Contents lists available at ScienceDirect

Neurocomputing

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

Discriminant analysis via support vectors

Suicheng Gu [a,b], Ying Tan [a,b,*], Xingui He [a,b]

[a] Key Laboratory of Machine Perception (MOE), Peking University, Beijing 100871, PR China
[b] Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, PR China

ARTICLE INFO

Available online 12 March 2010

Keywords:
Discriminant analysis via support vectors
Fisher's criteria
Linear discriminant analysis
Regularized discriminant analysis

ABSTRACT

In this paper, we show how support vector machine (SVM) can be employed as a powerful tool for *k*-nearest neighbor (kNN) classifier. A novel multi-class dimensionality reduction approach, discriminant analysis via support vectors (SVDA), is proposed. First, the SVM is employed to compute an optimal direction to discriminant each two classes. Then, the criteria of class separability is constructed. At last, the projection matrix is computed. The kernel mapping idea is used to derive the non-linear version, kernel discriminant via support vectors (SVKD). In SVDA, only support vectors are involved to compute the transformation matrix. Thus, the computational complexity can be greatly reduced for kernel based feature extraction. Experiments carried out on several standard databases show a clear improvement on LDA-based recognition.

1. Introduction

The *k*-nearest neighbors (kNN) [1] rule is one of the oldest and simplest methods for pattern classification. Feature extraction (dimensionality reduction) are often employed in helping kNN classifier to reduce computational complexity and improve classification accuracy.

The generic problem of linear dimensionality reduction is the following. Given a dataset $X=(x_1,x_2,\ldots,x_N)\in\mathcal{R}^{n\times N}$, find a transformation matrix $A=(a_1,\ldots,a_k)\in\mathcal{R}^{n\times k}$ that maps these N points to a set of points $Z=(z_1,z_2,\ldots,z_N)\in\mathcal{R}^{k\times N}$, such that z_i represents x_i, where $z_i=A^Tx_i$.

1.1. PCA and LDA

Principal component analysis (PCA) [2], also known as Karhunen–Loeve expansion, is a classical feature extraction and data representation technique widely used in the areas of pattern recognition and computer vision. Due to its simplicity and effectiveness, many variants of PCA were developed [3–5].

Linear discriminant analysis (LDA) [6], or called Fisher's linear discriminant (FLD), for feature extraction has been applied to a wide variety of problems such as face recognition. It often produces much better results than PCA. However, in practice, the LDA has three major problems: (1) It suffers from the small sample size (SSS) problem when dimensionality is greater than the sample size. (2) It creates subspaces that favor well separated classes over those that are not. (3) LDA assumes the data obey normal distribution. And it simply uses $\mu_a-\mu_c$ to discriminate two classes ω_a and ω_c. It fails to obtain the optimal direction to separate two classes.

Many algorithms tried to alleviate one or two of the problems in LDA. The regularized discriminant analysis (RDA) [7] added a multiple of identify matrix to the within-class matrix with regard to the small sample size problem. Another well-known approach is the Fisherface [8], in which LDA is employed after the PCA is used. Another technique, newLDA [9], first transforms the data into the null space of S_w. It then applies PCA to maximize the between-class scatter matrix in the transformed space.

1.2. Local learning

More recent years, many manifold (graph) based methods are implemented to preserve the local information and obtain a new subspace [10,11]. Some popular ones include: discriminant locally linear embedding (DLLE) [12], geometric mean for subspace selection (MGMD) [13], harmonic mean for subspace selection (MHMD) [14], discriminative locality alignment [15], transductive component analysis (TCA) [16], locality preserving projection (LPP) [17], marginal Fisher analysis (MFA) [18] and locality sensitive discriminant analysis (LSDA) [19], etc. To learn more about local learning methods, one can refer to [11].

1.3. Margin based discriminant

Large margin nearest neighbor (LMNN) [20] learns a Mahanalobis distance metric for kNN classification by semidefinite programming. Large margin component analysis (LMCA) [21]

[*] Corresponding author at: Key Laboratory of Machine Perception (MOE), Peking University, Beijing 100871, PR China.
E-mail address: ytan@pku.edu.cn (Y. Tan).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.09.021

solves for a low-dimensional embedding of the data such that Euclidean distance in this space minimizes the large margin metric objective described in [20]. Yuan and Pang [22] iteratively selects a series of simple but effective 1D subspaces, and then combines the corresponding 1D projections by Adaboost.

Support vector machine (SVM) [23] is based on the statistical learning theory of Vapnik and quadratic programming learning theory. The superior classification performance of SVM has been justified in numerous experiments, particularly in high dimensionality and small sample size (SSS) problems. Bi et al. [24] described a methodology for performing variable ranking and selection using support vector machines (SVMs). Margin maximizing discriminant analysis (MMDA) [25] attempted to preserve as much discriminant information as possible by projecting the dataset onto margin maximizing directions (separating hyperplane normals) found by an SVM algorithm. The corresponding normal vectors of the hyperplanes are taken as new features and the data are projected onto them. The first MMDA feature is obtained by simply using the standard SVM. Then, after obtaining orthogonal MMDA features, the second feature is found by optimizing the SVM in the remaining feature subspace. It is intrinsically a two-class approach.

