSVM
Leukemia	92.67 ± 0.0592	93.18 ± 0.0650	**96.27** ± 0.0463
Colon	**81.20** ± 0.0621	63.15 ± 0.0645	80.83 ± 0.0436
Lymphoma	91.52 ± 0.0393	92.91 ± 0.0375	**95.75** ± 0.0217
Prostate	91.61 ± 0.0387	90.97 ± 0.0316	**93.81** ± 0.0523

The better experimental results are highlighted in bold font.

Tables 2 and 3 show the experimental results in the linear case and the nonlinear case, respectively. As can be seen, in the linear case, LMLP achieves better classification accuracy than MCVSVM since the null space contains the most discriminative information, however, the proposed method performs better than the other two methods on the whole. The reason is that our method exploits the information both in the non-null space and in the null space. In the nonlinear case, TSSVM also obtains the highest mean classification accuracy. It is worthwhile to note that MCVSVM performs better in contrast with LMLP in the nonlinear case although LMLP does better in the linear case. The reason may be that in the nonlinear case more discriminative information resides in the non-null space of within-class scatter matrix $\mathbf{S}_W^{\varphi}$. Nevertheless, our method TSSVM takes full advantage of useful discriminant information in the two spaces and so performs best.

5.3. COIL-20 dataset

The COIL-20 dataset [33] contains 20 objects. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 greyscale images. Every image is 128×128 pixels, leading to 16,384 dimensions. In [31], the size of each image is re-sized to 32×32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector. The dataset in Matlab format after being preprocessed is available at: http://www.cad.zju.edu.cn/home/dengcai/Data/data.html.

In the experiments, a random subset with $Ntr(= 30, 40, 50, 60)$ images per object was taken with labels to form the training set, and the rest of the dataset was considered to be the test set. All experimental procedures are the same as Section 5.2.

Tables 4 and 5 report the experimental results in the linear case and the nonlinear case, respectively. Obviously, in contrast with MCVSVM and LMLP, the proposed method TSSVM achieves higher mean classification accuracy and performs better on the whole in the both linear and nonlinear cases. The experimental results demonstrate further that it is helpful to improve the classification accuracy in the high-dimensional data space by incorporating the information both in the non-null space and in the null space. This idea is embodied in our method.

5.4. Bioinformatics datasets

In order to further evaluate the proposed method, we also performed experiments on several bioinformatics datasets. The selected datasets include the leukemia dataset [36], the colon cancer dataset [36], the prostate cancer dataset [37] and the lymphoma dataset [38]. The leukemia dataset contains measurements corresponding to acute lymphoblast leukemia (ALL) and acute myeloid leukemia (AML) samples from bone marrow and peripheral blood. The dataset consists of 72 samples: 25 samples of AML, and 47 samples of ALL. Each sample is measured over 7129 genes. The colon cancer dataset consists of 22 normal and 40 tumor tissue samples and each sample contains 2000 genes. The lymphoma dataset contains expression levels of 7129 genes taken over 58 diffuse large B-cell lymphoma samples and 19 follicular lymphoma samples. The prostate cancer dataset includes 52 tumor samples and 50 control samples and each sample is measured over 2135 genes.

In the experiments, we took 70% samples of each dataset to form the training set, and the rest was considered to be the test set. Here we adopted the linear kernel. All experimental procedures are the same as Section 5.2. We repeated this process 50 times and computed the mean classification accuracy and the corresponding standard deviation.

Table 6 reports the experimental results. It is can observed that MCVSVM and LMLP achieves statistically similar accuracies on the leukemia, lymphoma and prostate datasets. This further reveals the fact that certain discriminative information respectively resides in the null space, in which LMLP is employed, and the non-null space, in which MCVSVM is used. For the colon cancer dataset, MCVSVM gets the highest mean classification accuracy. However, on the whole, the proposed method TSSVM obviously outperforms MCVSVM and LMLP in that it incorporates the information both in the non-null space and in the null space.

6. Conclusion

In this paper, we propose a novel algorithm called TSSVM to deal with the classification task of the high-dimensional data space where the within-class scatter matrix is singular. Different from MCVSVM and LMLP, which use only the discriminant information in a single subspace, TSSVM takes fully into consideration the information both in the null space and in the non-null space. Experimental results indicate the effectiveness of TSSVM by comparing it with LMLP and MCVSVM. However, the experiments conducted in the paper focus on the image data. Actually, the high-dimensional data classification tasks commonly exist, for example in the field of bioinformatics. So, we still hope to study its more effective version to deal well with these problems and tasks and extend it to more fields in the future.

Acknowledgments

This work is supported in part by the National Science Foundation of China (Grant No. 61103168) and the Key Scientific Research Foundation of Sichuan Provincial Department of Education (Grant No. 11ZA004).

