Neighborhood MinMax Projections *

Feiping Nie[1], **Shiming Xiang**[2], **and Changshui Zhang**[2]
State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing 100080, China
[1]nfp03@mails.tsinghua.edu.cn; [2]{xsm, zcs}@mail.tsinghua.edu.cn

Abstract

A new algorithm, *Neighborhood MinMax Projections* (NMMP), is proposed for supervised dimensionality reduction in this paper. The algorithm aims at learning a linear transformation, and focuses only on the pairwise points where the two points are neighbors of each other. After the transformation, the considered pairwise points within the same class are as close as possible, while those between different classes are as far as possible. We formulate this problem as a constrained optimization problem, in which the global optimum can be effectively and efficiently obtained. Compared with the popular supervised method, *Linear Discriminant Analysis* (LDA), our method has three significant advantages. First, it is able to extract more discriminative features. Second, it can deal with the case where the class distributions are more complex than Gaussian. Third, the singularity problem existing in LDA does not occur naturally. The performance on several data sets demonstrates the effectiveness of the proposed method.

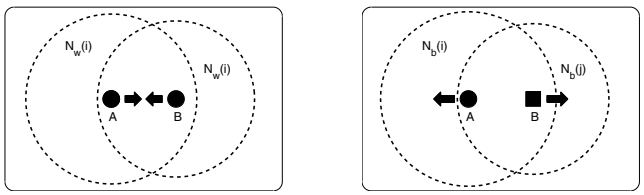
1 Introduction

Linear dimensionality reduction is an important method when facing with high-dimensional data. Many algorithms have been proposed during the past years. Among these algorithms, *Principal Component Analysis* (PCA) [Jolliffe, 2002] and *Linear Discriminant Analysis* (LDA) [Fukunaga, 1990] are two of the most widely used methods. PCA is an unsupervised method, which does not take the class information into account. LDA is one of the most popular supervised dimensionality reduction techniques for classification. However, there exist several drawbacks in it. One drawback is that it often suffers from the *Small Sample Size* problem when dealing with high dimensional data. In this case, the within-class scatter matrix $\mathbf{S}_w$ may become singular, which makes LDA difficult to be performed. Many approaches have been proposed to address this problem [Belhumeur *et al.*, 1997; Chen *et al.*, 2000; Yu and Yang, 2001]. However, these variants of LDA discard a subspace and thus some important discriminative information may be lost. Another drawback in LDA is its distribution assumption. LDA is optimal in the case that the data distribution of each class is Gaussian, which can not always be satisfied in real world applications. When the class distribution is more complex than Gaussian, LDA may fail to find the optimal discriminative directions. Moreover, the number of available projection directions in LDA is smaller than the class number [Duda. *et al.*, 2000], but it may be insufficient for many complex problems, especially when the number of class is small.

For the distance metric based classification methods, such as the nearest neighbor classifier, learning an appropriate distance metric plays a vital role. Recently, a number of methods have been proposed to learn a Mahalanobis distance metric [Xing *et al.*, 2003; Goldberger *et al.*, 2005; Weinberger *et al.*, 2006]. Linear dimensionality reduction can be viewed as a special case of learning a Mahalanobis distance metric(see section 5). This viewpoint can give a reasonable interpretation for the fact that the performance of nearest neighbor classifier can always be improved after performing linear dimensionality reduction.

In this paper, we propose a new supervised linear dimensionality reduction method, *Neighborhood MinMax Projections* (NMMP). The method is largely inspired by the classical supervised linear dimensionality reduction method, i.e., LDA, and the recent proposed distance metric learning method, *large margin nearest neighbor* (LMNN) classification [Weinberger *et al.*, 2006]. In our method, we focus only on the pairwise points where the two points are neighbors of each other. After the transformation, we try to pull the considered pairwise points within the same class as close as possible, and take those between different classes apart. This goal can be achieved by formulating the task as a constrained optimization problem, in which the global optimum can be effectively and efficiently obtained. Compared with LDA, our method avoids the three drawbacks in LDA discussed in above. Compared with the LMNN method, our method is computationally much more efficient. The performance on several data sets demonstrates the effectiveness of our method.

*This paper is Funded by Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).