In this paper, we developed a supervised dimensionality reduction approach for multiple-class problems, by employing SVM. To make a contrast with LDA, we call this approach discriminant analysis via support vectors (SVDA). Both linear and nonlinear models, discriminant analysis via support vectors (SVDA) and kernel discriminant via support vectors (SVKD), are described.

The rest of this paper is organized as follows. In Section 2, the LDA and SVM are reviewed briefly. In Section 3, the proposed SVDA algorithm is introduced. We describe how to perform SVDA in reproducing kernel Hilbert space (RKHS) which gives rise to kernel SVDA in Section 4. The experimental results are presented in Section 5. Finally, a conclusion is given in Section 6.

Notation conventions used in this paper:

i,N	counter and number of training samples;
n	dimension of training samples;
X	training samples with size of $n \times N$;
φ	$\mathcal{R}^n \rightarrow \mathcal{F}$;
$\mathcal{K}$	$\mathcal{K}(x_i,x_j)=\langle \varphi(x_i),\varphi(x_j)\rangle$;
K	kernel matrix, $K_{i,j}=\mathcal{K}(x_i,x_j)$;
a,M	counter and number of classes;
μ_a	mean vector of class ω_a;
N_a	number of samples in class ω_a;
I_a	collection of sample indexes in class ω_a;

2. LDA and SVM

2.1. LDA

In LDA, within-class and between-class scatter matrices are used to formulate the criteria of class separability. A within-class scatter matrix characterizes the scatter of samples around their respective class mean vectors, and it is expressed by

$$S_w=\sum_{a=1}^{M}\sum_{i\in I_a}(x_i-\mu_a)(x_i-\mu_a)^T. \tag{1}$$

A between-class scatter matrix characterizes the scatter of the class means around the mixture mean μ. It is expressed by

$$S_b=\sum_{a=1}^{M}N_a(\mu_a-\mu)(\mu_a-\mu)^T. \tag{2}$$

Linear discriminant analysis (LDA) seeks directions that are efficient for discrimination. Fisher criterion is used to find the projection matrix and the objective function of LDA is

$$a_{opt}=\arg\max_a \frac{a^T S_b a}{a^T S_w a}. \tag{3}$$

One can solve the generalized eigenvalue problem:

$$S_b a=\lambda S_w a. \tag{4}$$

2.1.1. RDA

In practice, the small sample size (SSS) problem is often encountered, where S_w is singular. Therefore, the maximization problem can be difficult to solve. To address this issue, the term εI is added, where ε is a small positive number and I is the identity matrix of proper size. This results in maximizing

$$a_{opt}=\arg\max_a \frac{a^T S_b a}{a^T(S_w+\varepsilon I)a}. \tag{5}$$

This is a special case of Friedman regularized discriminant analysis with regard to the small sample size problem [7].

2.2. SVM

Generally, an SVM [23] solves a binary (two-class) classification problem, and multi-class classification is accomplished by combining multiple binary SVMs. An M-class problem can be decomposed into M binary problems with each separating one class from the others, or into $M(M-1)/2$ binary problems with each discriminating between a pair of classes. On a pattern x, the discriminant function of a binary SVM is given by

$$f(x)=\sum_{i=1}^{l} y_i\alpha_i\mathcal{K}(x,x_i)+b, \tag{6}$$

where l is the number of learning patterns, y_i is the target value of learning pattern x_i (+1 for the first class and -1 for the second class), b is a bias, and $\mathcal{K}(x,x_i)$ is a kernel function which implicitly defines an expanded feature space:

$$\mathcal{K}(x,x_i)=\varphi(x)\cdot\varphi(x_i), \tag{7}$$

where $\varphi(x)$ is the feature vector in the expanded feature space and may have infinite dimensionality. Several popular kernels are: linear kernel $K(x_i, x_j)=x_i^T x_j$; polynomial kernel $K(x_i, x_j)=(1+x_T^i x_j)^p$ and RBF kernel $K(x_i,x_j)=\exp(-\|x_i-x_j\|^2/\sigma^2)$.

The discriminant function of Eq. (6) can be viewed as a generalized linear discriminant function with weight vector

$$w=\sum_{i=1}^{l} y_i\alpha_i\varphi(x_i). \tag{8}$$

The coefficients α_i ($i=1,2,\ldots,l$) are determined according to the learning patterns by solving the following optimization problem:

$$\text{Minimize } \tau(w)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\zeta_i$$

subject to $y_i f(x_i)\geq 1-\zeta_i$ and $\zeta_i\geq 0$, $i=1,2,\ldots,l$.

This is a quadratic programming problem and can be converted into the following dual problem:

$$\begin{aligned}&\text{Minimize} \quad Q(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\alpha_i\alpha_j y_i y_j\mathcal{K}(x_i,x_j)\\ &\text{subject to} \quad 0\leq\alpha_i\leq C, \quad i=1,2,\ldots,l,\\ &\qquad \text{and} \quad \sum_{i=1}^{l}\alpha_i y_i=0,\end{aligned} \tag{9}$$

where C (default $C=100$) is a parameter to control the tolerance of classification errors in learning.

