# Efficient and Robust Feature Extraction by Maximum Margin Criterion

Haifeng Li* and Tao Jiang
Department of Computer Science and Engineering
University of California, Riverside, CA 92521
{hli, jiang}@cs.ucr.edu

Keshu Zhang
Human Interaction Research Lab
Motorola, Inc., Tempe, AZ 85282
keshu.zhang@motorola.com

**Abstract**

In pattern recognition, feature extraction techniques are widely employed to reduce the dimensionality of data and to enhance the discriminatory information. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two most popular linear dimensionality reduction methods. However, PCA is not very effective for the extraction of the most discriminant features and LDA is not stable due to the *small sample size problem*. In this paper, we propose some new (linear and nonlinear) feature extractors based on *maximum margin criterion (MMC)*. Geometrically, feature extractors based on MMC maximize the (average) margin between classes after dimensionality reduction. It is shown that MMC can represent class separability better than PCA. As a connection to LDA, we may also derive LDA from MMC by incorporating some constraints. By using some other constraints, we establish a new linear feature extractor that does not suffer from the small sample size problem, which is known to cause serious stability problems for LDA. The kernelized (nonlinear) counterpart of this linear feature extractor is also established in the paper. Our extensive experiments demonstrate that the new feature extractors are effective, stable, and efficient.

**Keywords:** Feature extraction, maximum margin criterion, linear discriminant analysis, small sample size problem

*To whom correspondence should be addressed.

# 1 Introduction

In statistical pattern recognition, high dimensionality is a major cause of the practical limitations of many pattern recognition technologies. Moreover, it has been observed that a large number of features may actually degrade the performance of classifiers if the number of training samples is small relative to the number of features [16, 22, 23]. This fact, which is referred to as the "peaking phenomenon", is caused by the "curse of dimensionality" [2]. Actually, the number of parameters in a $D$-dimensional distribution usually goes up much faster than O($D$) unless one makes the very strong assumption that the features are independent. For instance, given that the features are not independent, the normal distributions have $O(D^2)$ parameters and the binary distributions have $O(2^D)$ parameters [4]. In other words, the complexity of a distribution increases rapidly when the dimensionality increases. So, very large data would be needed in general to train a classifier well. It is generally accepted that one needs at least ten times as many training samples per class as the number of features to obtain well-trained classifiers [7, 16, 22]. However, the number of samples is often small due to limitations on sample availability, identification, time, and cost. Consequently, dimensionality reduction is essential not only to engineering applications but also to the design of classifiers. In fact, the design of a classifier becomes extremely simple if all patterns in the same class hold the same feature vector which is different from the feature vectors held by patterns from other classes.

In the past several decades, many dimensionality reduction techniques have been proposed. The most well-known feature extraction methods may be Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA, also known as the Karhunen-Loève transformation in communication theory, is given by a linear transformation matrix $\mathbf{W} \in \mathcal{R}^{D\times d}$ minimizing the mean squared error criterion [17]. In PCA, $\mathbf{W}$ is constituted by the largest eigenvectors (called principal components) of the sample covariance (or correlation) matrix. The purpose of PCA is to keep the information in terms of variance as much as possible. By expanding data on these orthogonal principal components, one can obtain the minimal reconstruction error. But the decorrelation and high measures of statistical significance provided by the first few principal components cannot guarantee the revelation of necessary class structural information needed for a proper classification. Besides, the fact that the category information associated with patterns is neglected implies that PCA may be significantly sub-optimal.

Linear discriminant analysis (also called Fisher's Linear Discriminant) is another popular linear dimensionality reduction method. In many applications, LDA has proven to be much more effective than PCA. Fisher originally introduced LDA for two classes [6], while Rao generalized LDA to handle multi-class cases [21]. LDA is given by a linear transformation matrix $\mathbf{W} \in \mathcal{R}^{D\times d}$ maximizing the so-called Fisher criterion (a kind of *Rayleigh* coefficient)

$$J_F(\mathbf{W}) = tr\left(\frac{\mathbf{W}^T\mathbf{S}_b\mathbf{W}}{\mathbf{W}^T\mathbf{S}_w\mathbf{W}}\right) \tag{1}$$

where $\mathbf{S}_b = \sum_{i=1}^{c} p_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$ and $\mathbf{S}_w = \sum_{i=1}^{c} p_i \mathbf{S}_i$ are the between-class scatter matrix and the within-class scatter matrix, respectively; $c$ is the number of classes; $\mathbf{m}_i$ and $p_i$ are the mean vector and *a priori* probability of class $i$, respectively; $\mathbf{m} = \sum_{i=1}^{c} p_i \mathbf{m}_i$ is the overall mean vector; $\mathbf{S}_i$ is the within-class scatter matrix of class $i$; $D$ and $d$ are the dimensionalities of the data before and after the transformation, respectively; and $tr$ denotes the trace of a square matrix, i.e. the sum of the diagonal elements. Besides, the total/mixture scatter matrix, i.e. the covariance matrix of all samples regardless of their class assignments, is defined as $\mathbf{S}_t = E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T] = \mathbf{S}_w + \mathbf{S}_b$ [9]. To maximize (1), the transformation matrix $\mathbf{W}$ must be constituted by the largest eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$. A major drawback of LDA is that it cannot be applied when $\mathbf{S}_w$ is singular due to the *small sample size problem* [9]. The small sample size problem arises whenever the number of samples is smaller than the dimensionality of samples. The small sample size problem occurs frequently in practice, for example face recognition, cancer classification with gene expression profiling, and web document classification, etc. In the classification and diagnostic prediction of cancers using gene expression profiling, for example, the data have several thousand features (genes). Due to limitations on sample availability, identification, acquisition, time, and cost, however, the number of samples is usually only several dozen [11, 18, 20].

In recent years, many researchers have noticed this problem and tried to overcome the computational difficulty with LDA. A simple and direct attempt is to replace $\mathbf{S}_w^{-1}$ with the pseudo-inverse matrix $\mathbf{S}_w^{+}$ [27]. However, it does not guarantee that Fisher's criterion is still optimized by the largest eigenvectors of $\mathbf{S}_w^{+}\mathbf{S}_b$. Another approach is to first reduce the dimensionality with some other feature selection/extraction method and then apply LDA on the dimensionality-reduced data. For instance, Belhumeur *et al.* proposed the Fisherface (also called PCA+LDA) method which first employs PCA to reduce the dimensionality of the feature space to $n - c$, and then applies the standard LDA to reduce the dimensionality to $c - 1$, where $n$ is the number of samples and $c$ is the number of classes [1]. Note that this method is sub-optimal because PCA has to keep $n - 1$ principal components in order not to lose information. However, the first step of PCA+LDA keeps only $n - c$ principal components. Such a setting will lose too much information if the number of classes is large.