References

[1] B. Scholkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[2] F. Camastra, A. Verri, A novel kernel method for clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 801–805.
[3] X.M. Wang, F.L. Chung, S.T. Wang, Theoretical analysis for solution of support vector data description, Neural Netw. 24 (4) (2011) 360–369.
[4] Z. Harchaoui, F. Bach, O. Cappe, E. Moulines, Kernel-based methods for hypothesis testing: a unified view, IEEE Signal Process. Mag. 30 (4) (2013) 87–97.
[5] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, New York, 1995.
[6] J. Ruan, X. Wang, Y. Shi, Developing fast predictors for large-scale time series using fuzzy granular support vector machines, Appl. Soft Comput. 13 (9) (2013) 3981–4000.
[7] M. Sabzekar, M. Naghibzadeh, Fuzzy c-means improvement using relaxed constraints support vector machines, Appl. Soft Comput. 13 (2) (2013) 881–890.
[8] C.H. Wu, Y. Ken, T. Huang, Patent classification system using a new hybrid genetic algorithm support vector machine, Appl. Soft Comput. 10 (4) (2010) 1164–1177.
[9] D.H. Liu, H. Qian, G. Dai, Z.H. Zhang, An iterative SVM approach to feature selection and classification in high-dimensional datasets, Pattern Recognit. 46 (9) (2013) 2531–2537.
[10] Z. Wang, Y.H. Shao, T.R. Wu, A GA-based model selection for smooth twin parametric-margin support vector machine, Pattern Recognit. 46 (8) (2013) 2267–2277.
[11] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discov. 2 (2) (1998) 121–167.
[12] B. Scholkopf, A. Smola, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[13] B. Scholkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[14] S. Zafeiriou, A. Tefas, I. Pitas, Minimum class variance support vector machines, IEEE Trans. Image Process. 16 (10) (2007) 2551–2564.
[15] Q.X. Gao, J.J. Liu, H.J. Zhang, J. Hou, X.J. Yang, Enhanced fisher discriminant criterion for image recognition, Pattern Recognit. 45 (10) (2012) 3717–3724.
[16] A. Rozza, G. Lombardi, E. Casiraghi, P. Campadelli, Novel fisher discriminant classifiers, Pattern Recognit. 45 (10) (2012) 3725–3737.
[17] I. Kotsia, I. Pitas, S. Zafeiriou, Novel multiclass classifiers based on the minimization of the within-class variance, IEEE Trans. Neural Netw. 20 (1) (2009) 14–34.
[18] M. Wang, F.L. Chung, S.T. Wang, On minimum class locality preserving variance support vector machine, Pattern Recognit. 43 (8) (2010) 2753–2762.
[19] J. Yang, A.F. Frangi, J. Yang, D. Zhang, Z. Jin, KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2) (2005) 230–244.
[20] P. Howland, J. Wang, H. Park, Solving the small sample size problem in face recognition using generalized discriminant analysis, Pattern Recognit. 39 (2) (2006) 277–287.
[21] A.K. Qin, P.N. Suganthan, M. Loog, Generalized null space uncorrelated Fisher discriminant analysis for linear dimensionality reduction, Pattern Recognit. 39 (9) (2006) 1805–1808.
[22] D.L. Chu, G.S. Thye, A new and fast implementation for null space based linear discriminant analysis, Pattern Recognit. 43 (4) (2010) 1373–1379.
[23] A. Sharma, K.K. Paliwal, A new perspective to null linear discriminant analysis method and its fast implementation using random matrix multiplication with scatter matrices, Pattern Recognit. 45 (6) (2012) 2205–2213.
[24] X.X. Zhang, Y.D. Jia, A linear discriminant analysis framework based on random subspace for face recognition, Pattern Recognit. 40 (9) (2007) 2585–2591.
[25] F.X. Song, J.Y. Yang, S.H. Liu, Large margin linear projection and face recognition, Pattern Recognit. 37 (9) (2004) 1953–1955.
[26] J. Yang, J.Y. Yang, Why can LDA be performed in PCA transformed space? Pattern Recognit. 36 (2) (2003) 563–566.
[27] R. Fletcher, Practical Methods of Optimization, 2nd ed., Wiley, New York, 1987.
[28] K.I. Diamantaras, S.Y. Kung, Principal Component Neural Networks, Wiley, New York, 1996.
[29] C.W. Hsu, C.C. Chang, C.J. Lin, A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, University of National Taiwan, Taipei, 2003.
[30] A. Scholkopf, B. Smola, K.R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[31] X.F. He, S.C. Yan, Y.X. Hu, P. Niyogi, H.J. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[32] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, in: Proc. IEEE Intl Conf. Automatic Face and Gesture Recognition, May 2002.
[33] S. Nene, S.K. Nayar, H. Murase, Columbia object image library (coil-20). Technical report, 1996.
[34] L.G. Abril, C. Angulo, F. Velasco, J.A. Ortega, A note on the bias in SVMs for multiclassification, IEEE Trans. Neural Netw. 19 (4) (2008) 723–725.
[35] Yale University, Face Database, 2002 http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
[36] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Cell Biol. 96 (12) (1999) 6745–6750.
[37] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2) (2002) 203–209.
[38] M. Shipp, K. Ross, P. Tamayo, A. Weng, J. Kutok, et al., Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, Nat. Med. 8 (1) (2002) 68–74.