2.3. SVM vs. LDA

From Fig. 1, we can see that the LDA fails to obtain the optimal direction to separate two classes, and approximated by

$$w_{ac}^{\mathrm{LDA}} = \mu_a - \mu_c. \tag{10}$$

Another problem with Fisher criterion is that in multi-class problems, it creates subspaces that favor well separated classes over those that are not. This is because the solution to Eq. (3) is a linear transform that maximizes the mean squared distance between the classes in the transformed space. As a result, an outlier (far away) class can be further separated from the remaining classes that really need a clear separation.

Alternately, the SVM can discover the optimal directions to maximize the margin between two classes. Also the optimal direction w_{ac}^{SVM} satisfies

$$\|w_{ac}^{\mathrm{SVM}}\| = \frac{2}{d_{ac}}, \tag{11}$$

where d_{ac} is the margin distance between ω_a and ω_c. Therefore, the SVM can create subspaces that favor closer classes over far away classes.

3. Discriminant analysis via support vectors (SVDA)

In SVDA, the within-class and between-class matrices are used to formulate the criteria of class separability, similarly as in LDA. But, the SVDA only employed the distinct support vectors (SVs) to compute between-class matrix V_b and within-class matrix V_w.

3.1. Between-class matrix

For each two classes, ω_a and ω_c, $1 \le a < c \le M$, a linear SVM is employed first. An optimal direction $w_{ac} = \sum_{i=1}^{l} y_i \alpha_i x_i$ can be found by solving Problem (9). Then, let an $n \times M(M-1)/2$ matrix W be the collection of $M(M-1)/2$ optimal w_{ac}, each column is one w_{ac}, $1 \le a < c \le M$.

The between-class matrix V_b is given by

$$V_b = \sum_{1 \le a < c \le M} w_{ac} w_{ac}^T = WW^T. \tag{12}$$

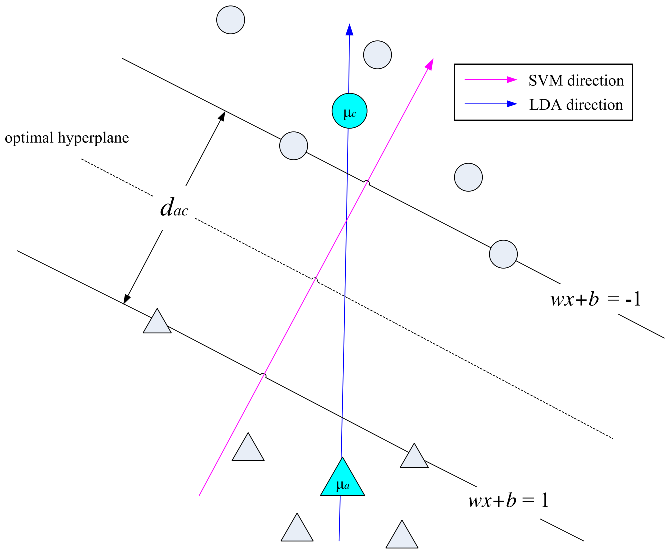


Fig. 1. SVM direction and LDA direction of binary (two-class) classification problem.

3.2. Within-class matrix

Let $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{\hat{N}}] \in \Re^{n \times \hat{N}}$ be the data matrix of SVs, where $\hat{N} \le N$ is the number of SVs. $\hat{N}_a$ denotes the number of SVs in class ω_a; $\hat{I}_a$ are the collection of indexes of SVs in class ω_a; $\hat{\mu}_a$ denotes mean of SVs in class ω_a.

Similar as in LDA, the within-class matrix V_w is given by

$$V_w = \sum_{a=1}^{M} \sum_{i \in \hat{I}_a} (\hat{x}_i - \hat{\mu}_a)(\hat{x}_i - \hat{\mu}_a)^T. \tag{13}$$

3.3. SVDA

While the between class matrix and the within class matrix are computed, the SVDA seeks to find the optimal projection by

$$a_{opt} = \arg\max_a \frac{a^T V_b a}{a^T V_w a}. \tag{14}$$

Note that, $\mathrm{rank}(V_w) \le \hat{N} - M$. If $\hat{N} - M < n$, then V_w will be singular. Therefore, we add a small multiple of identity matrix though

$$V_w^* = (1-\gamma)V_w + \gamma \cdot \frac{\mathrm{trace}(V_w)}{\hat{N} - M} \cdot I, \tag{15}$$

where $0 \le \gamma \le 1$ (default $\gamma = 0.05$) is a parameter of the regularizer.

One can solve the generalized eigenvalue problem:

$$V_b a = \lambda V_w^* a. \tag{16}$$

The eigenvectors corresponding to the k largest eigenvalues form the columns of the final transformation matrix A.

3.4. Optimal projection dimensionality

We have known that the optimal projection dimensionality of LDA, d^{LDA}, is with constraint $d^{LDA} \le M-1$, where M is the number of classes. We have $\mathrm{rank}(V_b) \le \min(N-1, M(M-1)/2)$, then the potential projection dimensionality of SVDA can be $\min(N-1, M(M-1)/2)$. However, the optimal projection dimensionality may be different. We will evaluate this in our experiments.