(a) (b)

Figure 1: *In the left figure, point A and B belong to the same class i, and the two circles denote the within-class neighborhood of A and B respectively. A is B's within-class neighborhood and B is A's within-class neighborhood. After the transformation, we try to pull the two points as close as possible; In the right figure, point A belongs to class i, point B belongs to class j, and the two circles denote the between-class neighborhood of A and B respectively. A is B's between-class neighborhood and B is A's between-class neighborhood. After the transformation, we try to push the two points as far as possible.*

2 Problem Formulation

Given the data matrix $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n], \boldsymbol{x}_i \in \mathbb{R}^d$, our goal is to learn a linear transformation$\mathbf{W} : \mathbb{R}^d \rightarrow \mathbb{R}^m$, where $\mathbf{W} \in \mathbb{R}^{d \times m}$ and $\mathbf{W^T W} = \mathbf{I}$. $\mathbf{I}$ is $m \times m$ identity matrix. Then the original high-dimensional data $\boldsymbol{x}$ is transformed into a low-dimensional vector:

$$\boldsymbol{y} = \mathbf{W}^T \boldsymbol{x} \tag{1}$$

Let each data point of class i have two kinds of neighborhood: within-class neighborhood $\mathcal{N}_w(i)$ and between-class neighborhood $\mathcal{N}_b(i)$, where $\mathcal{N}_w(i)$ is the set of the data's $k_w(i)$ nearest neighbors in the same class i and $\mathcal{N}_b(i)$ is the set of the data's $k_b(i)$ nearest neighbors in the class other than i. Obviously, $1 \leq k_w(i) \leq n_i - 1$, and $1 \leq k_b(i) \leq n - n_i$, where n_i is the data number of class i.

Here, we focus only on the pairwise points where the two points are neighbors of each other. After the transformation, we hope that the distance of the considered pairwise points within the same class will be minimized, while the distance of those between different classes will be maximized. (see Figure 1).

After the transformation $\mathbf{W}$, the sum of the Euclidean distances of the pairwise points within the same class can be formulated as:

$$s_w = tr(\mathbf{W}^T \tilde{\mathbf{S}}_w \mathbf{W}) \tag{2}$$

where $tr(\cdot)$ denotes the trace operator of matrix, and

$$\tilde{\mathbf{S}}_w = \sum_{i,j:\boldsymbol{x}_i \in \mathcal{N}_w(C_j) \& \boldsymbol{x}_j \in \mathcal{N}_w(C_i)} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{3}$$

Here, C_i denote the class label of $\boldsymbol{x}_i$, and C_j denote the class label of $\boldsymbol{x}_j$. Obviously, $C_i = C_j$, and $\tilde{\mathbf{S}}_w$ is positive semi-definite.

Similarly, the sum of the Euclidean distances of the pairwise points between different classes is:

$$s_b = tr(\mathbf{W}^T \tilde{\mathbf{S}}_b \mathbf{W}) \tag{4}$$

where

$$\tilde{\mathbf{S}}_b = \sum_{i,j:\boldsymbol{x}_i \in \mathcal{N}_b(C_j) \& \boldsymbol{x}_j \in \mathcal{N}_b(C_i)} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{5}$$

Here, $C_i \neq C_j$, and $\tilde{\mathbf{S}}_b$ is positive semi-definite too.

To achieve our goal, we should maximize s_b while minimize s_w. The following two function can be used as objective:

$$\mathcal{M}_1(\mathbf{W}) = tr\left(\mathbf{W}^T(\tilde{\mathbf{S}}_b - \lambda \tilde{\mathbf{S}}_w)\mathbf{W}\right) \tag{6}$$

$$\mathcal{M}_2(\mathbf{W}) = \frac{tr(\mathbf{W}^T \tilde{\mathbf{S}}_b \mathbf{W})}{tr(\mathbf{W}^T \tilde{\mathbf{S}}_w \mathbf{W})} \tag{7}$$

As it is difficult to determine a suitable weight λ for the former objective function, we select the latter as our objective to optimization. In fact, as we will see, the latter is just a special case of the former, where the weight λ is automatically determined.

Therefore, we formulate the problem as a constrained optimization problem:

$$\mathbf{W}^* = \arg \max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T \tilde{\mathbf{S}}_b \mathbf{W})}{tr(\mathbf{W}^T \tilde{\mathbf{S}}_w \mathbf{W})} \tag{8}$$

Fortunately, the globally optimal solution of this problem can be efficiently calculated. In the next section, we will describe the details for solving this constrained optimization problem.