To handle the singularity problem, it is also popular to add a singular value perturbation to $\mathbf{S}_w$ to make it nonsingular [14]. A similar but more systematic method is regularized discriminant analysis (RDA) [8]. In RDA, one tries to obtain more reliable estimates of the eigenvalues by correcting the eigenvalue distortion in the sample covariance matrix with a ridge-type regularization. Besides, RDA is also a compromise between LDA and QDA (quadratic discriminant analysis), which allows one to shrink the separate covariances of QDA towards a common covariance as in LDA. Penalized discriminant analysis (PDA) is another regularized version of LDA [12, 13]. The goals of PDA are not only to overcome the small sample size problem but also to smooth the coefficients of discriminant vectors for better interpretation. In PDA, $\mathbf{S}_w$ is replaced with $\mathbf{S}_w + \lambda\Omega$

and then LDA proceeds as usual, where $\Omega$ is a symmetric and non-negative definite penalty matrix. The choice of $\Omega$ depends on the problem. If the data are log-spectra or images, $\Omega$ is defined in such a way so as to force nearby components of discriminant vectors to be similar. The main problem with RDA and PDA is that they do not scale well. In applications such as face recognition and cancer classification with gene expression profiling, the dimensionality of covariance matrices are often more than ten thousand. It is not practical for RDA and PDA to process such large covariance matrices, especially when the computing platform is made of PCs.

Recently, several methods that play with the null space of $\mathbf{S}_w$ have been widely investigated. A well-known null subspace method is the LDA+PCA method [5]. When $\mathbf{S}_w$ is of full rank, the LDA+PCA method just calculates the maximum eigenvectors of $\mathbf{S}_t^{-1}\mathbf{S}_b$ to form the transformation matrix. Otherwise, a two-stage procedure is employed. First, the data are transformed into the null space $\mathcal{V}_0$ of $\mathbf{S}_w$. Second, it tries to maximize the between-class scatter in $\mathcal{V}_0$, which is accomplished by performing PCA on the between-class scatter matrix in $\mathcal{V}_0$. Although this method solves the small sample size problem, it could be sub-optimal because it maximizes the between-class scatter in the null space of $\mathbf{S}_w$ instead of the original input space. For example, the performance of the LDA+PCA method drops significantly when $n - c$ is close to the dimensionality $D$, where $n$ is the number of samples and $c$ is the number of classes. The reason is that the dimensionality of the null space $\mathcal{V}_0$ is very small in this situation, and too much information is lost when we try to extract the discriminant vectors in $\mathcal{V}_0$. LDA+PCA also needs to calculate the rank of $\mathbf{S}_w$, which is an ill-defined operation due to floating-point imprecision. Another problem with LDA+PCA is that the computational complexity of determining the null space of $\mathbf{S}_w$ is very high. In [15], a more efficient null subspace method was proposed, which has the same accuracy as LDA+PCA in principle. This method first removes the null space of $\mathbf{S}_t$, which has been proven to be the common null space of both $\mathbf{S}_b$ and $\mathbf{S}_w$, and useless for discrimination. Then, LDA+PCA is performed in the lower-dimensional projected space. Direct LDA is another null space method that discards the null space of $\mathbf{S}_b$ [29]. This is achieved by diagonalizing first $\mathbf{S}_b$ and then diagonalizing $\mathbf{S}_w$, which is in the reverse order of conventional simultaneous diagonalization procedure [9]. In Direct LDA, one may also employ $\mathbf{S}_t$ instead of $\mathbf{S}_w$. In this way, Direct LDA is actually equivalent to the PCA+LDA [29]. Therefore, Direct LDA may be regarded as a "unified PCA+LDA" since there is no separate PCA step.

Kernel Fisher's Discriminant (KFD) [19] is a well-known nonlinear extension to LDA. The instability problem is more severe for KFD because $\mathbf{S}_w$ in the (nonlinear) feature space $\mathcal{F}$ is always singular (the rank of $\mathbf{S}_w$ is $n - c$). Similar to [14], KFD simply adds a perturbation $\mu I$ to $\mathbf{S}_w$. However, it is known that an eigenvector could be very sensitive to small perturbation if its corresponding eigenvalue is close to another eigenvalue of the same matrix [26]. Besides, it is hard to determine an optimal $\mu$ theoretically.

In this paper, a simple, efficient, and stable method is proposed to calculate the most dis-

criminant vectors and to avoid the small sample size problem based on a new feature extraction criterion, the *maximum margin criterion (MMC)*. Geometrically, MMC maximizes the (average) margin between classes. It can be shown that MMC represents class separability better than PCA. As a connection to Fisher's criterion, we may also derive LDA from MMC by incorporating some constraint. By using some other constraints, we establish the new linear and nonlinear feature extractors that do not suffer from the small sample size problem, which is known to cause serious stability problems for LDA. Different from LDA+PCA, the new feature extractors based on MMC maximize the between-class scatter in the input space instead of the null space of $\mathbf{S}_w$. Hence, it has a better overall performance than LDA+PCA, as confirmed by our experimental results.

The rest of the paper is organized as follows. Section 2 introduces the maximum margin criterion for feature extraction. Based on this criterion, Section 3 obtains an optimal linear feature extractor by solving an eigenvalue problem. Section 4 deals with nonlinear dimensionality reduction by kernelizing the above linear feature extractor. Section 5 discusses the experimental results on many different kinds of data. Section 6 presents the conclusions along with some directions for further research.

