Pattern Recognition 43 (2010) 2753–2762

Contents lists available at ScienceDirect

Pattern Recognition

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

On minimum class locality preserving variance support vector machine

Xiaoming Wang [a], Fu-lai Chung [b], Shitong Wang [a,b,*]

[a] School of Information Technology, Jiangnan University, WuXi, JiangSu, China
[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Article history:
Received 7 July 2009
Received in revised form
14 January 2010
Accepted 13 February 2010

Keywords:
Supervised learning
Support vector machine
Minimum class variance support machine
Locality preserving projections

ABSTRACT

In this paper, a so-called minimum class locality preserving variance support machine (MCLPV_SVM) algorithm is presented by introducing the basic idea of the locality preserving projections (LPP), which can be seen as a modified class of support machine (SVM) and/or minimum class variance support machine (MCVSVM). MCLPV_SVM, in contrast to SVM and MCVSVM, takes the intrinsic manifold structure of the data space into full consideration and inherits the characteristics of SVM and MCVSVM. We discuss in the paper the linear case, the small sample size case and the nonlinear case of the MCLPV_SVM. Similar to MCVSVM, the MCLPV_SVM optimization problem in the small sample size case is solved by using dimensionality reduction through principal component analysis (PCA) and one in the nonlinear case is transformed into an equivalent linear MCLPV_SVM problem under kernel PCA (KPCA). Experimental results on real datasets indicate the effectiveness of the MCLPV_SVM by comparing it with SVM and MCVSVM.

1. Introduction

In the past decade, kernel methods [1] are widely studied and applied [2,3]. Support vector machine (SVM), as a kernel method, is a powerful machine learning method based on Vapnik's Statistical Learning Theory [4]. Different from other pattern recognition 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 [4–6]. The essential point of SVM is to find a linear separating hyperplane which achieves the maximal margin among two different classes of data in the linear case. Furthermore, SVM can be extended to build nonlinear separating decision hyperplane by exploiting kernelization techniques [1]. The optimization problem of SVM can be formulated as a quadratic programming problem which can be solved very efficiently by its dual optimization problem.

However, SVM solution does not take into consideration the class distribution and may result in a non-robust solution [7]. In order to overcome the drawback of SVM, a modified class of SVM called minimum class variance support vector machine (MCVSVM) is presented in [7] which is inspired from the optimization of Fisher's discriminant ratio [8,9]. When the training set contains fewer samples than the dimensionality of the training vectors, it has been proved that the solution of MCVSVM problems in such cases can be found through principal component analysis (PCA) dimensionality reduction [9,10]. In the nonlinear case, it has also been shown that, under kernel PCA (KPCA) [11], the nonlinear optimization problem can be transformed into an equivalent linear MCVSVM problem. Unlike SVM, the solution of MCVSVM takes into consideration both the samples in the boundaries and the distribution of the classes and gives a robust solution. However, the intrinsic manifold structure of the data space has not been taken into full consideration in MCVSVM.

Recently, a novel linear dimensionality reduction algorithm called locality preserving projections (LPP) [12–14] is proposed. LPP aims to preserve the local manifold structure of the samples space. To be specific, the manifold structure is modeled by a nearest-neighbor graph which preserves the local structure of the data space. The LPP method is the linear approximation to the eigenfunctions of the Laplace Beltrami operator on the samples manifold. In [15], the author presented the Laplacian support vector machine (LSVM) algorithm by combining SVM and LPP. However, LSVM aimed at the semi-supervised learning method.

Aiming at the drawback of SVM and MCVSVM that the intrinsic manifold structure of the data space is ignored, in this paper, we propose a novel learning algorithm called minimum class locality preserving variance support machine (MCLPV_SVM) in which the manifold structure within each class is explicitly considered. First, an adjacency graph in the same class is built, which can best reflect the geometry structure of the data manifold and the class

* Corresponding author at: School of Information Technology, Jiangnan University, WuXi, JiangSu, China.
E-mail address: wxwangst@yahoo.com.cn (S. Wang).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.02.013

relationship between the sample points. Then, by using the basic idea of the LPP, we define the locality preserving within-class scatter matrix. Finally, the optimization problem of MCLPV_SVM is formulated by using the locality preserving within-class scatter matrix. MCLPV_SVM are very closely related to MCVSVM and share some properties with the SVM and the MCVSVM. Both MCVSVM and MCLPV_SVM can also be seen as the large margin classifier but they employ the Mahalanobis distance metric rather than the Euclidean distance metric in SVM. However, a significant difference between MCVSVM and MCLPV_SVM is that the latter explicitly considers the data manifold structure. Moreover, the experimental results show that the intrinsic manifold structure is helpful to improve the classification performance. Besides, as is pointed out in [13], recently, a number of research efforts have shown that the face images possibly reside on a nonlinear submanifold. Therefore, in this case MCLP_SVM can usually achieve better performance in contrast to SVM and MCVSVM.

The rest of this paper is organized as follows. The related works will be reviewed in Section 2. In Section 3, the linear case of MCLPV_SVM is presented and the small sample size case is discussed where the number of the training vectors is smaller than the dimensions of samples. In Section 4, the nonlinear decision hyperplanes will be defined and solved. In Section 5, a discussion is carried out about the relationship of the proposed method with SVM and MCVSVM and constructing the weight matrix. The experimental results are reported in Section 6. Finally, conclusions are drawn in Section 7.

2. Related work

In this section, SVM, MCVSVM and LPP will be briefly introduced. For simplicity, only binary classification tasks are considered here. Multi-class cased can be solved by one-against-one [16]. Let a training dataset contain two classes of N samples, represented by $\{(\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_N, z_N)\}$ with training samples $\boldsymbol{x}_i = [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iM}]^T \in \boldsymbol{X}$ and labels $z_i \in \{1, -1\}$, where $i = 1, \ldots, N$, T denotes transpose and the dimensionality of the sample space is denoted by M (i.e. $\boldsymbol{X} \in \boldsymbol{R}^M$).

2.1. Minimum class variance support vector machine (MCVSVM)

Rather than simply minimizing the training error, support vector machine (SVM) [4] minimizes the structural risk which expresses an upper bound on generalization error [17]. However, actually, SVM is a local method in the sense that solution is exclusively determined by support vectors, whereas all other data points are irrelevant to the decision hyperplane [17]. Thus, a modified class of SVM called minimum class variance support machine (MCVSVM) which takes into consideration both the samples in the boundaries and the distribution of the classes and gives a robust solution is proposed in [7]. In the case where the training vectors are linearly separable MCVSVM optimization problem is defined as

$$\min_{\boldsymbol{w},b} \boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w} \tag{1}$$

$$\text{s.t.} \quad z_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, N \tag{2}$$

where the matrix $\boldsymbol{S}_W$ is the within-class scatter matrix defined as

$$\boldsymbol{S}_W = \sum_{K=1}^{2} \sum_{\boldsymbol{x} \in \boldsymbol{X}_K} (\boldsymbol{x} - \boldsymbol{u}_K)(\boldsymbol{x} - \boldsymbol{u}_K)^T \tag{3}$$

Here, $\boldsymbol{u}_K = (1/N_K)\sum_{\boldsymbol{x} \in \boldsymbol{X}_K} \boldsymbol{x}$ is the mean sample vector for the Kth class (K=1,2). MCVSVM can be seen as a compromise between SVM and FLDA [9]. Similar to SVM, the optimization problem (1) subject to the separability constraints (2) of MCVSVM can be efficiently solved by switching to its Wolfe dual problem using a Lagrangian formulation of the problem when the within-class scatter matrix $\boldsymbol{S}_W$ is nonsingular. However, MCVSVM encounter the same small size sample problem [18] as FLDA where the training set contains fewer samples than the dimensionality of the training vectors. It has been proven that the solution of the MCVSVM optimization problems in such cases can be found through PCA dimensionality reduction. It has also showed that the nonlinear MCVSVM problem is equivalent to a linear one, subject to an initial KPCA embedding of the training data.