3 The Constrained Optimization Problem

We address the above optimization problem in a more general form which is described as follows:

The constrained optimization problem: Given the real symmetric matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ and the positive semi-definite matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$, $rank(\mathbf{B}) = r \leq d$. Find a matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ that maximize the following objective function with the constraint of $\mathbf{W}^T\mathbf{W} = \mathbf{I}$:

$$\mathbf{W}^* = \arg \max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T \mathbf{A} \mathbf{W})}{tr(\mathbf{W}^T \mathbf{B} \mathbf{W})} \tag{9}$$

At first, we propose *Lemma 1*, which shows that when $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ and $m > d - r$, the value of $tr(\mathbf{W}^T\mathbf{B}\mathbf{W})$ will not be equal to zero.

Lemma1. Suppose $\mathbf{W} \in \mathbb{R}^{d \times m}$, $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, $\mathbf{B} \in \mathbb{R}^{d \times d}$ is a positive semi-definite matrix, and $rank(\mathbf{B}) = r \leq d, m > d - r$, then it holds that $tr(\mathbf{W}^T\mathbf{B}\mathbf{W}) > 0$.

proof. According to the result of Rayleigh quotient [Golub and van Loan, 1996], $\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T\mathbf{B}\mathbf{W}) = \sum_{i=1}^{m} \beta_i$, where $\beta_1, \beta_2, \ldots, \beta_m$ are the first m smallest eigenvalues of $\mathbf{B}$. As $\mathbf{B}$ is positive semi-definite, $rank(\mathbf{B}) = r$, and $m > d - r$, then $\sum_{i=1}^{m} \beta_i > 0$. Therefore, with the constraint of $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, $tr(\mathbf{W}^T\mathbf{B}\mathbf{W}) \geq \min tr(\mathbf{W}^T\mathbf{B}\mathbf{W}) > 0$. □

Thus we discuss this optimization problem in two cases.

Case 1: $m > d - r$,

Lemma 1 ensures that the optimal value is finite in this case. Suppose the optimal value is λ^*, Guo [2003] has derived that $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T(\mathbf{A} - \lambda^*\mathbf{B})\mathbf{W}) = 0$.

Note that $tr(\mathbf{W}^T\mathbf{BW}) > 0$, so we can easy to see, $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T(\mathbf{A}-\lambda\mathbf{B})\mathbf{W}) < 0 \Rightarrow \lambda > \lambda^*$, and $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T(\mathbf{A}-\lambda\mathbf{B})\mathbf{W}) > 0 \Rightarrow \lambda < \lambda^*$.

On the other hand, $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T(\mathbf{A}-\lambda\mathbf{B})\mathbf{W}) = \gamma$, where γ is the sum of the first m largest eigenvalues of $\mathbf{A}-\lambda\mathbf{B}$. Given a value λ, if $\gamma = 0$,then λ is just the optimal value, otherwise $\gamma > 0$ implies λ is smaller than the optimal value and vice versa. Thus the global optimal value of the problem can be obtained by an iterative algorithm. Subsequently, in order to give a suitable value λ, we need to determine the possible bound of the optimal value. *Theorem 1* is proposed to solve this problem.

Theorem 1. Given the real symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and the positive semi-definite matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$, $rank(\mathbf{B}) = r \le d$. If $\mathbf{W}_1 \in \mathbb{R}^{d\times m_1}$, $\mathbf{W}_2 \in \mathbb{R}^{d\times m_2}$ and $m_1 > m_2 > d-r$, then $\max_{\mathbf{W}_1^T\mathbf{W}_1=\mathbf{I}} \frac{tr(\mathbf{W}_1^T\mathbf{AW}_1)}{tr(\mathbf{W}_1^T\mathbf{BW}_1)} \le \max_{\mathbf{W}_2^T\mathbf{W}_2=\mathbf{I}} \frac{tr(\mathbf{W}_2^T\mathbf{AW}_2)}{tr(\mathbf{W}_2^T\mathbf{BW}_2)}$.

The proof of *Theorem 1* is based on the following lemma:

Lemma 2. If $\forall i, a_i \ge 0, b_i > 0$ and $\frac{a_1}{b_1} \le \frac{a_2}{b_2} \le \cdots \le \frac{a_k}{b_k}$, then $\frac{a_1+a_2+\cdots+a_k}{b_1+b_2+\cdots+b_k} \le \frac{a_k}{b_k}$.

Proof. Let $\frac{a_k}{b_k} = q$. So $\forall i, a_i \ge 0, b_i > 0$, we have $a_i \le qb_i$. Therefore $\frac{a_1+a_2+\cdots+a_k}{b_1+b_2+\cdots+b_k} \le \frac{a_k}{b_k}$ □

Now we give the proof of *Theorem 1* in the following.