# 2 Maximum Margin Criterion

Suppose that we are given the empirical data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$$

Here, the domain $\mathcal{X} \in \mathcal{R}^D$ is some nonempty set that the patterns $\mathbf{x}_i$ are taken from. The $y_i$'s are called labels or targets. By studying these samples, we want to predict the label $y \in \{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$ of some new pattern $\mathbf{x} \in \mathcal{X}$. In other words, we choose $y$ such that $(\mathbf{x}, y)$ is in some sense similar to the training examples. For this purpose, some measure need be employed to assess similarity or dissimilarity. We want to keep such similarity/dissimilarity information as much as possible after the dimensionality reduction, *i.e.*, transforming $\mathbf{x}$ from $\mathcal{R}^D$ to $\mathcal{R}^d$, where $d \ll D$.

If some distance metric is used to measure the dissimilarity, we would hope that a pattern is close to those in the same class but far from those in different classes. So, a good feature extractor should maximize the distances between classes after the transformation. Therefore, we may define the feature extraction criterion as

$$J = \frac{1}{2} \sum_{i=1}^{c} \sum_{j=1}^{c} p_i p_j d(\mathcal{C}_i, \mathcal{C}_j) \tag{2}$$

We call (2) the maximum margin criterion (MMC). It is actually the summation of all pair interclass margins.

One may use the distance between mean vectors as the distance between classes, *i.e.*

$$d(\mathcal{C}_i, \mathcal{C}_j) = d(\mathbf{m}_i, \mathbf{m}_j) \tag{3}$$

where $\mathbf{m}_i$ and $\mathbf{m}_j$ are the mean vectors of the class $\mathcal{C}_i$ and the class $\mathcal{C}_j$, respectively. However, (3) is not suitable since it neglects the scatter of classes. Even if the distance between the mean vectors is large, it is not easy to separate two classes that have the large spread and overlap with each other. By considering the scatter of classes, we define the interclass distance (or margin) as

$$d(\mathcal{C}_i, \mathcal{C}_j) = d(\mathbf{m}_i, \mathbf{m}_j) - s(\mathcal{C}_i) - s(\mathcal{C}_j) \tag{4}$$

where $s(\mathcal{C}_i)$ is some measure of the scatter of the class $\mathcal{C}_i$. In statistics, we usually use the generalized variance $|\mathbf{S}_i|$ or overall variance $tr(\mathbf{S}_i)$ to measure the scatter of data, where $\mathbf{S}_i$ is the covariance matrix of the class $\mathcal{C}_i$. In this paper, we employ the overall variance $tr(\mathbf{S}_i)$ because it is easy to analyze. The weakness of the overall variance is that it ignores covariance structure altogether. Note that, by employing the overall/generalized variance, the expression (4) measures the "average margin" between two classes while the minimum margin is used in support vector machines (SVMs) [28].

With (4) and $s(\mathcal{C}_i)$ being $tr(\mathbf{S}_i)$, we may decompose (2) into two parts

$$\begin{aligned} J &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (d(\mathbf{m}_i, \mathbf{m}_j) - tr(\mathbf{S}_i) - tr(\mathbf{S}_j)) \\ &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(\mathbf{m}_i, \mathbf{m}_j) - \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (tr(\mathbf{S}_i) + tr(\mathbf{S}_j)) \end{aligned}$$

The second part is easily simplified to $tr(\mathbf{S}_w)$

$$\frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (tr(\mathbf{S}_i) + tr(\mathbf{S}_j)) = \sum_{i=1}^{c} p_i tr(\mathbf{S}_i) = tr\left(\sum_{i=1}^{c} p_i \mathbf{S}_i\right) = tr(\mathbf{S}_w) \tag{5}$$

By employing the Euclidean distance, we may also simplify the first part to $tr(\mathbf{S}_b)$ as follows

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(\mathbf{m}_i, \mathbf{m}_j) = \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j) \\ &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (\mathbf{m}_i - \mathbf{m} + \mathbf{m} - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m} + \mathbf{m} - \mathbf{m}_j) \end{aligned}$$

After expanding it, we can simplify the above equation to $\sum_{i=1}^{c} p_i (\mathbf{m}_i - \mathbf{m})^T (\mathbf{m}_i - \mathbf{m})$ by using the fact $\sum_{j=1}^{c} p_j (\mathbf{m} - \mathbf{m}_j) = 0$. So

$$\frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(\mathbf{m}_i, \mathbf{m}_j) = tr\left(\sum_{i=1}^{c} p_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T\right) = tr(\mathbf{S}_b) \tag{6}$$

Now we obtain

$$J = tr(\mathbf{S}_b - \mathbf{S}_w) \tag{7}$$

Since $tr(\mathbf{S}_b)$ measures the overall variance of the class mean vectors, a large $tr(\mathbf{S}_b)$ implies that the class mean vectors scatter in a large space. On the other hand, a small $tr(\mathbf{S}_w)$ implies that every class has a small spread. Thus, a large $J$ indicates that patterns are close to each other if they are from the same class but are far from each other if they are from different classes. Thus, this criterion may represent class separability better than PCA. Recall that PCA tries to maximize the total scatter $tr(\mathbf{S}_t)$ after a linear transformation. But the data set with a large within-class scatter can also have a large total scatter even when it has a small between-class scatter because $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$. Obviously, such data are not easy to classify. For LDA and MMC, it is clear that both have a very similar goal. Later, we will discuss the connection and difference between them in detail.

# 3 Linear Feature Extraction

When performing dimensionality reduction, we want to find a (linear or nonlinear) mapping from the measurement space $\mathcal{M}$ to some feature space $\mathcal{F}$ such that $J$ is maximized after the transformation. In this section, we discuss how to find an optimal linear feature extractor. In the next section, we will generalize it to the nonlinear case.