3.5. Computational complexity

The computational complexity of LDA is $O(n_3)$. The additional flops of SVDA for solving the quadratic programming (QP) problem is $O(N_a^3)$ and totally $O(M^2 N_{a*}^3)$, where $N_{a*} = \max_a(N_a)$. However, the original QP problem can be broken into a series of smaller QP problems, by using the well-known sequential minimal optimization (SMO) algorithm [26]. Especially for the high dimensional and small size problems, computational complexities of SVDA and LDA are nearly the same.

4. Kernel discriminant via support vectors (SVKD)

SVDA is a linear algorithm. It may fail to discover the intrinsic geometry when the data are highly nonlinear [27]. In this section, we will discuss how to perform SVDA in reproducing kernel Hilbert space (RKHS), which gives rise to kernel SVDA.

Let $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{\hat{N}}] \in \Re^{n \times \hat{N}}$ be the data matrix of SVs. And let $\hat{\Phi}$, with $\hat{N}$ column vectors, denote the data matrix of SVs in RKHS:

$$\hat{\Phi} = [\varphi(\hat{x}_1), \varphi(\hat{x}_2), \ldots, \varphi(\hat{x}_{\hat{N}})]. \tag{17}$$

First, the kernel SVM is employed to find the optimal direction w_{ac} to separate the two classes, ω_a and ω_c,

$$w_{ac} = \sum_{i=1}^{\hat{N}} y_i^{ac} \alpha_i^{ac} \varphi(\hat{x}_i) = \hat{\Phi} \alpha^{ac}, \tag{18}$$

where $\alpha^{ac} = (y_1^{ac}\alpha_1^{ac}, y_2^{ac}\alpha_2^{ac}, \ldots, y_{\hat{N}}^{ac}\alpha_{\hat{N}}^{ac})^T$ and

$$y_i^{ac} = \begin{cases} 1 & \text{if } i \in \hat{I}_a, \\ -1 & \text{if } i \in \hat{I}_c, \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Hence, V_b in Eq. (12) can be rewritten as

$$V_b = \sum_{a<c} \hat{\Phi}\alpha^{ac}(\hat{\Phi}\alpha^{ac})^T = \hat{\Phi}L_b\hat{\Phi}^T, \tag{20}$$

where $L_b = \sum_{a<c}\alpha^{ac}(\alpha^{ac})^T$.

For $i = 1,2,\ldots,\hat{N}$, if $i \in \hat{I}_a$, let

$$\theta_i(j) = \begin{cases} (\hat{N}_a - 1)/\hat{N}_a & \text{if } j = i, \\ -1/\hat{N}_a & \quad \text{if } j \in \hat{I}_a, \quad j \neq i, \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

Then $V_w = \sum_{i=1}^{\hat{N}} \hat{\Phi}\theta_i\theta_i^T\hat{\Phi}^T = \hat{\Phi}L_w\hat{\Phi}^T$, where $L_w = \sum_{i=1}^{\hat{N}} \theta_i\theta_i^T$.

Now, the eigenvector problem in RKHS can be written as follows:

$$\hat{\Phi}L_b\hat{\Phi}^T v = \lambda\hat{\Phi}L_w\hat{\Phi}^T v. \tag{22}$$

Because the eigenvectors of (22) are linear combinations of $\varphi(\hat{x}_1), \varphi(\hat{x}_2), \ldots, \varphi(\hat{x}_{\hat{N}})$, there exist coefficients β_i, $i = 1,2,\ldots,\hat{N}$, such that

$$v = \sum_{i=1}^{\hat{N}} \beta_i\varphi(\hat{x}_i) = \hat{\Phi}\beta, \tag{23}$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_{\hat{N}})^T$.

Following some algebraic formulations, we get

$$\hat{\Phi}L_b\hat{\Phi}^T v = \lambda\hat{\Phi}L_w\hat{\Phi}^T v,$$

$$\Rightarrow \hat{\Phi}L_b\hat{\Phi}^T\hat{\Phi}\beta = \lambda\hat{\Phi}L_w\hat{\Phi}^T\hat{\Phi}\beta,$$

$$\Rightarrow \quad \hat{\Phi}^T\hat{\Phi}L_b\hat{\Phi}^T\hat{\Phi}\beta = \lambda\hat{\Phi}^T\hat{\Phi}L_w\hat{\Phi}^T\hat{\Phi}\beta,$$

$$\Rightarrow \quad \hat{K}L_b\hat{K}\beta = \lambda\hat{K}L_w\hat{K}\beta, \tag{24}$$

where $\hat{K} \in \Re^{\hat{N}\times\hat{N}}$ is the kernel matrix, $\hat{K}(i,j) = \mathcal{K}(\hat{x}_i, \hat{x}_j)$.

Let $K_b = \hat{K}L_b\hat{K}$ and $K_w = \hat{K}L_w\hat{K}$, then the eigenvector problem can be rewritten as

$$K_b\beta = \lambda K_w^*\beta, \tag{25}$$

where $K_w^* = (1-\gamma)K_w + \gamma(\text{trace}(K_w)/(\hat{N}-M))I$, γ is a parameter of the regularizer. We recommend to use $\gamma = 0.05$ for default, and we use $\gamma = 0.05$ on both Isolet and USPS database in our experiments.