Proof of theorem 1. Suppose $\mathbf{W}_1{}^* = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_{m_1}]$ and $\mathbf{W}_1^* = \arg\max_{\mathbf{W}_1^T\mathbf{W}_1=\mathbf{I}} \frac{tr(\mathbf{W}_1^T\mathbf{AW}_1)}{tr(\mathbf{W}_1^T\mathbf{BW}_1)}$. Let $C_{m_1}^{m_2} = h$, without loss of generality, we suppose $\frac{tr(\mathbf{W}_{p(1)}^T\mathbf{AW}_{p(1)})}{tr(\mathbf{W}_{p(1)}^T\mathbf{BW}_{p(1)})} \le \frac{tr(\mathbf{W}_{p(2)}^T\mathbf{AW}_{p(2)})}{tr(\mathbf{W}_{p(2)}^T\mathbf{BW}_{p(2)})} \le \cdots \le \frac{tr(\mathbf{W}_{p(h)}^T\mathbf{AW}_{p(h)})}{tr(\mathbf{W}_{p(h)}^T\mathbf{BW}_{p(h)})}$

where $\mathbf{W}_{p(i)} \in \mathbb{R}^{d\times m_2}$ is the i-th combination of $\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_{m_1}$ with m_2 elements(note that $m_1 > m_2$), so the number of combinations is h.

Let $C_{m_1-1}^{m_2-1} = l$, note that each of $\boldsymbol{w}_j (1 \le j \le m_1)$ occurs l times in $\{\mathbf{W}_{p(1)}, \mathbf{W}_{p(2)}, ..., \mathbf{W}_{p(h)}\}$. According to *Lemma 2*, we have

$$\max_{\mathbf{W}_1^T\mathbf{W}_1=\mathbf{I}} \frac{tr(\mathbf{W}_1^T\mathbf{AW}_1)}{tr(\mathbf{W}_1^T\mathbf{BW}_1)} = \frac{l\cdot tr(\mathbf{W}_1{}^{*T}\mathbf{AW}_1{}^*)}{l\cdot tr(\mathbf{W}_1{}^{*T}\mathbf{BW}_1{}^*)} = \frac{tr(\mathbf{W}_{p(1)}^T\mathbf{AW}_{p(1)})+tr(\mathbf{W}_{p(2)}^T\mathbf{AW}_{p(2)})+\cdots+tr(\mathbf{W}_{p(h)}^T\mathbf{AW}_{p(h)})}{tr(\mathbf{W}_{p(1)}^T\mathbf{BW}_{p(1)})+tr(\mathbf{W}_{p(2)}^T\mathbf{BW}_{p(2)})+\cdots+tr(\mathbf{W}_{p(h)}^T\mathbf{BW}_{p(h)})}$$
$$\le \frac{tr(\mathbf{W}_{p(h)}^T\mathbf{AW}_{p(h)})}{tr(\mathbf{W}_{p(h)}^T\mathbf{BW}_{p(h)})} \le \max_{\mathbf{W}_2^T\mathbf{W}_2=\mathbf{I}} \frac{tr(\mathbf{W}_2^T\mathbf{AW}_2)}{tr(\mathbf{W}_2^T\mathbf{BW}_2)}$$ □

According to *Theorem 1* we know, with the reduced dimension m increases, the optimal value is decreased monotonously. When $m = d$, the optimal value is equal to $\frac{tr(\mathbf{A})}{tr(\mathbf{B})}$. So $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T\mathbf{AW})}{tr(\mathbf{W}^T\mathbf{BW})} \ge \frac{tr(\mathbf{A})}{tr(\mathbf{B})}$.

On the other hand, $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T\mathbf{AW}) = \sum_{i=1}^m \alpha_i$, and $\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T\mathbf{BW}) = \sum_{i=1}^m \beta_i$, where $\alpha_1, \alpha_2, \ldots, \alpha_m$ are the first m largest eigenvalues of $\mathbf{A}$, and $\beta_1, \beta_2, \ldots, \beta_m$ are the first m smallest eigenvalues of $\mathbf{B}$. Therefore, $\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T\mathbf{AW})}{tr(\mathbf{W}^T\mathbf{BW})} \le \frac{\alpha_1+\alpha_2+\cdots+\alpha_m}{\beta_1+\beta_2+\cdots+\beta_m}$.