Consider a linear mapping $\mathbf{W} \in \mathcal{R}^{D\times d}$ , where $D$ and $d$ are the dimensionalities of the data before and after the transformation, respectively. We would like to maximize

$$J(\mathbf{W}) = tr(\mathbf{S}_b^W - \mathbf{S}_w^W)$$

where $\mathbf{S}_b^W$ and $\mathbf{S}_w^W$ are the between- and within-class scatter matrices in the feature space $\mathcal{F}$, respectively. Since $\mathbf{W}$ is a linear mapping, it is easy to show $\mathbf{S}_b^W = \mathbf{W}^T\mathbf{S}_b\mathbf{W}$ and $\mathbf{S}_w^W = \mathbf{W}^T\mathbf{S}_w\mathbf{W}$. So, we have

$$J(\mathbf{W}) = tr\left(\mathbf{W}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{W}\right) \tag{8}$$

In this formulation, we have the freedom to multiply $\mathbf{W}$ by some nonzero constant. Thus, we additionally require that $\mathbf{W}$ is constituted by the unit vectors, *i.e.* $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d]$ and $\mathbf{w}_k^T\mathbf{w}_k = 1$. This means that we need solve the following constrained optimization

$$\begin{aligned} \max \quad & \sum_{k=1}^{d} \mathbf{w}_k^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{w}_k \\ \text{subject to} \quad & \mathbf{w}_k^T\mathbf{w}_k - 1 = 0 \qquad k = 1, \ldots, d \end{aligned}$$

Note that we may also use other constraints instead. For example, we may require $tr\left(\mathbf{W}^T\mathbf{S}_w\mathbf{W}\right) = 1$ and then maximize $tr\left(\mathbf{W}^T\mathbf{S}_b\mathbf{W}\right)$. It is easy to show that maximizing MMC with such a constraint in fact results in LDA. The only difference is that it involves a constrained optimization

whereas the traditional LDA solves an unconstrained optimization. The motivation for using the constraint $\mathbf{w}_k^T\mathbf{w}_k = 1$ is that it allows us to avoid calculating the inverse of $\mathbf{S}_w$ and thus the potential small sample size problem.

To solve the above optimization problem, we may introduce a Lagrangian

$$\mathcal{L}(\mathbf{w}_k, \lambda_k) = \sum_{k=1}^{d} \mathbf{w}_k^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{w}_k - \lambda_k(\mathbf{w}_k^T\mathbf{w}_k - 1) \tag{9}$$

with multipliers $\lambda_k$. The Lagrangian $\mathcal{L}$ has to be maximized with respect to $\lambda_k$ and $\mathbf{w}_k$. The condition that at the stationary point, the derivatives of $\mathcal{L}$ with respect to $\mathbf{w}_k$ must vanish

$$\frac{\partial\mathcal{L}(\mathbf{w}_k, \lambda_k)}{\partial\mathbf{w}_k} = ((\mathbf{S}_b - \mathbf{S}_w) - \lambda_k\mathbf{I})\mathbf{w}_k = 0 \qquad k = 1, \ldots, d \tag{10}$$

leads to

$$(\mathbf{S}_b - \mathbf{S}_w)\mathbf{w_k} = \lambda_k\mathbf{w}_k \qquad k = 1, \ldots, d \tag{11}$$

which means that the $\lambda_k$'s are the eigenvalues of $\mathbf{S}_b - \mathbf{S}_w$ and the $\mathbf{w}_k$'s are the corresponding eigenvectors. Thus

$$J(\mathbf{W}) = \sum_{k=1}^{d} \mathbf{w}_k^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{w}_k = \sum_{k=1}^{d} \lambda_k\mathbf{w}_k^T\mathbf{w}_k = \sum_{k=1}^{d} \lambda_k \tag{12}$$

Therefore, $J(\mathbf{W})$ is maximized when $\mathbf{W}$ is composed of the first $d$ largest eigenvectors of $\mathbf{S}_b - \mathbf{S}_w$. Here, we need not calculate the inverse of $\mathbf{S}_w$, which allows us to avoid the small sample size problem easily. We may also require $\mathbf{W}$ to be orthonormal, which may help preserve the shape of the distribution.

In what follows, we discuss how to calculate the eigenvectors of $\mathbf{S}_b - \mathbf{S}_w$ in an efficient way and to determine the optimal dimensionality $d$ of the feature space. First, we rewrite $\mathbf{S}_b - \mathbf{S}_w = 2\mathbf{S}_b - \mathbf{S}_t$ since $\mathbf{S}_w = \mathbf{S}_t - \mathbf{S}_b$. Note that the null space of $\mathbf{S}_t$ is a subspace of that of $\mathbf{S}_b$ since the null space of $\mathbf{S}_t$ is the common null space of $\mathbf{S}_b$ and $\mathbf{S}_w$ [15]. Thus, we can simultaneously diagonalize $\mathbf{S}_b$ and $\mathbf{S}_t$ to $\mathbf{\Lambda} = \text{diag}[\lambda_1, \ldots, \lambda_r]$ and $\mathbf{I}$ [9]:

$$\mathbf{P}^T\mathbf{S}_b\mathbf{P} = \mathbf{\Lambda} \tag{13}$$

$$\mathbf{P}^T\mathbf{S}_t\mathbf{P} = \mathbf{I} \tag{14}$$

where $\mathbf{P} \in \mathcal{R}^{D\times r}$, $D$ is the dimensionality of input space, and $r$ is the rank of $\mathbf{S}_t$, which is at most $\min\{n-1, D\}$. Besides, $\mathbf{\Lambda} \geq 0$ since $\mathbf{S}_b$ is positive semidefinite. Note that there are at most $c-1$ eigenvalues $\lambda_i > 0, i = 1, \ldots, r$, because the rank of $\mathbf{S}_b$ is $c-1$ where $c$ is the number of classes. The transformation matrix $\mathbf{P}$ is given by $\mathbf{\Phi\Theta}^{-1/2}\mathbf{\Psi}$, where $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are the eigenvalue and eigenvector matrices of $\mathbf{S}_t$, and $\mathbf{\Psi}$ is the eigenvector matrix of $\mathbf{\Theta}^{-1/2}\mathbf{\Phi}^T\mathbf{S}_b\mathbf{\Phi\Theta}^{-1/2}$ [9]. Clearly, the columns of $\mathbf{P}$ are the eigenvectors of $2\mathbf{S}_b - \mathbf{S}_t$ with the corresponding eigenvalues $2\mathbf{\Lambda} - \mathbf{I}$. Based on the above equalities, we have the following results:

- According to the maximum margin criterion, an eigenvector with an eigenvalue $2\lambda_i - 1 \geq 0$ means that samples in different classes are well separated (on average) in the direction of this eigenvector. In contrast, samples from different classes overlap in the direction of the eigenvectors of $2\lambda_i - 1 < 0$.