For a test point x, its projection to the obtained optimal direction is obtained as

$$F(x,\beta) = \frac{\sum_{i=1}^{N}\beta_i\mathcal{K}(x,\hat{x}_i)}{(\beta^T\hat{K}\beta)^{1/2}}. \tag{26}$$

Note that a uniform kernel must be used for SVM and the eigenvalue problem. It is also worthwhile to mention that the SVKD solves a smaller eigenvalue problem than kernel discriminant analysis (KDA) with less computational complexity of $O(\hat{N}^3)$. In some cases, we have $O(\hat{N}^3) \ll O(N^3)$.

5. Experiments and discussions

In this section, we investigate the use of SVDA on face recognition, speech recognition and handwritten recognition. RBF kernel is selected for the nonlinear SVDA and SVM.

5.1. Visualization on wine database

In the first experiment we sought to demonstrate the visualization capability of SVDA. We used the wine database from the UCI machine learning repository[1] which has 13 continuous attributes, three classes and 178 instances. The data were centered and scaled to have unit variance. We applied PCA, LDA, RDA and SVDA ($\gamma = 0.9$ for SVDA) to these datasets. Two dimensional projections of the data are shown in Fig. 2. The data are projected onto the eigenvectors corresponding to the two largest eigenvalues. Actually, the data are not linearly separable in the case of the PCA projection.

5.2. Face recognition on ORL and Yale databases

Two face databases were tested: ORL database, Yale database.[2] Table 1 lists some properties of the databases. The size of each cropped image in all the experiments is 32 × 32 pixels, with 256 gray levels per pixel. Thus, each image can be represented by a 1024-dimensional vector in image space. Each face image vector was normalized to unit before use.

Laplacian smoothing transform (LST) [28] is an efficient data independent pre-process approach for face recognition. It can discard high frequency features and make dimensionality reduction of an image. We reduced the input dimensionality (originally at 1024) by projecting the data onto its 90 (for ORL) and 80 (for Yale) low frequency coefficients, respectively.

5.2.1. Number of SVDA features

We employed ORL and Yale databases to evaluate how many SVDA features are proper for classification problems. From Fig. 3, we can get: (1) Number of RDA features cannot be greater than $M-1$, but SVDA can select more features. (2) If we choose less than 10 features for SVDA and RDA, SVDA obtains much higher recognition rates than RDA. (3) As number of features grows, the differences between SVDA and RDA get smaller. (4) For SVDA, best performance occurs when about $M-1$ SVDA features are selected.

5.2.2. Comparison

We compare our proposed algorithm with Fisherface (LDA, [6]), marginal Fisher analysis (MFA, [18]), discriminative locality alignment (DLA, [15]) and regularized discriminant analysis (RDA, [7]). Libsvm [29] is employed to train the SVDA and SVM. In all face recognition experiments, the dimensionality obtained by SVDA is always simplified as $k = M-1$, where M is the number of classes.

For each individual, G(=2, 3, 4, 5) images are randomly selected for training and the rest are used for testing. For each given G, we average the results over 50 random splits and report the mean. The regularizer γ is selected by using cross-validation. For each database, only one γ is used for all the experiments. We use $\gamma = 0.15$ for ORL database and $\gamma = 0.05$ for Yale database. The training set was used to learn a face subspace using the SVDA, RDA, MFA, DLA and Fisherface methods. Recognition was then performed in the subspaces. The result and dimensionality for each method on ORL and Yale databases are shown in Tables 2 and 3, respectively. As can be seen, our SVDA algorithm performed the best for all the cases, significantly.

From Tables 2 and 3, we have

- The manifold-based methods (MFA and DLA) achieved better performances than LDA. However, they did not show any advantage over RDA.
- NN classifier using SVDA features obtained higher recognition rates than SVM on the baseline features.

[1] Available at: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[2] Available at: http://www.cs.uiuc.edu/homes/dengcai2/Data/FaceData.html.