As a result, the bound of the optimal value is given by $\frac{tr(\mathbf{A})}{tr(\mathbf{B})} \le \max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T\mathbf{AW})}{tr(\mathbf{W}^T\mathbf{BW})} \le \frac{\alpha_1+\alpha_2+\cdots+\alpha_m}{\beta_1+\beta_2+\cdots+\beta_m}$

Now, we obtain an iterative algorithm for obtaining the optimal solution, which is described in Table 1. From the algorithm we can see, only a few iterative steps are needed to obtain a precise solution. Note that the algorithm need not calculate the inverse of $\mathbf{B}$, and thus the singularity problem does not exist in it naturally.

Case 2: $m \le d - r$,

In this case, when $\mathbf{W}$ lies in the null space of matrix $\mathbf{B}$, then $tr(\mathbf{W}^T\mathbf{BW}) = 0$, the value of the objective function becomes infinite. Therefore, we can reasonably replace the optimization problem with $\mathbf{V}^* = \arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}} tr(\mathbf{V}^T(\mathbf{Z}^T\mathbf{AZ})\mathbf{V})$, where $\mathbf{V} \in \mathbb{R}^{(d-r)\times m}$, and $\mathbf{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{d-r}]$ are the eigenvectors corresponding to $d-r$ zero eigenvalues of $\mathbf{B}$.

We know that $\mathbf{V}^* = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_m]$, where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_m$ are the first m largest eigenvectors of $\mathbf{Z}^T\mathbf{AZ}$. So, in this case, the final solution is $\mathbf{W}^* = \mathbf{Z}\cdot\mathbf{V}^*$

Input:

The real symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and the positive semi-definite matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$, $rank(\mathbf{B}) = r \le d$. The error constant ϵ.

Output:

Projection matrix $\mathbf{W}^*$, where $\mathbf{W}^* \in \mathbb{R}^{d\times m}$ and $\mathbf{W}^{*T}\mathbf{W}^* = \mathbf{I}$.

In the case of : $m > d - r$.

1.$\lambda_1 \leftarrow \frac{tr(\mathbf{A})}{tr(\mathbf{B})}$, $\lambda_2 \leftarrow \frac{\alpha_1+\alpha_2+\cdots+\alpha_m}{\beta_1+\beta_2+\cdots+\beta_m}$, $\lambda \leftarrow \frac{\lambda_1+\lambda_2}{2}$, where $\alpha_1, \alpha_2, ..., \alpha_m$ are the first m largest eigenvalues of $\mathbf{A}$, $\beta_1, \beta_2, ..., \beta_m$ are the first m smallest eigenvalues of $\mathbf{B}$.

2.While $\lambda_2 - \lambda_1 > \epsilon$, do

a) Calculate γ, where γ is the sum of the first m largest eigenvalues of $\mathbf{A}-\lambda\mathbf{B}$.

b) If $\gamma > 0$, then $\lambda_1 \leftarrow \lambda$, else $\lambda_2 \leftarrow \lambda$.

c) $\lambda \leftarrow \frac{\lambda_1+\lambda_2}{2}$.

End while.

$\mathbf{W}^* = [\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, ..., \boldsymbol{\nu}_m]$, where $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, ..., \boldsymbol{\nu}_m$ are the first m largest eigenvectors of $\mathbf{A}-\lambda\mathbf{B}$.

In the case of : $m \le d - r$.

$\mathbf{W}^* = \mathbf{Z}\cdot[\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_m]$, where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_m$ are the first m largest eigenvectors of $\mathbf{Z}^T\mathbf{AZ}$, and $\mathbf{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{d-r}]$ are the eigenvectors corresponding to $d-r$ zero eigenvalues of $\mathbf{B}$.

Table 1: *The algorithm for the optimization problem*

4 Neighborhood MinMax Projections

The method of Neighborhood MinMax Projections(NMMP) is described in Table 2 . In order to speed up, PCA can be used as a preprocessing step before performing NMMP.

Denote the covariance matrix of data by $\mathbf{S}_t$, and denote the null space of $\mathbf{S}_t$ by ϕ, the orthogonal complement of ϕ by $\phi^\perp$.

0. Preprocessing: eliminate the null space of the covariance matrix of data, and obtain new data $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, where $rank(\mathbf{X}) = d$

1. Input:

$\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, $k_w(i)$, $k_b(i)$, m

2. calculate $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$ according to Eq.(3) and Eq.(5)

3. calculate $\mathbf{W}$ using the algorithm described in Table 1

4. Output:

$\boldsymbol{y} = \mathbf{W}^T\boldsymbol{x}$, where $\mathbf{W} \in \mathbb{R}^{d\times m}$ and $\mathbf{W}^T\mathbf{W} = \mathbf{I}$.