- Based on Equation (12), we should choose the eigenvectors such that $2\lambda_i - 1 \geq 0$ to constitute the mapping matrix $\mathbf{W}$ maximizing the maximum margin criterion. Although eigenvectors of $2\lambda_i - 1 = 0$ do not increase the maximum margin criterion, we would still like to employ these eigenvectors for feature extraction because feature extraction is not only to reduce the dimensionality but also to keep as much information as possible. Extending this argument, we may also employ the eigenvectors with the eigenvalues such that $2\lambda_i - 1$ are slightly less than 0.

- To calculate the eigenvector matrix $\mathbf{P} = \mathbf{\Phi}\mathbf{\Theta}^{-1/2}\mathbf{\Psi}$, we use a fast two-step algorithm in virtue of singular value decomposition (SVD). SVD expresses a real $n \times m$ matrix $\mathbf{A}$ as a product $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}^T$, where $\mathbf{\Lambda}^{\frac{1}{2}}$ is a diagonal matrix with decreasing non-negative entries, and $\mathbf{U}$ and $\mathbf{V}$ are $n \times \min(n, m)$ and $m \times \min(n, m)$ orthonormal column matrices [10]. The columns of $\mathbf{U}$ and $\mathbf{V}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$, respectively, and the nonvanishing entries of $\mathbf{\Lambda}^{\frac{1}{2}}$ are the square roots of the non-zero eigenvalues of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$. Note that $\mathbf{S}_t$ is estimated by $\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$ and can be expressed in the form $\mathbf{S}_t = \mathbf{X}\mathbf{X}^T$ with $\mathbf{X} = \frac{1}{\sqrt{n}}[(\mathbf{x}_1 - \mathbf{m}), \ldots, (\mathbf{x}_n - \mathbf{m})]$. Thus, we can obtain the eigenvalues $\mathbf{\Theta}$ and the corresponding eigenvectors $\mathbf{\Phi}$ of $\mathbf{S}_t$ through the SVD of $\mathbf{X}$. Note that dimensionality of $\mathbf{X}$ is $D \times n$, where $n$ is the number of samples. When $n$ is much smaller than $D$ (i.e., small sample size problem happens), the SVD of $\mathbf{X}$ is much faster than the eigen decomposition of $\mathbf{S}_t$. To obtain the eigenvector matrix $\mathbf{\Psi}$ of $\mathbf{\Theta}^{-1/2}\mathbf{\Phi}^T\mathbf{S}_b\mathbf{\Phi}\mathbf{\Theta}^{-1/2}$, we observe $\mathbf{S}_b = \mathbf{M}\mathbf{M}^T$ with $\mathbf{M} = [\sqrt{p_1}(\mathbf{m}_1 - \mathbf{m}), \ldots, \sqrt{p_c}(\mathbf{m}_c - \mathbf{m})]$. Hence, we can get $\mathbf{\Psi}$ through the SVD of $\mathbf{\Theta}^{-1/2}\mathbf{\Phi}^T\mathbf{M}$.

- When $\mathbf{S}_w$ is nonsingular (so is $\mathbf{S}_t$), the transformation matrix $\mathbf{P}$ is also the eigenvector matrix of $\mathbf{S}_t^{-1}\mathbf{S}_b$ [9]. In this sense, LDA and MMC are equal because the eigenvectors of $\mathbf{S}_t^{-1}\mathbf{S}_b$ and $\mathbf{S}_w^{-1}\mathbf{S}_b$ have the same discriminatory capability [9]. However, the eigenvalues of $\mathbf{S}_t^{-1}\mathbf{S}_b$ (and $\mathbf{S}_w^{-1}\mathbf{S}_b$) are all nonnegative. Thus, all eigenvectors with nonnegative eigenvalues are employed to constitute the mapping matrix of LDA in practice. As we mentioned before, however, samples are not well separated in the direction of eigenvectors whose corresponding eigenvalues of $\mathbf{S}_b - \mathbf{S}_w$ are negative. In principle, MMC can make the samples more separable than LDA by discarding such eigenvectors. On the other hand, however, MMC may lose more information by discarding these eigenvectors. In experiments, we observe that MMC is usually able to achieve similar results as LDA in a lower dimensional discriminant space.

- When the small samples size problem arises, MMC is actually equivalent to LDA+PCA in general. Recall that LDA+PCA maximizes $\mathbf{S}_b$ in the null space of $\mathbf{S}_w$. According to Equation (12), it is easy to show that, in order to maximize $tr(\mathbf{W}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{W})$, the columns of $\mathbf{W}$ have to be in the null space of $\mathbf{S}_w$ but not in the null space of $\mathbf{S}_b$ since both $\mathbf{S}_w$ and $\mathbf{S}_b$ are positive semidefinite. Thus, MMC and LDA+PCA will be equal if the null space of $\mathbf{S}_w$ is large enough to contain the whole column space of $\mathbf{S}_b$. However, MMC and LDA+PCA will be different when the number of samples minus the number of classes is close to the dimensionality of input space. In this case, the null space of $\mathbf{S}_w$ will be very small and cannot contain the whole column space of $\mathbf{S}_b$. Thus, LDA+PCA may lose too much information and results in very poor performance. In contrast, MMC works in the whole input space rather than only in the null space of $\mathbf{S}_w$ and can keep more discriminatory information. Thus, MMC can achieve better results. Besides, MMC is much simpler and more efficient than LDA+PCA. Note that LDA+PCA needs to determine the rank and the null space of $\mathbf{S}_w$, which is time-consuming. Especially, it is ill-defined to calculate the rank of a matrix due to floating-point imprecisions.

# 4 Nonlinear Feature Extraction with Kernel

In this section, we follow the approach of nonlinear SVMs [28] to kernelize the above linear feature extractor. That is, we employ some positive definite kernel to implicitly perform a nonlinear mapping $\Phi$ to transform the data into a feature space $\mathcal{F}$, in which the linear feature extraction is applied. More precisely, we first reformulate the maximum margin criterion in terms of only dot-product $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ of input patterns. Then we replace the dot-product by some positive definite kernel $k(\mathbf{x}, \mathbf{y})$, *e.g.* polynomial kernel $\langle x, y \rangle^p$, Gaussian kernel $e^{-\gamma \|x-y\|^2}$, etc.