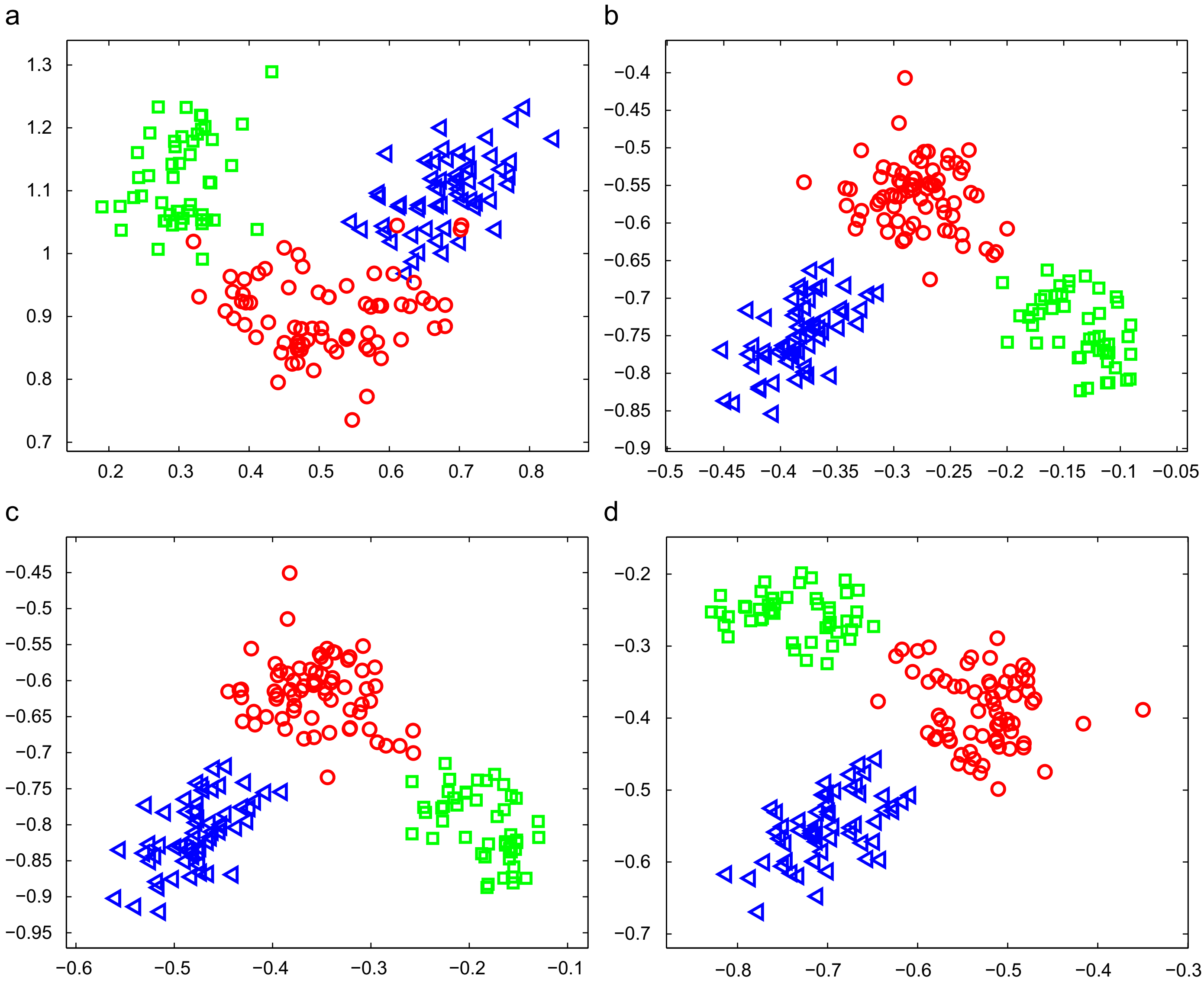


Fig. 2. Scatter plot of wine data projected onto a two-dimensional subspace. (a) PCA, (b) LDA (FLD), (c) RDA, (d) SVDA.

Table 1
Datasets for experiments.

Datasets	ORL	Yale	Isolet	USPS
Train set	400	165	6238	7291
Test set			1559	2007
Classes	40	15	26	10
Size	32 × 32	32 × 32	617	16 × 16
Dim(after LST)	90	80	–	–
Regularizer $\gamma =$	0.15	0.05	0.05	0.05

- SVDA not only improved performances of nearest neighbor classifier, but also improved performances of the SVM classifier itself.

5.3. Speech recognition on Isolet database

The Isolet dataset from UCI Machine Learning Repository has 6238 training samples, 1559 testing samples and 26 classes corresponding to letters of the alphabet. This database has been studied by many approaches. Nonlinear algorithms usually achieved better performance than linear algorithms. However, a kernel feature extraction method should solve the eigenproblem of a 6238 × 6238 matrix. Table 4 shows test error rates of different feature extraction methods and Table 5 shows performance of some other algorithms.

The SVKD is trained on the original data with 617 dimensionality without any preprocessing. Four thousand and twenty five SVs are remained after SVM is implemented. Then the SVKD only need to solve eigenvalues of a 4025×4025 matrix. We set $\sigma^2 = 100$ for the RBF kernel and the regularizer $\gamma = 0.05$. SVKD with kNN classifier obtains a test error rate of 3.2% and only 39 features are remained after SVKD.

5.4. Handwritten recognition on USPS database

The USPS database consisted of 16 × 16 pixel size-normalized images of handwritten digits, coming from US mail envelopes. The training and testing set had respectively 7291 and 2007 examples. The multi-class SVM with RBF kernel ($\sigma^2 = 125$) obtains a test error rate of 4.7% on this database.

The SVKD is trained on the original 256 dimensional data vectors without any preprocessing. Two thousand three hundred sixty six SVs are remained after SVM is implemented. Then the