Table 2: *Algorithm of NMMP*

It is well known that the null space of $\mathbf{S}_t$ can be eliminated without lose of any information. In fact, it can be easy to prove that the null space of $\mathbf{S}_t$ comprises the null space of $\tilde{\mathbf{S}}_\mathbf{w}$ and the null space of $\tilde{\mathbf{S}}_\mathbf{b}$ defined in Section 2. Suppose $\boldsymbol{w} \in \phi^\perp$ and $\boldsymbol{\xi} \in \phi$, then

$$\frac{(\boldsymbol{w}+\boldsymbol{\xi})^T\tilde{\mathbf{S}}_b(\boldsymbol{w}+\boldsymbol{\xi})}{(\boldsymbol{w}+\boldsymbol{\xi})^T\tilde{\mathbf{S}}_w(\boldsymbol{w}+\boldsymbol{\xi})} = \frac{\boldsymbol{w}^T\tilde{\mathbf{S}}_b\boldsymbol{w}}{\boldsymbol{w}^T\tilde{\mathbf{S}}_w\boldsymbol{w}} \tag{10}$$

Eq.(10) demonstrates that eliminating the null space of the covariance matrix of data will not affect the result of the proposed method. Thus we use PCA to eliminate the null space of the covariance matrix of data.

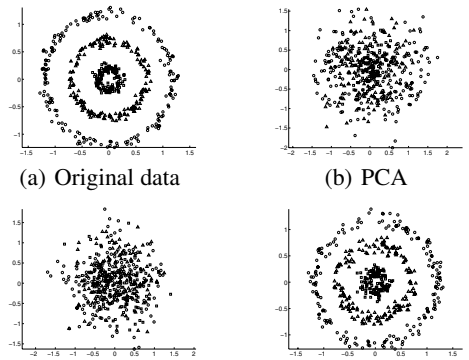
5 Discussion

Our method is closely connected with LDA. Both of them are supervised dimensionality reduction methods, and the goals are also similar. They both try to maximize the scatter between different classes, and minimize the scatter within the same class. The matrix $\tilde{\mathbf{S}}_b$ defined in Eq.(5) and $\tilde{\mathbf{S}}_w$ defined in Eq.(3) are parallel to the between-class scatter matrix $\mathbf{S}_b$ and within-class scatter matrix $\mathbf{S}_w$ in LDA respectively. In fact, when the number of neighbors reaches the number of the total available neighbors($k_w(i) = n_i - 1$, and $k_b(i) = n - n_i$, where n_i is the data number of class i, and n is the number of total data), we have $\tilde{\mathbf{S}}_b + \tilde{\mathbf{S}}_w = n^2\mathbf{S}_t$, which is similar to $\mathbf{S}_b + \mathbf{S}_w = \mathbf{S}_t$ in LDA.

However, in comparison with LDA, we do not impose the faraway pairwise points within the same class to be close to each other, which makes us focus more on the improvement of the discriminability of local structure. This property is especially useful when the distribution of class data is more complex than Gaussian. We give a toy example to illustrate it(Figure 2).

The toy data set consists of three classes(shown by different shapes). In the first two dimensions, the classes are distributed in concentric circles, while the other eight dimensions are all Gaussian noise with large variance. Figure 2 shows the two-dimensional subspace learned by PCA, LDA and NMMP, respectively. It illustrates that NMMP can find a low-dimensional transformation preserving manifold structure with more discriminability.

Moreover, compared with LDA, our method is able to extract more discriminative features and the singularity problem existing in LDA will not occur naturally.



(a) Original data (b) PCA (c) LDA (d) NMMP

Figure 2: *(a) is the first two dimensions of the original ten-dimensional data set; (b),(c),(d) are the two-dimensional subspace found by PCA, LDA and NMMP, respectively. It illustrates that NMMP can find a low-dimensional transformation preserving manifold structure with more discriminability.*