Consider the maximum margin criterion in the feature space $\mathcal{F}$

$$J^{\Phi}(\mathbf{W}) = \sum_{k=1}^{d} \mathbf{w}_k^T (\mathbf{S}_b^{\Phi} - \mathbf{S}_w^{\Phi}) \mathbf{w}_k$$

where $\mathbf{S}_b^{\Phi}$ and $\mathbf{S}_w^{\Phi}$ are the between-class scatter matrix and within-class scatter matrix in $\mathcal{F}$, *i.e.*, $\mathbf{S}_b^{\Phi} = \sum_{i=1}^{c} p_i (\mathbf{m}_i^{\Phi} - \mathbf{m}^{\Phi})(\mathbf{m}_i^{\Phi} - \mathbf{m}^{\Phi})^T$, $\mathbf{S}_w^{\Phi} = \sum_{i=1}^{c} p_i \mathbf{S}_i^{\Phi}$ and $\mathbf{S}_i^{\Phi} = \frac{1}{n_i} \sum_{j=1}^{n_i} (\mathbf{\Phi}(\mathbf{x}_j^{(i)}) - \mathbf{m}_i^{\Phi})(\mathbf{\Phi}(\mathbf{x}_j^{(i)}) - \mathbf{m}_i^{\Phi})^T$ with $\mathbf{m}_i^{\Phi} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{\Phi}(\mathbf{x}_j^{(i)})$, $\mathbf{m}^{\Phi} = \sum_{i=1}^{c} p_i \mathbf{m}_i^{\Phi}$, and $\mathbf{x}_j^{(i)}$ is the pattern of class $\mathcal{C}_i$ that has $n_i$ samples. Note that the small sample size problem always happens in $\mathcal{F}$ because the feature space $\mathcal{F}$ has very high or even infinite dimensionality. With a polynomial kernel of degree $p$, for example, the dimensionality $D'$ of $\mathcal{F}$ is $D' = \binom{D+p-1}{p}$ given a $D$ dimensional input space. For $D = 256$ and $p = 4$, $D' \approx 10^9$. Clearly, the small sample size problem cannot be avoided for such a nonlinear mapping in practice.

Since $\mathbf{S}_b^\Phi$ and $\mathbf{S}_w^\Phi$ have the very high dimensionality, it is not possible to calculate them directly in practice. To avoid the computation of $\mathbf{S}_b^\Phi$ and $\mathbf{S}_w^\Phi$, we notice that each $\mathbf{w}_k$ lies in the span of $\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_n)$. Therefore, we can find an expansion for $\mathbf{w}_k$ in the form $\mathbf{w}_k = \sum_{l=1}^n \alpha_l^{(k)} \Phi(\mathbf{x}_l)$. Using this expansion and the definition of $\mathbf{m}_i^\Phi$, we have

$$\mathbf{w}_k^T \mathbf{m}_i^\Phi = \sum_{l=1}^{n} \alpha_l^{(k)} \left( \frac{1}{n_i} \sum_{j=1}^{n_i} \langle \mathbf{\Phi}(\mathbf{x}_l), \mathbf{\Phi}(\mathbf{x}_j^{(i)}) \rangle \right)$$

Replacing the dot-product by some kernel function $k(\mathbf{x}, \mathbf{y})$ and defining $(\widetilde{\mathbf{m}}_i)_l = \frac{1}{n_i} \sum_{j=1}^{n_i} k(\mathbf{x}_l, \mathbf{x}_j^{(i)})$, we get $\mathbf{w}_k^T \mathbf{m}_i^\Phi = \boldsymbol{\alpha}_k^T \widetilde{\mathbf{m}}_i$ with $(\boldsymbol{\alpha}_k)_l = \alpha_l^{(k)}$. Similarly, we have

$$\mathbf{w}_k^T \mathbf{m}^\Phi = \mathbf{w}_k^T \sum_{i=1}^{c} p_i \mathbf{m}_i^\Phi = \boldsymbol{\alpha}_k^T \sum_{i=1}^{c} p_i \widetilde{\mathbf{m}}_i = \boldsymbol{\alpha}_k^T \widetilde{\mathbf{m}}$$

with $\widetilde{\mathbf{m}} = \sum_{i=1}^c p_i \widetilde{\mathbf{m}}_i$. This means $\mathbf{w}_k^T(\mathbf{m}_i^\Phi - \mathbf{m}^\Phi) = \boldsymbol{\alpha}_k^T(\widetilde{\mathbf{m}}_i - \widetilde{\mathbf{m}})$ and

$$\begin{aligned}
&\sum_{k=1}^{d} \mathbf{w}_k^T \mathbf{S}_b^\Phi \mathbf{w}_k = \sum_{k=1}^{d} \sum_{i=1}^{c} p_i (\mathbf{w}_k^T(\mathbf{m}_i^\Phi - \mathbf{m}^\Phi))(\mathbf{w}_k^T(\mathbf{m}_i^\Phi - \mathbf{m}^\Phi))^T \\
&= \sum_{k=1}^{d} \sum_{i=1}^{c} p_i^T \boldsymbol{\alpha}_k^T (\widetilde{\mathbf{m}}_i - \widetilde{\mathbf{m}})(\widetilde{\mathbf{m}}_i - \widetilde{\mathbf{m}})^T \boldsymbol{\alpha}_k = \sum_{k=1}^{d} \boldsymbol{\alpha}_k^T \widetilde{\mathbf{S}}_b \boldsymbol{\alpha}_k
\end{aligned}$$

where $\widetilde{\mathbf{S}}_b = \sum_{i=1}^c p_i (\widetilde{\mathbf{m}}_\mathbf{i} - \widetilde{\mathbf{m}})(\widetilde{\mathbf{m}}_\mathbf{i} - \widetilde{\mathbf{m}})^T$.