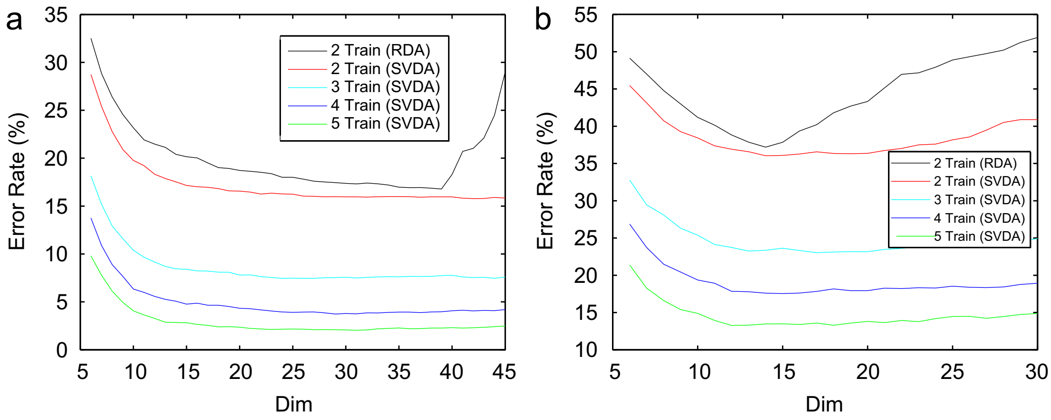


Fig. 3. Error rates of selecting different number of features. (a) ORL, (b) Yale.

Table 2
Error rates (%) of different algorithms on ORL database.

Feature	Classifier	2 Train	3 Train	4 Train	5 Train
Baseline	NN	33.2(90)	23.0(90)	18.3(90)	13.4(90)
Fisherface	NN	29.7(28)	16.6(39)	10.4(39)	6.8(39)
MFA	NN	20.7(39)	10.7(39)	8.7(39)	4.0(39)
DLA	NN	19.8(39)	9.9(39)	8.3(39)	3.7(39)
RDA	NN	17.9(39)	10.0(39)	5.8(39)	3.2(39)
SVDA	NN	**16.2**(39)	**7.5**(39)	**3.8**(39)	**2.1**(39)
Baseline	SVM-L	26.7(90)	15.6(90)	9.3(90)	6.3(90)
SVDA	SVM-L	16.1(39)	7.5(39)	4.1(39)	2.4(39)

NN: nearest neighbor classifier. SVM-L: SVM with linear kernel.

Table 3
Error rates (%) of different algorithms on Yale database.

Feature	Classifier	2 Train	3 Train	4 Train	5 Train
Baseline	NN	52.0(80)	45.8(80)	42.8(80)	39.5(80)
Fisherface	NN	52.0(14)	35.1(14)	27.1(14)	21.2(14)
MFA	NN	42.3(14)	29.3(14)	22.9(14)	20.1(14)
DLA	NN	40.9(14)	28.3(18)	20.3(24)	19.2(30)
RDA	NN	39.1(14)	25.6(14)	19.6(14)	15.4(14)
SVDA	NN	**36.0**(14)	**23.3**(14)	**17.8**(14)	**13.3**(14)
Baseline	SVM-L	45.7(80)	36.0(80)	31.1(80)	26.5(80)
SVDA	SVM-L	36.0(14)	23.1(14)	17.8(14)	13.5(14)

Table 4
Error rates (%) of different feature extraction methods on Isolet database.

Features	NN	kNN ($k=10$)
Baseline	11.4(617)	8.2(617)
PCA	11.2(80)	7.8(80)
LDA	6.9(25)	5.1(25)
SVDA	5.9(52)	4.8(52)
SVKD	**4.6(39)**	**3.2**(39)

SVKD only need to solve eigenvalues of a 2366×2366 matrix. It costs less than 4% time of traditional model.

We set $\sigma^2 = 125$ for the RBF kernel and the regularizer $\gamma = 0.05$. Table 6 shows test error rates of different feature extraction methods on USPS dataset. SVKD plus kNN classifier obtains a test error rate of 4.3% and only 38 features are remained after SVKD.

Table 5
Error rates (%) of different algorithms on Isolet database.

Feature	Dim	Classifier	Error rate
PCA	172	SVM-L	4.1
PCA	172	LMNN	3.7
PCA	172	SVM-R	3.3
SVKD	**39**	KNN	**3.2**

SVM-R: SVM with RBF kernel.

Table 6
Error rates (%) of different feature extraction methods on USPS database.

Features	Dim	Error rates (%)
Baseline	256	5.7
PCA	80	5.3
LDA	9	9.7
SVDA	44	5.5
SVKD	38	**4.3**

kNN ($k=6$) classifier is used.

6. Conclusions

This paper presents a novel multi-class dimension reduction approach, discriminant analysis via support vectors (SVDA), that potentially provide a solution to the small sample size problem, often associated with Fisher criterion. In particular, the paper has shown that: (1) (14) for dimension reduction is essentially a margin criterion; (2) the criterion has the potential to help alleviate Fisher bias toward outlier classes in multi-class problems; (3) the criterion can be easily and efficiently computed by using a regular SVM tool, such as Libsvm; (4) SVKD provides an efficient computation of kernel discriminant in large datasets. Therefore we believe that the proposed method will be a useful tool for researchers using machine learning.

This paper also introduced a universal regularizer (Eq. (15)) by fixing $\gamma = 0.05$. This non-parameter regularizer can achieve an optimal or nearly optimal performance in our experiments.

We have applied our algorithm to face recognition. Experiments on ORL, Yale, Isolet and USPS databases have been conducted to demonstrate the effectiveness of our algorithm.

Acknowledgements

This work was supported by National Natural Science Foundation of China (NSFC), under Grant no. 60875080 and

60673020, and also partially supported by the National High Technology Research and Development Program of China (863 Program), with Grant no. 2007AA01Z453.