Essentially, however, the optimization problem (1) subject to the separability constraints (2) of MCVSVM can be also written as

$$\max_{\boldsymbol{w},b} \rho \tag{4}$$

$$\text{s.t.} \quad \frac{z_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)}{\sqrt{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}} \geq \rho, \quad i = 1, \ldots, N \tag{5}$$

Note that the optimization problem (4) subject to the constraints (5) can be changed equivalently to (1) subject to the constraints (2). However, in (4) we can find easy that MCVSVM is also a large margin classifier but it employs the Mahalanobis distance metric [19] when calculating the distance from the separating hyperplane $\boldsymbol{w}^T \boldsymbol{x} + b = 0$ to the data point. Therefore, MCVSVM reflect the class distribution through introducing the matrix $\boldsymbol{S}_W$ in the Mahalanobis distance metric.

2.2. Locality preserving projections (LPP)

Locality preserving projections (LPP) is a linear dimensionality reduction algorithm by feature extraction or projection. It builds an adjacency graph incorporating neighborhood information of the data set. Using the Laplacian graph, LPP then computes a transformation matrix which maps the data points into a subspace. This linear transformation optimally preserves local neighborhood information in a certain sense. The representation map generated by this method may be viewed as a linear discrete approximation to a continuous map that naturally arises from the geometry of the manifold [12].

Let $N_k(\boldsymbol{x}_i)$ denotes k nearest neighbors of node i and G denote the adjacency graph of dataset $\boldsymbol{X}$ with N nodes. Here the ith node corresponds to the data point $\boldsymbol{x}_i$. Nodes i and j are connected by an edge if i is among k nearest neighbors of j or j is among k nearest neighbors of i, i.e. $\mathbf{x}_j \in N_k(\mathbf{x}_i)$ or $\mathbf{x}_i \in N_k(\mathbf{x}_j)$. In order to weigh the edges of the adjacency graph G, we generally need to calculate the weight matrix $\boldsymbol{W}$ in different ways. One common choice is weight of the Gaussian kernel as follows:

$$W_{ij} = \begin{cases} \exp\left(-\dfrac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right) & \text{if } \boldsymbol{x}_i \in N_k(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in N_k(\boldsymbol{x}_i) \\ 0 & \text{other} \end{cases} \tag{6}$$

where $||\boldsymbol{x}|| = (\sum_{i=1}^{M} \boldsymbol{x}_i^2)^{1/2}$ is the usual Euclidean (L_2) norm in $\boldsymbol{R}^M$, $t > 0$ is the Gaussian kernel parameter and can be empirically determined. Note, the weight matrix $\boldsymbol{W}$ of the graph G models the local structure of the data manifold. LPP finds the transformation vector $\boldsymbol{w}$=$\boldsymbol{R}^M$ by minimizing the following objective function:

$$\min_{\boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{w} \tag{7}$$

$$\text{s.t.} \quad \boldsymbol{w}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{w} = 1 \tag{8}$$

where $\boldsymbol{D}$ is a diagonal matrix and its entries are column (or row, since $\boldsymbol{S}$ is symmetric) sum of $\boldsymbol{W}$, i.e. $D_{ii} = \sum_j W_{ij}$, $\boldsymbol{L}$=$\boldsymbol{D} - \boldsymbol{W}$ is the Laplacian matrix. The transformation vector $\boldsymbol{w}$ that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem. Note that the two matrices $\boldsymbol{XLX}^T$ and $\boldsymbol{XDX}^T$ are both symmetric and positive semidefinite since the matrix $\boldsymbol{L}$ and the diagonal matrix $\boldsymbol{D}$ are both symmetric and positive semidefinite [13].

LPP seeks to preserve the intrinsic geometry of the data and local structure by minimizing the above objective function (7). LPP preserves well the intrinsic geometry of the data and local structure and has been successfully applied to face recognition. More detail can be found in [13,14].

3. Minimum class locality preserving variance support vector machine

In this section, MCLPV_SVM will be presented. Firstly, the locality preserving scatter matrix and the locality preserving within-class scatter matrix are defined. Secondly, the optimization problem formulation of the linear MCLPV_SVM is given by using the locality preserving within-class scatter matrix. Finally, we discuss MCLPV_SVM in the small sample size case.

3.1. Locality preserving within-class scatter matrix

Before continuing our discussion, here we would like to give the following definitions by using the basic idea of LPP.

Definition 2.1. Let $\boldsymbol{L}$ be the Laplacian matrix of the dataset $\boldsymbol{X}$, the matrix $\boldsymbol{Z}=\boldsymbol{XLX}^T=\boldsymbol{X}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{X}^T$ is called the locality preserving scatter matrix.

Definition 2.2. Suppose the dataset $\boldsymbol{X}$ be separated into two different classes, the matrix

$$\boldsymbol{Z}_W=\sum_{K=1}^{2}\boldsymbol{Z}_K \tag{9}$$

is called the locality preserving within-class scatter matrix. Here, $\boldsymbol{Z}_K$ (K=1,2) is the locality preserving scatter matrix of the Kth class $\boldsymbol{X}_K$, i.e. $\boldsymbol{Z}_K=\boldsymbol{X}_K(\boldsymbol{D}^K-\boldsymbol{W}^K)\boldsymbol{X}_K^T$, $\boldsymbol{D}^K$ is a diagonal matrix and $D_{ij}^K=\sum_j W_{ij}^K$, $\boldsymbol{W}^K$ is the weight matrix of the Kth class $\boldsymbol{X}_K$ which can be defined as

$$W_{ij}^K=\begin{cases}\exp\left(-\dfrac{\|\boldsymbol{x}_{Ki}-\boldsymbol{x}_{Kj}\|^2}{t}\right) & \text{if } \boldsymbol{x}_{Ki}\in N_k(\boldsymbol{x}_{Kj}) \text{ or } \boldsymbol{x}_{Kj}\in N_k(\boldsymbol{x}_{Ki})\\ 0 & \text{other}\end{cases} \tag{10}$$

where $\boldsymbol{x}_{Kj}$ refers to the ith sample point in the Kth class (K=1,2).

It is worthwhile to note that the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ is symmetric and positive semidefinite and formally similar to the within-class scatter matrix $\boldsymbol{S}_W$. However, the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ reflects the intrinsic geometry and local structure of the data. Furthermore, $\boldsymbol{Z}_W$ is different from the objective function of LPP when defining the weight matrix. In $\boldsymbol{Z}_W$ the weight matrix uses the class labels and carries not only the intrinsic manifold structure information of the data space but also the discriminating information, but in LPP the class labels are not used and the discriminating information is not carried.

3.2. Minimum class locality preserving variance support vector machine

In this subsection, we will present MCLPVSVM formulation. Assuming that the data is linearly separable, similar to the optimization problem (4) of MCVSVM, we define the optimization problem of MCLPV_SVM as follows:

$$\max_{\boldsymbol{w},b}\rho \tag{11}$$

$$\text{s.t.}\quad \frac{z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)}{\sqrt{\boldsymbol{w}^T\boldsymbol{Z}_W\boldsymbol{w}}}\ge\rho,\quad i=1,\ldots,N. \tag{12}$$

Compared with MCVSVM, MCLPV_SVM incorporate the intrinsic geometry and local structure of the data in the similar way. Here, the essential difference between SVM, MCVSVM and MCLPV_SVM can be clearly seen: when we define the distance for the sample point $\mathbf{x}_i$ to the decision hyperplane $\boldsymbol{w}^T\boldsymbol{x}+b=0$, SVM is $z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)/\sqrt{\boldsymbol{w}^T\boldsymbol{w}}$, MCVSVM is $z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)/\sqrt{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}$, and MCLPV_SVM is $z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)/\sqrt{\boldsymbol{w}^T\boldsymbol{Z}_W\boldsymbol{w}}$. However, SVM, MCVSVM and MCLPV_SVM are all the large margin classifier. SVM employ directly the Euclidean distance metric, but both MCVSVM and MCLPV_SVM employ the Mahalanobis distance metric. MCVSVM incorporate the distribution of the classes by using the within-class covariance matrix $\boldsymbol{S}_W$ in the distance metric, while MCLPV_SVM do the intrinsic geometry of the data and local structure by using the locality preserving within-class covariance matrix $\boldsymbol{Z}_W$. Note, the optimization problem (11) subject to the constraints (12) can be changed equivalently to

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\boldsymbol{w}^T\boldsymbol{Z}_W\boldsymbol{w} \tag{13}$$

$$\text{s.t.}\quad z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\ge 1,\quad i=1,\ldots,N \tag{14}$$

Fig. 1 describes the decision hyperplanes of SVM, MCVSVM, and MCLPV_SVM on an artificial dataset. As can be seen from the case illustrated in Fig. 1, the MCLPV_SVM decision hyperplane reflects the intrinsic manifold structure of the data and shows it is more reasonable. The MCSVM decision hyperplane reflects the average information of the class distribution and the SVM decision hyperplane does neither the intrinsic manifold structure of the data, nor the class distribution.