Similarly, one can simplify $\mathbf{W}^T \mathbf{S}_w^\Phi \mathbf{W}$. First, we have $\mathbf{w}_k^T(\Phi(\mathbf{x}_j^{(i)}) - \mathbf{m}_i^\Phi) = \boldsymbol{\alpha}_k^T(\mathbf{k}_j^{(i)} - \widetilde{\mathbf{m}}_i)$ with $(\mathbf{k}_j^{(i)})_l = k(\mathbf{x}_l, \mathbf{x}_j^{(i)})$. Considering $\mathbf{w}_k^T \mathbf{S}_i^\Phi \mathbf{w}_k = \frac{1}{n_i} \sum_{j=1}^{n_i} (\mathbf{w}_k^T(\Phi(\mathbf{x}_j^{(i)}) - \mathbf{m}_i^\Phi))(\mathbf{w}_k^T(\Phi(\mathbf{x}_j^{(i)}) - \mathbf{m}_i^\Phi))^T$, we have

$$\begin{aligned}
\mathbf{w}_k^T \mathbf{S}_i^\Phi \mathbf{w}_k &= \frac{1}{n_i} \sum_{j=1}^{n_i} \boldsymbol{\alpha}_k^T (\mathbf{k}_j^{(i)} - \widetilde{\mathbf{m}}_i)(\mathbf{k}_j^{(i)} - \widetilde{\mathbf{m}}_i)^T \boldsymbol{\alpha}_k \\
&= \frac{1}{n_i} \sum_{j=1}^{n_i} \boldsymbol{\alpha}_k^T \mathbf{K}_i (\mathbf{e}_j - \frac{1}{n_i} \mathbf{1}_{n_i})(\mathbf{e}_j - \frac{1}{n_i} \mathbf{1}_{n_i})^T \mathbf{K}_i^T \boldsymbol{\alpha}_k \\
&= \frac{1}{n_i} \sum_{j=1}^{n_i} \boldsymbol{\alpha}_k^T \mathbf{K}_i (\mathbf{e}_j \mathbf{e}_j^T - \frac{1}{n_i} \mathbf{e}_j \mathbf{1}_{n_i}^T - \frac{1}{n_i} \mathbf{1}_{n_i} \mathbf{e}_j^T + \frac{1}{n_i^2} \mathbf{1}_{n_i} \mathbf{1}_{n_i}^T) \mathbf{K}_i^T \boldsymbol{\alpha}_k \\
&= \frac{1}{n_i} \boldsymbol{\alpha}_k^T \mathbf{K}_i (\mathbf{I}_{n_i \times n_i} - \frac{1}{n_i} \mathbf{1}_{n_i} \mathbf{1}_{n_i}^T) \mathbf{K}_i^T \boldsymbol{\alpha}_k
\end{aligned}$$

where $(\mathbf{K}_i)_{lj} = k(\mathbf{x}_l, \mathbf{x}_j^{(i)})$, $I_{n_i \times n_i}$ is the $n_i \times n_i$ identity matrix, $\mathbf{1}_{n_i}$ is the $n_i$-dimensional vector of 1's, and $\mathbf{e}_j$ is the canonical basis vector of $n_i$ dimensions. Thus, we obtain

$$\begin{aligned}
&\sum_{k=1}^{d} \mathbf{w}_k^T \mathbf{S}_w^\Phi \mathbf{w}_k = \sum_{k=1}^{d} \sum_{i=1}^{c} p_i \frac{1}{n_i} \boldsymbol{\alpha}_k^T \mathbf{K}_i (\mathbf{I}_{n_i} - \frac{1}{n_i} \mathbf{1}_{n_i} \mathbf{1}_{n_i}^T) \mathbf{K}_i^T \boldsymbol{\alpha}_k \\
&= \sum_{k=1}^{d} \boldsymbol{\alpha}_k^T \left( \sum_{i=1}^{c} p_i \frac{1}{n_i} \mathbf{K}_i (\mathbf{I}_{n_i} - \frac{1}{n_i} \mathbf{1}_{n_i} \mathbf{1}_{n_i}^T) \mathbf{K}_i^T \right) \boldsymbol{\alpha}_k = \sum_{k=1}^{d} \boldsymbol{\alpha}_k^T \widetilde{\mathbf{S}}_w \boldsymbol{\alpha}_k
\end{aligned}$$

where $\widetilde{\mathbf{S}}_w = \sum_{i=1}^{c} p_i \frac{1}{n_i}\mathbf{K}_i(\mathbf{I}_{n_i} - \frac{1}{n_i}\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T)\mathbf{K}_i^T$. So the maximum margin criterion in the feature space $\mathcal{F}$ is

$$J(\mathbf{W}) = \sum_{k=1}^{d} \boldsymbol{\alpha}_k^T(\widetilde{\mathbf{S}}_b - \widetilde{\mathbf{S}}_w)\boldsymbol{\alpha}_k \tag{15}$$

Similar to the observations in developing the linear feature extractor based on MMC, one may obtain that the above criterion is maximized by the largest eigenvectors of $\widetilde{\mathbf{S}}_b - \widetilde{\mathbf{S}}_w$ with constraints $\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k = 1, k = 1, \ldots, d$. However, it is not appropriate to obtain the mapping matrix directly through the eigen decomposition of $\widetilde{\mathbf{S}}_b - \widetilde{\mathbf{S}}_w$. According to our previous discussion, the null space of $\mathbf{S}_w^\Phi$ contains important discriminatory information. However, we lose a very large portion of the null space of $\mathbf{S}_w^\Phi$ while we work with $\widetilde{\mathbf{S}}_w$ through the kernel trick. Note that the dimensionality of $\mathrm{NULL}(\widetilde{\mathbf{S}}_w)$ is usually only $c$. But the dimensionality of $\mathrm{NULL}(\mathbf{S}_w^\Phi)$ is much larger or even infinite. To avoid this problem, we should instead employ the maximum margin criterion in the feature space as

$$J(\mathbf{W}) = \sum_{k=1}^{d} \boldsymbol{\alpha}_k^T(2\widetilde{\mathbf{S}}_b - \widetilde{\mathbf{S}}_t)\boldsymbol{\alpha}_k \tag{16}$$

where $\widetilde{\mathbf{S}}_t = \frac{1}{n}\mathbf{K}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{K}^T$ and $(\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ [25]. In this case, we can directly use the eigenvectors of $2\widetilde{\mathbf{S}}_b - \widetilde{\mathbf{S}}_t$ with the largest eigenvalues to constitute the mapping matrix because the null space of $\mathbf{S}_t^\Phi$ is useless for discriminant analysis. As we discussed in the linear case, we may obtain the mapping matrix $\boldsymbol{\Lambda} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_k]$ by simultaneously diagonalizing $\widetilde{\mathbf{S}}_t$ and $\widetilde{\mathbf{S}}_b$.