References

[1] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions in Information Theory (1967) 21–26.

[2] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1991) 71–86.

[3] Y. Pang, Y. Yuan, X. Li, Iterative subspace analysis based on feature line distance, IEEE Transactions on Image Processing 18 (1) (2009) 903–907.

[4] Y. Pang, D. Tao, Y. Yuan, X. Li, Binary two-dimensional pca, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (4) (2008) 1176–1180.

[5] Y. Pang, Y. Yuan, X. Li, Gabor-based region covariance matrices for face recognition, IEEE Transactions on Circuits and Systems for Video Technology 18 (7) (2008) 989–993.

[6] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[7] J.H. Friedman, Regularized discriminant analysis, Journal of the American Statistical Association (1989) 165–175.

[8] P. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces versus Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 711–720.

[9] L.F. Chen, H.Y. Liao, M.T. Ko, J.C. Lin, G.J. Yu, A new lda-based face recognition system which can solve the small sample size problem, Pattern Recognition (2001) 1713–1726.

[10] T. Zhang, D. Tao, X. Li, J. Yang, A unifying framework for spectral analysis based dimensionality reduction, in: IJCNN, 2008, pp. 1670–1677.

[11] T. Zhang, D. Tao, X. Li, J. Yang, Patch alignment for dimensionality reduction, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1299–1313.

[12] X. Li, S. Lin, S. Yan, D. Xu, Discriminant locally linear embedding with high-order tensor data, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (2) (2008) 342–352.

[13] D. Tao, X. Li, X. Wu, S.J. Maybank, Geometric mean for subspace selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 260–274.

[14] W. Bian, D. Tao, Harmonic mean for subspace selection, in: ICPR, 2008, pp. 1–4.

[15] T. Zhang, D. Tao, J. Yang, Discriminative locality alignment, in: ECCV, 2008, pp. 725–738.

[16] W. Liu, D. Tao, J. Liu, Transductive component analysis, in: ICDM, 2008, pp. 433–442.

[17] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

[18] S. Yan, D. Xu, B. Zhang, Q.Y.H. Zhang, S. Lin, Graph embedding and extension: A general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (1) (2007) 40–51.

[19] D. Cai, X. He, K. Zhu, J. Han, H. Bao, Locality sensitive discriminant analysis, in: IJCAI, 2007.

[20] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: NIPS, 2006.

[21] L. Torresani, K. Lee, Large margin component analysis, in: NIPS, 2006.

[22] Y. Yuan, Y. Pang, Boosting simple projections for multi-class dimensionality reduction, in: IEEE International Conference on Systems, Man and Cybernetics, 2008, pp. 2231–2235.

[23] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, New York, 1995.

[24] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, The Journal of Machine Learning Research (2003) 1229–1243.

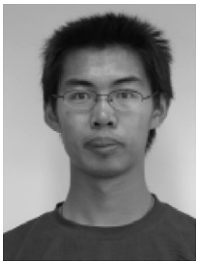
[25] A. Kocsor, K. Kovacs, C. Szepesvari, Margin maximizing discriminant analysis, in: ECML, 2004.

[26] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods-Support Vector Learning, 1999, pp. 185–208.

[27] Y. Pang, L. Wang, Y. Yuan, Generalized kpca by adaptive rules in feature space, International Journal of Computer Mathematics.

[28] S. Gu, Y. Tan, X. He, Laplacian smoothing transform for face recognition, Science in China Series F- Information Science, 2010.

[29] R.-E. Fan, P.-H. Chen, C.-J. Lin, Working set selection using second order information for training support vector machines, Journal of Machine Learning Research 6 (2005) 1889–1918.



Suicheng Gu received the BS degree from the Department of Applied Mathematics, Peking University, China, in 2004. He is currently a PhD student in the Key Laboratory of Machine Perception (MOE), Peking University, and Department of Machine Intelligence, EECS, Peking University. His research interests are in computer vision, machine learning, and statistical pattern recognition.

Ying Tan received his PhD degree in signal and information processing from Southeast University, Nanjing, China, in 1997. Since then, he became a postdoctoral research fellow and then an associate professor with University of Science and Technology of China. From 2000, he was a full professor, advisor of PhD candidates, and director of the Institute of Intelligent Information Science of his university. He worked with the Chinese University of Hong Kong in 1999 and in 2004–2005. He is also a recipient of the One Hundred Talent Program of the Chinese Academy of Science in 2005. Now, he is a full professor and PhD advisor of Key Laboratory of Machine Perception (MOE), Peking University, and Department of Machine Intelligence, EECS, Peking University, China. He has authored or coauthored more than 180 academic papers in refereed journals and conference proceedings. His current research interests include computational intelligence, machine learning algorithms, swarm intelligence, AIS, pattern recognition, intelligent information processing, etc. He is a senior member of the IEEE, ACM and CIE.

Xingui He has experience of years on theoretical research and engineering practices in the fields of computer software and artificial intelligence. Especially, he has made important contribution to the areas of fuzzy theory and technology, neural computing, databases and software engineering, has published more than 130 academic papers and 10 books in the areas. Now, he is an academician of Chinese Engineering Academy, a professor and PhD Advisor at Peking University in China.