In the case where the training vectors are not linearly separable, the optimum decision hyperplane is found by using the soft margin [4] formulation and solving the following optimization problem:

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\boldsymbol{w}^T\boldsymbol{Z}_W\boldsymbol{w}+C\sum_{i=1}^{N}\xi_i \tag{15}$$

$$\text{s.t.}\quad z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\ge 1-\xi_i,\quad \xi_i\ge 0,\quad i=1,\ldots,N. \tag{16}$$

where ξ_i denotes the non-negative slack variables for data point $\boldsymbol{x}_i$, C is a given constant that defines the cost of the errors after the classification. C is also called regularization parameter. Larger values of C correspond to higher penalty assigned to errors. The linearly separable case can be achieved when choosing $C=\infty$. Obviously, (15) is a quadratic programming problem. As in SVM and MCVSVM, we can transform this problem into its corresponding dual problem as follows. The primal Lagrangian is

$$L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\beta},\xi)=\frac{1}{2}\boldsymbol{w}^T\boldsymbol{Z}_W\boldsymbol{w}+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\boldsymbol{\alpha}_i[z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)-1+\xi_i]-\sum_{i=1}^{N}\beta_i\xi_i \tag{17}$$

where the vectors $\boldsymbol{\alpha}=[\alpha_1,\ldots,\alpha_N]^T$ and $\boldsymbol{\beta}=[\beta_1,\ldots,\beta_N]^T$ ($\boldsymbol{\alpha},\boldsymbol{\beta}\in R^N$) are the Lagrangian multipliers for the constraints (16). By differentiating with respect to $\boldsymbol{w}$, b and $\boldsymbol{\beta}$ and using the Karush–Kuhn–Tucker (KKT) conditions [20], the following holds:

$$\frac{\partial L}{\partial\boldsymbol{w}}=\boldsymbol{Z}_W\boldsymbol{w}-\sum_{i=1}^{N}\alpha_i z_i\boldsymbol{x}_i=0$$

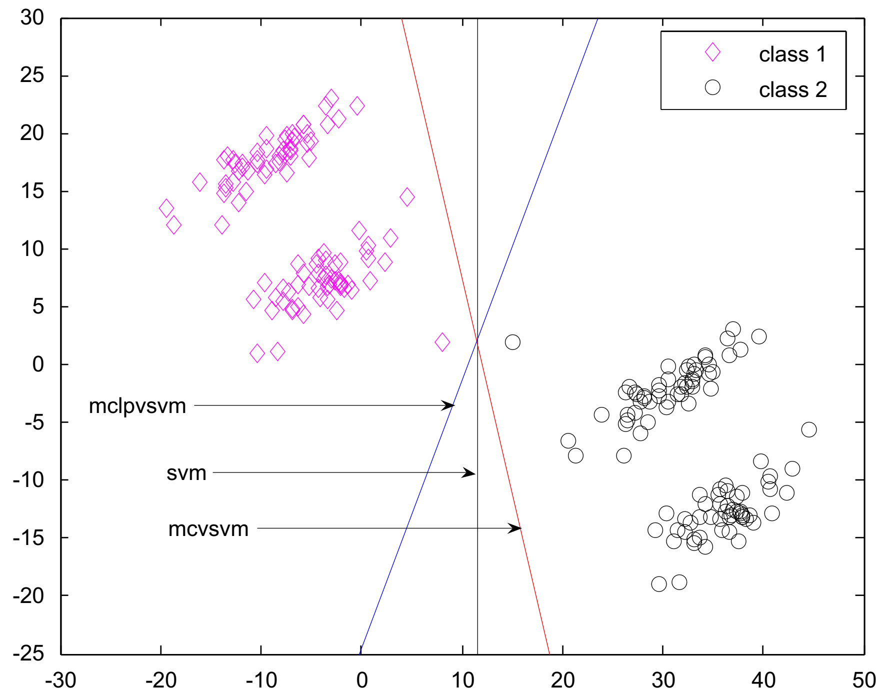

Fig. 1. Illustration of the decision hyperplanes generated by SVM, MCVSVM, and MCLPV_SVM.

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i z_i = \mathbf{0}$$

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = C\boldsymbol{e} - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0$$

$$\alpha_i (z_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0, \quad i = 1, \ldots, N \tag{18}$$

where $\boldsymbol{e}$ is a N-dimensional vector of ones, i.e., $\boldsymbol{e} = [1, \ldots, 1]^T$, $\mathbf{0} = [0, \ldots, 0]^T$. If the matrix $\boldsymbol{Z}_W$ is nonsingular or invertible, we have

$$\boldsymbol{w} = \boldsymbol{Z}_W^{-1} \sum_{i=1}^{N} \alpha_i z_i \boldsymbol{x}_i \tag{19}$$

By replacing (19) into (17) and using the KKT conditions, the constraint optimization problem (15) is reformulated to the Wolf dual problem

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{H} \boldsymbol{\alpha} + \boldsymbol{e}^T \boldsymbol{\alpha} \tag{20}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i z_i = 0, \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, N. \tag{21}$$

where $\boldsymbol{H}_{ij} = z_i z_j \boldsymbol{x}_i^T \boldsymbol{Z}_W^{-1} \boldsymbol{x}_j$. Suppose $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_N^*]^T$ can be used to solve the above optimization problem, then the optimal weight vector

$$\boldsymbol{w}^* = \boldsymbol{Z}_W^{-1} \sum_{i=1}^{N} z_i \alpha_i^* \boldsymbol{x}_i \tag{22}$$

If $0 < \alpha_i^* < C$, the corresponding data point $\boldsymbol{x}_i$ can be called a support vector. The optimal threshold b^* can be found by exploiting the fact that for all support vectors $\boldsymbol{x}_i$ their corresponding slack variables are zero, according to the KKT condition (18). However, averaging over all support vectors yields usually a numerically stable solution. We can calculate the optimal threshold b^* as follows:

$$b^* = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (z_i - z_i \sum_{j=1}^{N} \alpha_j^* \boldsymbol{H}_{ij}), \quad \boldsymbol{x}_i \in D_{SV} \tag{23}$$

where D_{SV} consist of all support vectors and N_{SV} is the number of the support vectors. So, the corresponding decision function of MCLPV_SVM will be

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} z_i \alpha_i^* (\boldsymbol{x}_i^T \boldsymbol{Z}^{-1} \boldsymbol{x}) + b^* \right) \tag{24}$$