Distance metric learning is an important problem for the distance based classification method. Learning a Mahalanobis distance metric is to learn a positive semidefinite matrix $\mathbf{M}$, and using the Mahalanobis distance metric $(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\mathbf{M}(\boldsymbol{x}_i - \boldsymbol{x}_j)$ to replace the Euclidean distance metric $(\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)$, where $\mathbf{M} \in \mathbb{R}^{d\times d}, \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$. Note that $\mathbf{M}$ is positive semidefinite, with the eigen-decomposition, $\mathbf{M} = \mathbf{V}\mathbf{V}^T$, where $\mathbf{V} = [\sigma_1\boldsymbol{v}_1, \sigma_2\boldsymbol{v}_2, ..., \sigma_d\boldsymbol{v}_d]$, σ and $\boldsymbol{v}$ are eigenvalues and eigenvectors of $\mathbf{M}$. Therefore, the Mahalanobis distance metric can be formulated as $(\mathbf{V}^T\boldsymbol{x}_i - \mathbf{V}^T\boldsymbol{x}_j)^T(\mathbf{V}^T\boldsymbol{x}_i - \mathbf{V}^T\boldsymbol{x}_j)$. In this form, we can see that Learning a Mahalanobis distance metric is to learn a weighted orthogonal linear transformation. NMMP learns a linear transformation $\mathbf{W}$ with the constraint of $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. So it can be viewed as a special case of learning a Mahalanobis distance metric, where the weight value σ_i is either 0 or 1. Note that directly learning the matrix $\mathbf{M}$ is a very difficult problem and it is usually formulated as a semidefinite programming (SDP) problem, where the computation burden is extremely heavy. However, if we learn the transformation $\mathbf{V}$ instead of learning the matrix $\mathbf{M}$, the problem will become much easier to solve.

6 Experimental Results

We evaluated the proposed NMMP algorithm on several data sets, and compared it with LDA and LMNN method. The data sets we used belong to different fields, a brief description of these data sets is list on Table 3.

We use PCA as the preprocessing step to eliminate the null space of data covariance matrix $\mathbf{S}_t$. For LDA, due to the singularity problem existing in it, we further reduce the dimension of data such that the within-class scatter matrix $\mathbf{S}_w$ is nonsingular.

In each experiment, we randomly select several samples per class for training and the remaining samples for testing. the average results and standard deviations are reported over 50 random splits. The classification is based on k-nearest neighbor classifier($k = 3$ in these experiments).

	Iris	Bal	Faces	Objects	USPS	News
class	3	3	40	20	4	4
training number	60	60	200	120	80	120
testing number	90	565	200	1320	3794	3850
input dimensionality	4	4	10304	256	256	8014
dimensionality after PCA	4	4	199	119	79	119

Table 3: *A brief description of the data sets.*

data set	method	Projection number	Accuracy(%)	Std. Dev.(%)	Training time(per run)
Iris	baseline	4	95.4	1.8	–
	LDA	2	**96.6**	1.6	0s
	LMNN	4	96.2	1.5	4.91s
	NMMP	3	96.5	1.6	0.02s
Bal	baseline	4	61.6	2.8	–
	LDA	2	**74.9**	3.2	0s
	LMNN	4	70.1	3.7	3.37s
	NMMP	2	72.9	4.2	0.02s
Faces	baseline	199	86.9	2.1	–
	LDA	39	92.2	1.8	0.15s
	LMNN	199	95.9	1.6	399.03s
	NMMP	60	**96.6**	1.6	2.84s
Objects	baseline	119	76.8	1.8	–
	LDA	19	78.2	2.0	0.04s
	LMNN	119	84.1	1.8	221.29s
	NMMP	60	**86.5**	1.6	0.66s
USPS	baseline	79	93.2	1.1	–
	LDA	3	84.2	2.6	0.01s
	LMNN	79	86.2	2.2	70.83s
	NMMP	60	**94.5**	0.9	0.12s
News	baseline	119	30.9	2.8	–
	LDA	3	46.9	5.7	0.05s
	LMNN	119	**62.1**	7.3	73.38s
	NMMP	60	58.5	5.0	0.40s

Table 4: *Experimental results in each data set.*

It is worth noting that the parameters in our method are not sensitive. In fact, in each experiment, we simply set $k_b(i)$ to 10, and set $k_w(i)$ to $n_i/2+2$ for each class i, where n_i is the training number of class i.

The experimental results are reported in Table 4. We use the recognition result directly performed after the preprocessing by PCA as the baseline.

In the following we describe the details of each experiment.

The UCI data sets

In this experiment, we perform on two small data sets, Iris and Balance, taken from the UCI Machine Learning Repository[1]. As the class distributions of this two data sets are not very complex, LDA works well, and our method also demonstrates the competitive performance.

Face recognition

The AT&T face database (formerly the ORL database) includes 40 distinct individuals and each individual has 10 different images. Some images were taken at different times, and have variations [Samaria and Harter, 1994] including expression and facial details. Each image in the database is of size 112×92 and with 256 gray-levels.

In this experiment, no other preprocessings are performed except the PCA preprocessing step. The result of our method is much better than those of LDA and the baseline. LMNN have a good performance too, but the computation burden is extremely heavy.