3.3. MCLPV_SVM for small sample size problem

Similar to MCVSVM, the major drawback of MCLPV_SVM is that it may encounter the so-called small sample size problem [18]. The small sample size problem occurs whenever the number of samples is smaller than the dimensionality of the samples. In this case, the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ and the within-class scatter matrix $\boldsymbol{S}_W$ which are both $M \times M$ matrix will be singular. To deal with the problem, through dimensionality reduction using PCA, the optimization problem of MCVSVM is reformulated into an equivalent one in a lower dimensional space, where the within-class scatter matrix $\boldsymbol{S}_W$ is nonsingular and MCVSVM optimization problem can be efficiently solved. For MCLPV_SVM, we employ the same way, i.e. the data in the sample space is first transformed to a low-dimension space where the matrix $\boldsymbol{Z}_W$ is nonsingular through dimensionality reduction using PCA and then MCLPV_SVM is applied in the transformed space. In PCA, the total scatter matrix be defined as

$$\boldsymbol{S}_t = \sum_{\boldsymbol{x} \in \boldsymbol{X}} (\boldsymbol{x} - \boldsymbol{u})(\boldsymbol{x} - \boldsymbol{u})^T \tag{25}$$

where $\boldsymbol{u} = (1/N)\sum_{\boldsymbol{x} \in \boldsymbol{X}} \boldsymbol{x}$ is the total sample mean vector. Let $\boldsymbol{\Psi}$ and $\boldsymbol{\Pi}$ be the complementary dimensional spaces spanned by the orthonormal eigenvectors of $\boldsymbol{S}_t$ that correspond to nonzero eigenvalues and to zero eigenvalues, respectively. For MCLPV_SVM, similar to MCVSVM, we have the following theorem.

Theorem. *Let $\boldsymbol{w}=\boldsymbol{v}+\gamma$ where $\boldsymbol{w}\in\boldsymbol{R}^M$, $\boldsymbol{v}\in\boldsymbol{\Psi}$, $\gamma\in\boldsymbol{\Pi}$, thus the optimization problem (15) subject to the constraints (16) of MCLPV_SVM is equivalent to*

$$\min_{\boldsymbol{v},b,\xi}\frac{1}{2}\boldsymbol{v}^T\boldsymbol{Z}_W\boldsymbol{v}+C\sum_{i=1}^{N}\xi_i \tag{26}$$

$$\text{s.t.}\quad z_i(\boldsymbol{v}^T\boldsymbol{x}_i+b)\geq 1-\xi_i,\quad \xi_i\geq 0,\quad i=1,\ldots,N \tag{27}$$

A proof of the above Theorem can be found in Appendix.

According to the above theorem, we can conclude that the optimization problem of MCLPV_SVM can be derived from $\boldsymbol{\Psi}$ without any loss of the optimal discriminatory information. Suppose $\boldsymbol{S}_t$ has m nonzero eigenvectors and let the column vectors of $\boldsymbol{P}$ are eigenvectors corresponding to nonzero eigenvectors of $\boldsymbol{S}_t$, by linear algebra theory, $\boldsymbol{\Psi}$ is isomorphic to m-dimensional Euclidean space $\boldsymbol{R}^m$ [21,22]. And the corresponding isomorphic mapping is

$$\boldsymbol{v}=\boldsymbol{P}\boldsymbol{\eta},\quad \boldsymbol{v}\in\boldsymbol{\Psi},\quad \boldsymbol{\eta}\in\boldsymbol{R}^m \tag{28}$$

Thus, in $\boldsymbol{R}^m$ the optimization problem of MCLPV_SVM can be written as

$$\min_{\boldsymbol{\eta},b,\xi}\frac{1}{2}\boldsymbol{\eta}^T\tilde{\boldsymbol{Z}}_W\boldsymbol{\eta}+C\sum_{i=1}^{N}\xi_i \tag{29}$$

$$\text{s.t.}\quad z_i(\boldsymbol{\eta}^T y_i+b)\geq 1-\xi_i,\quad \xi_i\geq 0,\quad i=1,\ldots,N \tag{30}$$

where $\boldsymbol{\eta}\in\boldsymbol{R}^m$, $\boldsymbol{y}_i=\boldsymbol{P}^T\boldsymbol{x}_i$ and $\tilde{\boldsymbol{Z}}_W=\boldsymbol{P}^T\boldsymbol{Z}_W\boldsymbol{P}$. However, in $\boldsymbol{R}^m$ the locality preserving within-class scatter matrix $\tilde{\boldsymbol{Z}}_W$ may be still singular. In order to find the MCLPV_SVM hyperplane, we can transform the data into a lower dimension space through using PCA. So, the column vectors of $\boldsymbol{P}$ can be eigenvectors corresponding to the largest $\tilde{m}$ nonzero eigenvectors of $\boldsymbol{S}_t$. Here $\tilde{m}$ must be small enough to make $\tilde{\boldsymbol{Z}}_W=\boldsymbol{P}^T\boldsymbol{Z}_W\boldsymbol{P}$ nonsingular. Suppose $\{\boldsymbol{\eta}^*,b^*,\xi^*\}$ solves the above optimization problem, the decision function is

$$f(\boldsymbol{x})=\text{sgn}(\boldsymbol{\eta}^{*T}\boldsymbol{P}^T\boldsymbol{x}+b^*) \tag{31}$$

4. MCLPV_SVM in the nonlinear case

In the previous discussion, the derived decision function (or hyperplane) is derived in a linear form. In order to handle with nonlinear classification problems, we can seek to use the kernelization trick [1] to map the M-dimensional data points into a high-dimensional feature space, where a linear hyperplane corresponds to a nonlinear hyperplane in the original space. However, MCVSVM or MCLPV_SVM could not directly get the hyperplane because the within-class scatter matrix $\boldsymbol{S}_W^{\phi}$ or the locality preserving within-class scatter matrix $\boldsymbol{Z}_W^{\phi}$ is generally singular in the feature space. In order to overcome the difficulty, MCVSVM employ PCA to transform the data in the feature space into a low-dimension space where the matrix $\boldsymbol{S}_W^{\phi}$ is nonsingular , and then the linear MCVSVM algorithm is applied in the a low-dimension space. In [7] the author pointed out that this process is in nature to transform the data in the sample space using KPCA into a new space where the matrix $\boldsymbol{S}_W^{\phi}$ is nonsingular, and then the linear MCVSVM algorithm is applied in the space.

Similarly, for the nonlinear MCLPV_SVM, we can first transform the data in the original space into a new space where the matrix $\boldsymbol{Z}_W^{\phi}$ is nonsingular using KPCA, and the linear MCLPV_SVM is used in the transformed space. In the transformed space, the optimization problem is reformulated as

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\boldsymbol{w}^T\tilde{\boldsymbol{Z}}_W^{\phi}\boldsymbol{w}+C\sum_{i=1}^{N}\xi_i \tag{32}$$

$$\text{s.t.}\quad z_i(\boldsymbol{w}^T\boldsymbol{y}_i+b)\geq 1-\xi_i,\quad \xi_i\geq 0,\quad i=1,\ldots,N \tag{33}$$

where $\boldsymbol{y}_i$ is the corresponding projected sample point in the transformed space using KPCA for the sample point $\boldsymbol{x}_i$ in the original space, $\tilde{\boldsymbol{Z}}_W^{\phi}$ is the locality preserving within-class scatter matrix in the transformed space. Assume the optimum case of (32) is $\{\boldsymbol{w}^*,b^*,\boldsymbol{\xi}^*\}$, the decision function can be written as

$$f(\boldsymbol{x})=\text{sgn}(\boldsymbol{w}^{*T}\boldsymbol{y}+b^*) \tag{34}$$

where $\boldsymbol{y}$ is the projected vector in the transformed space using the KPCA transform for the data point $\boldsymbol{x}$ which is unlabeled.

5. Discussion

5.1. Connection to SVM

When the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ is nonsingular, let $\boldsymbol{P}=(\boldsymbol{Z}_W)^{-1/2}$, we have $\boldsymbol{P}^T=((\boldsymbol{Z}_W)^{-1/2})^T=(\boldsymbol{Z}_W)^{-1/2}=\boldsymbol{P}$ since $\boldsymbol{Z}_W$ is invertible and symmetric. Therefore, (15) subject to the constraints (16) can be written as

$$\min_{\boldsymbol{v},b,\xi}\boldsymbol{v}^T\boldsymbol{v}+C\sum_{i=1}^{N}\xi_i \tag{35}$$

$$\text{s.t.}\quad z_i(\boldsymbol{v}^T\boldsymbol{y}_i+b)\geq 1-\xi_i,\quad \xi_i\geq 0,\quad i=1,\ldots,N \tag{36}$$

where $\boldsymbol{y}_i=\boldsymbol{P}^T\boldsymbol{x}_i$, $\boldsymbol{v}=\boldsymbol{P}\boldsymbol{w}$. This is the primal optimization problem of SVM. It can be shown that the MCLPV_SVM formulation can be solved using the existing SVM software, thus making the solution easy to be computed. Assume the optimum case of (35) is $\{\boldsymbol{v}^*,b^*,\boldsymbol{\xi}^*\}$, thus, the decision function can written as

$$f(\boldsymbol{x})=\text{sgn}(\boldsymbol{v}^{*T}\boldsymbol{P}^T\boldsymbol{x}+b^*) \tag{37}$$

5.2. Connection to MCVSVM

In MCLPV_SVM, if the weight matrix $\boldsymbol{W}^K$ in the Kth (K=1,2) class $\boldsymbol{X}_K$ is defined as follows:

$$W_{ij}^K=\frac{1}{N_K} \tag{38}$$

according to [13], the locality preserving scatter matrix $\boldsymbol{Z}_K=\boldsymbol{X}_K\boldsymbol{L}_K(\boldsymbol{X}_K)^T$ is equivalent to the scatter matrix $\boldsymbol{S}_K$. So, we have

$$\boldsymbol{Z}_W=\sum_{K=1}^{2}\boldsymbol{Z}_K=\sum_{K=1}^{2}\boldsymbol{S}_K=\boldsymbol{S}_W \tag{39}$$

This suggests that in this case the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ is equivalent to the within-class scatter matrix $\boldsymbol{S}_W$. So, it can be easily concluded that MCVSVM can be obtained from the special affinity (or adjacency) matrix in MCLPV_SVM. MCLPV_SVM can be also seen as the generalized version of MCVSVM since its affinity (or adjacency) matrix is more general.

5.3. Constructing the weight matrix

Generally, we construct the weight matrix $\boldsymbol{W}_K$ in the Kth (K=1,2) class according to (10). However, the neighbor parameter k must firstly be determined when constructing the adjacency graph of the Kth (K=1,2) class. Furthermore, even though two data points in the same class are close to each other but not connected in the adjacency graph, the weight of their edge is 0. This means that they do not have similarity and are seen as data points which have different class labels. So, we suggest that here the weight matrix can be constructed as follows:

$$W_{ij}^K = \exp\left(-\frac{\|\boldsymbol{x}_i^K - \boldsymbol{x}_j^K\|^2}{\boldsymbol{t}}\right) \quad (40)$$

This means that two data points in the same class have always similarity. If the above way does not severely influence the classification accuracy, it will obviously bring two aspects of benefit: reduction in the parameter and time. This is in that we do not require to construct the adjacency graph.

6. Experiments

In this section, experimental results will be reported. In the first experiment we investigate the influence of parameters on MCLPV_SVM performance when using (10) to construct the weight matrix. In the second experiment we report the evaluation results of the linear MCLPV_SVM in comparison with SVM and MCVSVM on several datasets of the well-known University of California at Irvine (UCI) Repository of machine learning databases [23]. At last, we report the experimental results on a face dataset in the small sample size case and the nonlinear case. For the SVM algorithm, we employ LIBSVM [24].

At present, choosing the algorithms 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 [16], is that a set of the parameters is first given and then the best cross-validation mean rate among the set is used to estimate the generalized accuracy. In this work we adopted this strategy.

6.1. Parameter influence on performance of MCLPV_SVM

Compared with SVM and MCVSVM on the single parameter, however, MCLPV_SVM introduce two additional parameters—the heat parameter t and the neighborhood parameter k as shown in (10). In order to investigate the influence of the parameters on classification accuracy in MCLPV_SVM, we test MCLPV_SVM with different parameters on the heart dataset [23]. In this experiment, we use 40% data points of the heart dataset for training and the rest for testing. The data have been normalized (that is, mean zero and standard deviation one). For SVM, MCVSVM and MCLPV_SVM, the regularization parameter C is selected from the set {0.001, 0.01, 0.1, 1, 10, 100}. In addition, in MCLPV_SVM, the heat kernel parameter t and the neighborhood parameter k are, respectively, selected from the set {0.5, 1.0, 1.5, 2.0, 2.5} and {3, 6, 9, 12, 15, 30, 60, 90}.

The classification accuracy of SVM and MCVSVM on different regularization parameter C is reported in Table 1. Table 2 presents the classification accuracy of MCLPV_SVM on different regularization parameter C and different neighborhood parameter k when constructing the weigh matrix according to (10). Note, here the heat kernel parameter t vary in its set for each pair (C,k), and we give the best classification accuracy on different t. When constructing the weigh matrix according to (40) the classification accuracy on different regularization parameter C and different heat kernel t are reported in Table 3.

Table 1
Classification accuracy (%) on different C in SVM and MCVSVM.

	C=0.001	C=0.01	C=0.1	C=1	C=10	C=100
SVM	59.259	87.037	85.185	85.185	85.185	85.185
MCVSVM	55.556	55.556	55.556	87.037	85.185	85.185

Table 2
Classification accuracy (%) on different C and k in MCLPV_SVM.

	C=0.001	C=0.01	C=0.1	C=1	C=10	C=100
k=3	87.037	90.741	90.741	85.185	85.185	85.185
	t=0.5	t=1	t=2.5	t=0.5	t=1	t=1
k=6	87.037	90.741	90.741	85.185	87.037	85.185
	t=0.5	t=1	t=2.5	t=0.5	t=0.5	t=1
k=9	87.037	88.889	92.593	87.037	85.185	85.185
	t=0.5	t=1.5	t=2.5	t=0.5	t=1	t=1
k=12	87.037	88.889	92.593	87.037	85.185	85.185
	t=0.5	t=1.5	t=2.5	t=2.5	t=1	t=1
k=15	87.037	88.889	90.741	87.037	87.037	85.185
	t=0.5	t=1.5	t=2.5	t=2.5	t=0.5	t=1
k=30	87.037	88.889	90.741	87.037	85.185	85.185
	t=0.5	t=1.5	t=2.5	t=2.5	t=1	t=1
k=60	87.037	88.889	90.741	87.037	87.037	85.185
	t=0.5	t=1.5	t=2.5	t=2.5	t=0.5	t=1
k=90	87.037	88.889	90.741	87.037	85.185	85.185
	t=0.5	t=1.5	t=2.5	t=2.5	t=1	t=1

Table 3
Classification accuracy (%) on different C and t in MCLPV_SVM.

	C=0.001	C=0.01	C=0.1	C=1	C=10	C=100
t=0.5	87.037	87.037	87.037	81.481	87.037	85.185
t=1	77.778	88.889	87.037	85.185	85.185	85.185
t=1.5	55.556	88.889	88.889	85.185	85.185	85.185
t=2	55.556	55.556	90.741	85.185	85.185	85.185
t=2.5	55.556	55.556	88.889	87.037	85.185	85.185

From the experimental results, it can be found that the choice of C plays an important role in terms of the accuracy. Furthermore, if the best is selected as the experimental result, MCLPV_SVM shows better classification accuracy no matter what the weigh matrix is constructed according to (10) or (40). In addition, we can find that it is feasible to construct the weight matrix according to (40) and this way does not influence severely the performance in MCLPV_SVM. However, it reduces the parameters and the running time when employing (40) to construct the weight matrix since we do not need to construct the adjacency graph. Consequently, in the following experiments, we give the classification performance based on (40).