We also perform the experiments on many other face databases, and obtain the similar results, say, our method demonstrates the much better performances uniformly.

Object recognition

The COIL-20 database [Nene *et al.*, 1996] consists of images of 20 objects viewed from varying angles at the interval of five degrees, resulting in 72 images per object.

In this experiment, each image is down-sampled to the size of 16×16 for saving the computation time.

Similar to the face recognition experiments, the results of our method and LMNN are much better than those of LDA

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

and the baseline. Note that both face images and object images distribute on an underlying manifold, the experiments verify that NMMP can preserve manifold structure with more discriminability than LDA.

Digit recognition

In this experiment, we focus on the digit recognition task using the USPS handwritten 16×16 digits data set[2]. The digits 1,2,3,and 4 are used in this experiment as the four classes. There are 1269, 929, 824 and 852 examples for each class, with a total of 3874.

On this data set, the baseline already works well, and our method still makes a little improvement. LDA fails in this case, which demonstrates that the available projection number of LDA may be insufficient when the data distributions are more complex than Gaussian.

Text categorization

In this experiment, we investigated the task of text categorization using the 20-newsgroups data set[3]. The topic *rec* which contains *autos*, *motorcycles*, *baseball*, and *hockey* was chosen from the version 20-news-18828. The articles were preprocessed with the same procedure as in [Zhou *et al.*, 2004]. This results in 3970 document vectors in a 8014-dimensional space. Finally the documents were normalized into TFIDF representation.

Our method and LMNN both bring significant improvements comparing with the baseline on this data set. In comparison, the performance of LDA is limited in that the available projection number of it is only 3, which is insufficient for this complex task.

7 Conclusion

In this paper, we propose a new method, *Neighborhood Min-Max Projections* (NMMP), for supervised dimensionality reduction. NMMP focuses only on the pairwise points where the two points are neighbors of each other. After the dimensionality reduction, NMMP minimizes the distance of the considered pairwise points within the same class, and maximizes the distance of those between different classes. In comparison with LDA, NMMP focuses more on the improvement of the discriminability of local structure. This property is especially useful when the distribution of class data is more complex than Gaussian. Toy example and real world experiments are presented to validate it. Moreover, other disadvantages of LDA i.e., the singularity problem of S_w and the limitation of the available number of dimension are also avoided in our method.

As a linear dimensionality reduction method, NMMP can be viewed as a special case of learning a Mahalanobis distance metric. Usually, the computation burden of learning a Mahalanobis distance metric is extremely heavy. Our method formulates the problem as a constrained optimization problem, and the global optimum can be effectively and efficiently obtained. Experiments demonstrate that our method has a competitive performance compared with the recent proposed Mahalanobis distance metric learning method, LMNN, but the computation cost is much lower.

[2]Available at http://www.kernel-machines.org/data

[3]Available at http://people.csail.mit.edu/jrennie/20Newsgroups/

References

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[Chen *et al.*, 2000] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new lda based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, October 2000.

[Duda. *et al.*, 2000] Richard O. Duda., Peter E. Hart., and David G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2000.

[Fukunaga, 1990] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition,Second Edition.* Academic Press, Boston, MA, 1990.

[Goldberger *et al.*, 2005] Jacob Goldberger, Sam Roweis, Geoffrey Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, Cambridge, MA, 2005.

[Golub and van Loan, 1996] Gene H. Golub and Charles F. van Loan. *Matrix Computations, 3rd Edition.* The Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[Guo *et al.*, 2003] Yue-Fei Guo, Shi-Jin Li, Jing-Yu Yang, Ting-Ting Shu, and Li-De Wu. A generalized foley-sammon transform based on generalized fisher discriminant criterion and its application to face recognition. *Pattern Recognition Letter*, 24(1-3):147–158, January 2003.

[Jolliffe, 2002] I. T. Jolliffe. *Principal Component Analysis,2nd Edition.* Springer-Verlag, New York, 2002.

[Nene *et al.*, 1996] S. A. Nene, S. K. Nayar, and H. Murase. *Columbia object image library (COIL-20), Technical Report CUCS-005-96*. Columbia University, 1996.

[Samaria and Harter, 1994] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[Weinberger *et al.*, 2006] Kilian Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*, pages 1475–1482. MIT Press, Cambridge, MA, 2006.

[Xing *et al.*, 2003] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 2003.

[Yu and Yang, 2001] H. Yu and J. Yang. A direct lda algorithm for high-dimensional data - with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001.

[Zhou *et al.*, 2004] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.