6.2. Performance comparison for linear cases

In this subsection, experimental results are presented for several benchmarking datasets from the well-known University of California at Irvine (UCI) Repository of machine learning databases [23]. A summary of the characteristics of the selected datasets are presented in Table 4. In these datasets, two-class cases and multi-class cases are both included. In multi-class cases, one-against-one strategy [16] is used. All data have been normalized before experiment.

The criteria used to estimate the generalized performance is the 5-fold cross validation accuracy [16] on the whole dataset, according to the size of the dataset, in order to ensure good statistical behavior. In 5-fold cross validation test, a dataset is divided into five subsets. Each time, one of the five subsets is used as the test set and all other four subsets are put together to form a training set. This procedure is repeated five times and then the average accuracy or error across all trials is computed. We give here the mean accuracy and standard deviation of the 5-fold cross validation. The regularization parameter C is selected from the set {0.001, 0.01, 0.1, 1, 10, 100}.

Table 5 reports the experimental results of SVM, MCVSVM and MCLPV_SVM on the selected datasets. Please note, here we give the best results on different parameters. From Table 5, on the whole, it can be found that there is an improvement in the generalized performance of MCLPV_SVM over SVM and MCVSVM. MCVSVM also has the comparable performance to SVM. These experimental results indicate that the generalized performance in the large margin classifier can be improved when the characteristics of data, especially the intrinsic manifold structure of the data space, are considered.

For a rigorous comparison of SVM, MCVSVM and MCVLP_SVM, we further performed the paired two-tailed t-tests [27] on these algorithms. The p-value of a t-test represents the probability that two sets of compared samples come from distributions with equal means. The smaller the p-value, the more significant the difference of the two average values is, and a p-value of 0.05 is a typical threshold which is considered statistically significant. Table 6 reports the experimental results of the t-tests. For example, the p-value of the t-test when comparing MCVSVM and SVM on the Newthyroid dataset is **0.039301** (<0.05), meaning that MCVSVM performs significantly better than SVM on this dataset at the 0.05 significant level. From Table 6, although MCLPV_SVM have on the whole better generalized performance, compared with MCVSVM, its improvement in the generalized performance is not significant. However, MCLPV_SVM significantly outperforms SVM on six of 10 datasets. Please note, for the Breast dataset, since SVM, MCVSVM, and MCLPV_SVM always keep the same accuracy in each classification test, so the t-test cannot be performed for these algorithms.

Table 4
Characteristics of the selected datasets.

Dataset	No. of patterns	No. of features	No. of classes
Heart	270	13	2
Breast	699	9	2
Pima	768	8	2
Wdbc	569	30	2
Ionosphere	351	34	2
Newthyroid	215	5	3
Wine	178	13	3
Iris	150	4	3
Vehicle	846	8	4
Glass	214	9	6

Table 5
Mean accuracy (%) and standard deviation of the cross validation on the selected datasets.

Dataset	SVM	MCVSVM	MCLPV_SVM
Heart	84.815 ± 0.0428 $C=0.01$	85.185 ± 0.0370 $C=10$	$\mathbf{85.556 \pm 0.0318}$ $C=1$, $t=2$
Breast	$\mathbf{96.855 \pm 0.0209}$ $C=1$	$\mathbf{96.855 \pm 0.0209}$ $C=100$	$\mathbf{96.855 \pm 0.0209}$ $C=100$, $t=0.1$
Pima	$\mathbf{77.473 \pm 0.0107}$ $C=0.01$	77.341 ± 0.0174 $C=10$	77.471 ± 0.0155 $C=10$, $t=2$
Wdbc	97.534 ± 0.00675 $C=0.1$	96.484 ± 0.0097 $C=100$	$\mathbf{97.699 \pm 0.0120}$ $C=10$, $t=3$
Ionosphere	87.167 ± 0.0316 $C=10$	84.889 ± 0.0459 $C=100$	$\mathbf{90.020 \pm 0.0434}$ $C=100$, $t=0.8$
Newthyroid	96.279 ± 0.0237 $C=10$	$\mathbf{98.140 \pm 0.0271}$ $C=100$	$\mathbf{98.140 \pm 0.0271}$ $C=100$, $t=0.8$
Wine	93.235 ± 0.0455 $C=0.1$	93.268 ± 0.0375 $C=100$	$\mathbf{95.523 \pm 0.0375}$ $C=100$, $t=5$
Iris	96.667 ± 0.0421 $C=0.1$	$\mathbf{98.667 \pm 0.0266}$ $C=1$	$\mathbf{98.667 \pm 0.0266}$ $C=1, t=5$
Vehicle	79.787 ± 0.0142 $C=10$	80.498 ± 0.0184 $C=1$	81.086 ± 0.0155 $C=100$, $t=6$
Glass	60.299 ± 0.0695 $C=10$	60.310 ± 0.0706 $C=10$	$\mathbf{63.577 \pm 0.0655}$ $C=100$, $t=2$

Table 6
P-value of t-test on the selected datasets.

Dataset	MCVSVM/SVM	MCLPV_SVM /SVM	MCLPV_SVM/ MCVSVM
Heart	0.62131	0.47666	0.3739
Breast	–	–	–
Pima	0.86447	0.99724	0.70453
Wdbc	0.18003	0.7163	0.18228
Ionosphere	0.12102	**0.04509**	**0.0010826**
Newthyroid	**0.039301**	**0.039301**	1
Wine	0.76549	**0.03172**	**0.046973**
Iris	**0.020484**	**0.020484**	1
Vehicle	0.53438	**0.019329**	0.67022
Glass	0.99695	**0.011586**	**0.029963**

6.3. *Performance comparison for small size sample case and nonlinear cases*

In the subsection, we report the experimental results in the small sample size case and the nonlinear case. In this study, the Yale face databases [25] were tested. The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 gray scale images of 15 individuals. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). In the experiments, preprocessing to locate the faces was applied. 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 1,024-dimensional vector in image space. More details can be found in [26]. The databases in Matlab format after being preprocessed is available at: http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html. Fig. 2 depicts some sample images after being preprocessed.

The typical kernel used in our experiments is the Gaussian kernel, i.e. $\exp(-(\boldsymbol{u}-\boldsymbol{v})^T(\boldsymbol{u}-\boldsymbol{v})/2\sigma^2)$, where σ is the spread of the Gaussian kernel. Here, we let $\sigma=10$. As we described previously, in this case both the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ and the within-class scatter matrix $\boldsymbol{S}_W$ are singular. So, PCA or KPCA is used to project the data in the original space into a subspace where the locality preserving within-class scatter matrix $\boldsymbol{Z}_W$ in MCLPV_SVM or the within-class scatter matrix $\boldsymbol{S}_W$ in MCVSVM is nonsingular. In order to make our experimental results fair, we took the same dimension reduction before all three algorithms run.

The experimental results in the small sample size case are reported in Table 7 and those in the nonlinear case do in Table 8. Note, here we give the experimental results after reducing to different dimensions using PCA or KPCA. Since the training samples are linearly independent, we can project the data onto the $N_{tr}-2$ dimension space. Here N_{tr} is the number of the training samples. The experimental results obtained by directly applying SVM are reported in the bottom row of the table. From Tables 7 and 8, it can be found that MCLPV_SVM outperforms SVM and MCVSVM on the whole. This suggests that the performance can indeed be improved when the intrinsic manifold structure of the data space is taken into full consideration. This characteristic is embodied in MCLPV_SVM.

Tables 9 and 10 report respectively the experimental results of t-tests in the linear and nonlinear cases. It can be found that in the linear case MCLPV_SVM performs significantly better in

Fig. 2. The sample cropped face image in the Yale face image dataset after being preprocessed.

Table 7
Mean accuracy (%) and standard deviation of the 5-fold cross validation in the linear case on the Yale face image dataset.

Dim	SVM	MCVSVM	MCLPV_SVM
3	60.889 ± 0.0585 C=1	60.000 ± 0.0421 C=100	$\mathbf{64.000 \pm 0.0388}$ C=100, t=10
6	72.889 ± 0.1226 C=0.1	72.444 ± 0.1075 C=1	$\mathbf{76.000 \pm 0.1103}$ C=1, t=10
9	74.222 ± 0.0997 C=0.1	73.778 ± 0.1273 C=1	$\mathbf{76.667 \pm 0.1192}$ C=1, t=15
12	77.556 ± 0.0989 C=0.1	76.889 ± 0.1149 C=1	$\mathbf{80.889 \pm 0.0935}$ C=1, t=25
15	76.889 ± 0.1026 C=0.1	76.222 ± 0.1105 C=1	$\mathbf{78.444 \pm 0.0939}$ C=0.1, t=50
$N_{tr}-2$	77.556 ± 0.0989 C=0.1	$\mathbf{79.333 \pm 0.1103}$ C=1	$\mathbf{79.333 \pm 0.1103}$ C=1, t=50
All	76.889 ± 0.1026 C=1	–	–

Table 8
Mean accuracy (%) and standard deviation of the 5-fold cross validation in the nonlinear case on the Yale face image dataset.

Dim	SVM	MCVSVM	MCLPV_SVM
3	62.000 ± 0.0884 C=100	62.444 ± 0.0656 C=100	$\mathbf{63.778 \pm 0.0413}$ C=100, t=5
6	70.667 ± 0.0646 C=100	$\mathbf{75.111 \pm 0.1296}$ C=1	74.444 ± 0.1169 C=10, t=20
9	76.222 ± 0.1119 C=1	75.778 ± 0.1048 C=100	$\mathbf{76.889 \pm 0.1259}$ C=1, t=5
12	75.556 ± 0.1307 C=10	77.556 ± 0.1211 C=100	$\mathbf{79.556 \pm 0.1170}$ C=10, t=5
15	$\mathbf{77.333 \pm 0.1388}$ C=10	76.000 ± 0.1254 C=100	77.111 ± 0.1125 C=10, t=5
$N_{tr}-2$	76.000 ± 0.1388 C=10	77.333 ± 0.1388 C=100	$\mathbf{78.667 \pm 0.1240}$ C=10, t=5
All	76.667 ± 0.1333 C=10	–	–

Table 9
P-value of *t*-test in the linear case on the Yale face image dataset.

Dim	MCVSVM/SVM	MCLPV_SVM/SVM	MCLPV_SVM/MCVSVM
3	0.55426	**0.03598**	**0.012685**
6	0.71743	**0.03166**	**0.029925**
9	0.82464	**0.04937**	**0.039657**
12	0.70405	**0.04628**	**0.01781**
15	0.50228	**0.04016**	**0.032046**
$N_{tr}-2$	0.077737	0.077737	–

comparison with SVM and MCVSVM, while in the nonlinear case it does not so. This suggests that although in the nonlinear case MCLPV_SVM has insignificant performance difference in contrast to SVM and MCVSVM, it performs on the whole better in the sense of the average accuracy.

Table 10
P-value of *t*-test in the linear case on the Yale face image dataset.

Dim	MCVSVM /SVM	MCLPV_SVM /SVM	MCLPV_SVM/MCVSVM
3	0.82464	0.54475	0.3739
6	0.33432	0.33102	0.62116
9	0.71759	0.62124	0.60047
12	0.20799	**0.02799**	**0.02080**
15	0.39201	0.22049	0.095771
$N_{tr}-2$	0.3739	**0.049301**	0.17781

7. Conclusion

In this paper, we propose a novel minimum class locality preserving variance support machines (MCLPV_SVM). Different from SVM and MCVSVM, MCLPV_SVM take the intrinsic manifold structure of the data space into full consideration and inherit the characteristics of SVM and MCVSVM. In small sample size cases and nonlinear cases, similar to MCVSVM, PCA or KPCA is used to transform the data in original space into a low dimension space where the optimization problem of linear MCLPV_SVM can be efficiently solved. Experimental results indicate the effectiveness of MCLPV_SVM by comparing it with SVM and MCVSVM. Although the proposed method here demonstrates our initial attempt to solve small sample size problems and nonlinear classification tasks from a new perspective, we still hope to study its more effective version to deal well with these problems and tasks.

Acknowledgments

This work is supported by the Hong Kong Polytechnic University Grants (Grant no.Z-08R and G-U296), National Science Foundation of China (Grant nos.60773206, 60704047 and 90820002), 2007 Cultivation Fund of the key Scientific and Technical Innovation Project of Ministry of Education of China, National Key Lab. Of CAD & CG at ZheJiang University, Key Lab. of Computer Information Technologies at JiangSu Province, 2008 Postgraduate student's Research Fund at JiangSu Province.

Appendix

Proof of Theorem in Section 3.3.

Proof. Since $\gamma \in \boldsymbol{\Pi}$, $\gamma^T \boldsymbol{S}_t \gamma = 0$, then under the projection γ, for all training vectors $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ with $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ then $\gamma^T \boldsymbol{x}_i = \gamma^T \boldsymbol{x}_j$. In other words, all the training vectors $\boldsymbol{x}_i$ fall in the same point under the projection γ. Thus, $\gamma^T \boldsymbol{x}_i = r$ is a constant, $\forall \boldsymbol{x}_i$. So, we have

$$\boldsymbol{w}^T \boldsymbol{x}_i = (\boldsymbol{v}+\gamma)^T \boldsymbol{x}_i = \boldsymbol{v}^T \boldsymbol{x}_i + \boldsymbol{r} \tag{41}$$

For the Kth (K=1,2) class $\boldsymbol{X}_K$, according to Definition 2.2, we have

$$\boldsymbol{Z}_K = \boldsymbol{X}_K \boldsymbol{L}_K \boldsymbol{X}_K^T = \boldsymbol{X}_K(\boldsymbol{D}_K - \boldsymbol{W}_K)\boldsymbol{X}_K^T \tag{42}$$

where $\boldsymbol{W}_K$ is the weight matrix and $\boldsymbol{L}_K$ is the Laplacian matrix in the Kth (K=1,2) class, $\boldsymbol{D}^K$ is a diagonal matrix and $D_{ij}^K = \sum_j W_{ij}^K$. Since $\gamma^T \boldsymbol{x}_i = r$, it follows that

$$\begin{aligned}
\boldsymbol{Z}_K\gamma &= \boldsymbol{X}_K(\boldsymbol{D}_K - \boldsymbol{S}_K)\boldsymbol{X}_K^T\gamma \\
&= \left(\sum_{i=1}^{N_K} D_{ii}^K \boldsymbol{x}_{Ki}\boldsymbol{x}_{Ki}^T - \sum_{i=1}^{N_K}\sum_{j=1}^{N_K} W_{ij}^K \boldsymbol{x}_{Ki}\boldsymbol{x}_{Kj}^T \right)\gamma \\
&= \sum_{i=1}^{N_K}\sum_{j=1}^{N_K} W_{ij}^K \boldsymbol{x}_{Ki}\boldsymbol{x}_{Ki}^T\gamma - \sum_{i=1}^{N_K}\sum_{j=1}^{N_K} W_{ij}^K \boldsymbol{x}_{Ki}\boldsymbol{x}_{Kj}^T\gamma \\
&= r\sum_{i=1}^{N_K}\sum_{j=1}^{N_K} W_{ij}^K \boldsymbol{x}_{Ki} - r\sum_{i=1}^{N_K}\sum_{j=1}^{N_K} W_{ij}^K \boldsymbol{x}_{Ki} = 0
\end{aligned} \tag{43}$$

where $\boldsymbol{x}_{Ki}$ is the ith sample point in the Kth class, N_K is the number of samples in the Kth class, $\mathbf{0}$ is a M-dimensional vector of zeros. Similarly, we have

$$\gamma^T \boldsymbol{Z}_K = \mathbf{0}^T \tag{44}$$

$$\gamma^T \boldsymbol{Z}_K \gamma = 0 \tag{45}$$

According to (44) and (45), we can conclude that

$$\boldsymbol{v}^T \boldsymbol{Z}_K \gamma = r\boldsymbol{v}^T \mathbf{0} = 0 \tag{46}$$

$$\gamma^T \boldsymbol{Z}_K \boldsymbol{v} = r\mathbf{0}^T \boldsymbol{v} = 0 \tag{47}$$

Thus, it follows that

$$\begin{aligned}
\boldsymbol{w}^T \boldsymbol{Z}_W \boldsymbol{w} &= (\boldsymbol{v}+\gamma)^T \left(\sum_{K=1}^{2} \boldsymbol{Z}_K \right)(\boldsymbol{v}+\gamma) \\
&= \sum_{K=1}^{2} (\boldsymbol{v}^T\boldsymbol{Z}_K\boldsymbol{v} + \boldsymbol{v}^T\boldsymbol{Z}_K\gamma + \gamma^T\boldsymbol{Z}_K\boldsymbol{v} + \gamma^T\boldsymbol{Z}_K\gamma) \\
&= \sum_{K=1}^{2} \boldsymbol{v}^T\boldsymbol{Z}_K\boldsymbol{v} = \boldsymbol{v}^T\boldsymbol{Z}_W\boldsymbol{v}
\end{aligned} \tag{48}$$

Now, using the KKT condition $\boldsymbol{\alpha}^T \boldsymbol{z} = \sum_{i=1}^{N} \alpha_i z_i = 0$, the following holds:

$$\sum_{i=1}^{N} \alpha_i z_i r = r\sum_{i=1}^{N} \alpha_i z_i = 0 \tag{49}$$

Using the previous facts the Lagrangian in (17) can be written as

$$\begin{aligned}
&L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\xi}) \\
&= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{Z}_W\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[z_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\beta_i\xi_i \\
&= \frac{1}{2}\boldsymbol{v}^T\boldsymbol{Z}_W\boldsymbol{v} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[z_i(\boldsymbol{v}^T\boldsymbol{x}_i + r + b) - 1 + \xi_i] - \sum_{i=1}^{N}\beta_i\xi_i \\
&= \frac{1}{2}\boldsymbol{v}^T\boldsymbol{Z}_W\boldsymbol{v} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[z_i(\boldsymbol{v}^T\boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\alpha_i z_i r - \sum_{i=1}^{N}\beta_i\xi_i \\
&= \frac{1}{2}\boldsymbol{v}^T\boldsymbol{Z}_W\boldsymbol{v} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[z_i(\boldsymbol{v}^T\boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\beta_i\xi_i
\end{aligned} \tag{50}$$

Note, this is the primal Lagrangian of (26). Then using the chain rule we can easily prove that

$$\left.\frac{\partial L}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}^*} = \left.\frac{\partial L}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{v}^*} = \mathbf{0} \Leftrightarrow \boldsymbol{Z}\boldsymbol{v}^* - \sum_{i=1}^{N}\alpha_i z_i \boldsymbol{x}_i = \mathbf{0} \tag{51}$$

Thus, the decision hyperplane depends only on $\boldsymbol{v} \in \boldsymbol{\Psi}$ (an arbitrary vector $\gamma \in \boldsymbol{\Pi}$ can be chosen). So, the theorem has been proven. □

References

[1] B. Scholkopf, A. Smola, Learning With Kernels, MIT Press, Cambridge, MA, 2002.
[2] I.W. Tsang, J.T. Kwok, P.M. Cheung, Core machines: fast svm training on very large data sets, Journal of Machine Learning Research 6 (2005) 363–392.
[3] Z.H. Deng, F.L. Chung, S.T. Wang, FRSDE: fast reduced set density estimator using minimal enclosing ball approximation, Pattern Recognition 41 (2008) 1363–1372.
[4] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[5] C.J.C. Burges, A tutorial on support machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.
[6] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.R. Muller, G. Ratsch, A.J. Smola, Input space vs. feature space in Kernel-based methods, IEEE Transactions on Neural Network 10 (5) (1999) 1000–1017.
[7] S. Zafeiriou, A. Tefas, I. Pitas, Minimum class variance support vector machines, IEEE Transactions on Image Processing 16 (10) (2007) 2551–2564.
[8] K. Fukunaga, Statistical Pattern Recognition, Academic, San Diego, CA, 1990.
[9] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed, Wiley, New York, 2001.
[10] K.I. Diamantaras, S.Y. Kung, Principal Component Neural Networks, Wiley, New York, 1996.
[11] A. Scholkopf, B. Smola, K.R. Muller, Nonlinear component analysis as a Kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.
[12] X. He, P. Niyogi, Locality preserving projections, in: Proceedings of the Conference on Advances in Neural Information Processing Systems, 2003, pp. 585–591.
[13] X.F. He, S.C. Yan, Y.X. Hu, P. Niyogi, H.J. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.
[14] E. Kokiopoulou, Y. Saad, Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2143–2156.
[15] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, The Journal of Machine Learning Research 7 (2006) 2399–2434.
[16] L.G. Abril, C. Angulo, F. Velasco, J.A. Ortega, A note on the bias in SVMs for multiclassification, IEEE Transactions on Neural Networks 19 (4) (2008) 723–725.
[17] T. Xiong, V. Cherkassky. A combined SVM and LDA approach for classification, in: Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31–August 4, 2005, pp. 1455–1459.
[18] P. Howland, J. Wang, H. Park, Solving the small sample size problem in face recognition using generalized discriminant analysis, Pattern Recognition 39 (2) (2006) 277–287.
[19] S.M. Xiang, F.P. Nie, C.S. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recognition 41 (2008) 3600–3612.
[20] R. Fletcher, Practical Methods of Optimization, second ed, Wiley, New York, 1987.
[21] J. Yang, J.Y. Yang, Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2) (2003) 563–566.
[22] 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 Transactions on Pattern Analysis and Machine Intelligence 27 (2) (2005) 230–244.
[23] C. Blake, C. Merz, UCI Repository of machine learning databases. Available: 〈http://www.ics.uci.edu/∼mlearn/MLRepository.html〉.
[24] C.C. Chang, C.J. Lin, LIBSVM: a library for support machines, 2001. Software available at 〈http://www.csie.ntu.edu.tw/∼cjlin/libsvm〉.
[25] Yale Univ, Face Database, 〈http://cvc.yale.edu/projects/yalefaces/yalefaces.html〉, 2002.
[26] D. Cai, X. He, J. Han, H.J. Zhang, Orthogonal Laplacianfaces for face recognition, IEEE Transactions on Image Processing 15 (11) (2006) 3608–3614.
[27] E. Alpaydin, Introduction to Machine Learning, The MIT Press, Cambridge, MA, USA, 2004.

About the author—XIAOMING WANG received B.S. and M.S degrees in the school of information from JiangNan University in 2001 and 2007, respectively. He is currently a Ph.D. candidate in the same school of the same university. His current research interests include artificial intelligence, pattern recognition, data mining and fuzzy system.

About the author—FU-LAI CHUNG received the B.Sc. degree from the University of Manitoba, Canada, in 1987, and the M.Phil. and Ph.D. degrees from the Chinese University of Hong Kong, in 1991 and 1995, respectively. He joined the Department of Computing, Hong Kong Polytechnic University, in 1994, where he is currently an associate professor. He has published widely in the areas of fuzzy systems, neural networks, and pattern recognition. His current research interests include fuzzy data mining, fuzzy neural network modeling, and fuzzy techniques for multimedia applications.

About the author—SHITONG WANG received the M.S. degree in computer science from Nanjing University of Aeronautics and Astronautics, China, in 1987. He visited London University and Bristol University in U.K., Hiroshima International University in Japan, Hong Kong University of Science and Technology, Hong Kong Polytechnic University, as a research scientist, for over 5 years. Currently, he is a full professor in The School of Information of Southern Yangtze University, China. His research interests include AI, neuron-fuzzy systems, pattern recognition, and image processing. He has published about 80 papers in international/national journals and has authored seven books. (Tel.: +86 510 8373140; e-mail: wxwangst@yahoo.com.cn).