# 5 Experiments

In this section, we compare our methods with LDA, LDA+PCA, and kernel PCA on several different datasets. All the experiments are performed on a Pentium 2.4G and 1G RAM machine. First, we will compare MMC with LDA on the famous Fisher's iris dataset. which has no small sample size problem. The iris dataset gives the measurements in centimeters of the sepal length and width and the petal length and width, respectively, of 50 flowers from each of 3 species of iris. The eigenvalues of $\mathbf{S}_b - \mathbf{S}_w$ is $0.97$ and $-0.06$. Since there is only one positive eigenvalue, we use the corresponding eigenvector to constitute the map matrix of MMC. For LDA, we use two (i.e. $c-1$) eigenvectors as usual. The experimental procedure is as follows. After feature extraction by MMC or LDA, we use the nearest centroid method to predict the class of a new observation $\mathbf{x}$ as

$$\mathcal{C}(\mathbf{x}) = \arg\min_k \sum_{i=1}^{d} (\mathbf{w}_i^T(\mathbf{x} - \mathbf{m}_k))^2 \tag{17}$$

i.e. the class whose mean vector is closest to $\mathbf{x}$ in the space of discriminant variables. To estimate the error rates, we randomly split the dataset into three parts in a class-proportional manner, of

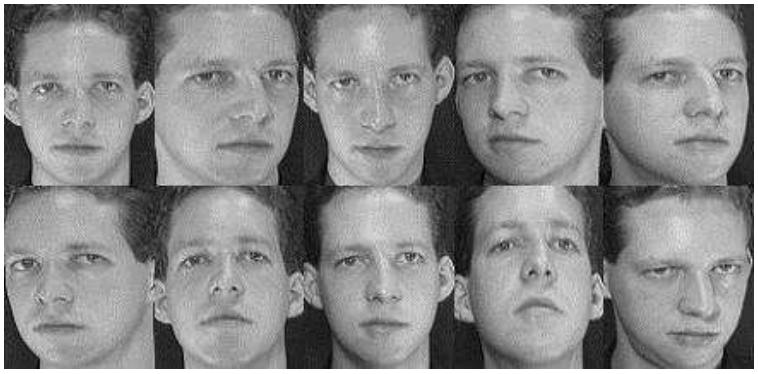

Figure 1: Samples from the ORL dataset.

which two parts are used for training and the third part is kept for test. This procedure is repeated for 200 times and the average error rates of LDA and MMC are $2.06\%$ and $1.94\%$, respectively. This shows that MMC can achieve a competing performance to LDA even though it maps the data into a lower dimensional discriminant space.

Because the linear methods already achieves very high accuracies on the iris dataset, we employ another dataset, the vehicle dataset from STATLOG databases, to compare kernel MMC and kernel PCA. The vehicle dataset contains 18 features of 846 silhouettes of four types of vehicle. The vehicle may be viewed from one of many different angles. On the vehicle dataset, LDA and MMC achieve a $23.84\%$ error rate. In contrast, kernel MMC achieves a $19.39\%$ error rate. In this experiment, we employ the normalized polynomial kernel $\frac{k(x,y)}{\sqrt{k(x,x)}\sqrt{k(y,y)}}$, where $k(\cdot,\cdot)$ is the polynomial kernel. The degree of the polynomial kernel is set to 2. For LDA, MMC, and kernel MMC, we map the data into a three-dimensional (i.e. $c-1$) discriminant space. We also apply kernel PCA [25] on this dataset. We use the eigenvectors with eigenvalues larger than $0.01$ to constitute the mapping matrix. The number of such eigenvectors is dependent on the selected training data. With the same kernel as the one employed by kernel MMC, there are usually around 14 such eigenvectors. After feature extraction by kernel PCA, a support vector machine is employed to classify samples as Schölkopf *et al.* did in [25]. The average error rate is $22.71\%$, which is only slightly better than those of the linear methods LDA and MMC.

In what follows, we will compare MMC with LDA+PCA on the small sample size datasets. The original LDA+PCA [5] needs to determine the null space of $\mathbf{S}_w$, which is very time consuming. So, a pixel grouping method is applied before LDA+PCA in [5] [1]. Huang *et al.* improved LDA+PCA by first removing the null space of $\mathbf{S}_t$ firstly. The algorithm of Huang *et al.* has the same recognition accuracy as the original LDA+PCA but is more (both time and memory) efficient and can be applied to the original (unresized) datasets directly. Thus, we compare MMC with the algorithm

[1] In the previous conference version of this paper, we simply resized the images to a smaller size

Table 1: The averages and standard deviations of error rates on the ORL dataset. The first column is the number of samples per class for training. The kernel MMC and kernel PCA use Gaussian kernel with $\gamma = 0.0075$ and $\gamma = 0.0025$, respectively. MMC, LDA+PCA, and kernel MMC reduce the dimensionality to $c - 1$. Kernel PCA uses the largest eigenvectors that preserve $95\%$ of the variance.

| Training samples | MMC | LDA+PCA | kernel MMC | kernel PCA |
|---|---|---|---|---|
| 3 | $8.90 \pm 2.37$ | $8.90 \pm 2.37$ | $9.13 \pm 2.39$ | $12.17 \pm 2.42$ |
| 4 | $5.71 \pm 1.47$ | $5.71 \pm 1.47$ | $5.82 \pm 1.64$ | $9.53 \pm 1.95$ |
| 5 | $3.89 \pm 1.43$ | $3.89 \pm 1.43$ | $3.82 \pm 1.33$ | $9.00 \pm 2.16$ |
| 6 | $3.12 \pm 1.40$ | $3.12 \pm 1.40$ | $2.91 \pm 1.53$ | $9.19 \pm 1.88$ |
| 7 | $2.20 \pm 1.38$ | $2.20 \pm 1.38$ | $1.95 \pm 1.31$ | $9.93 \pm 2.64$ |

of Huang *et al.* instead of the original LDA+PCA here. In the experiment, we used the ORL face dataset [24], which had been used to investigate the small sample size problem in [15, 29, 30]. Figure 1 depicts some sample images of a typical subset in the ORL dataset. The ORL dataset consists of 10 face images from 40 subjects for a total of 400 images, with some variation in pose, facial expression and details. The resolution of the images is $112 \times 92$, with 256 gray-levels. Before the experiments, we scale the pixel gray level to the range $[0, 1]$. We conduct a series of experiments on ORL dataset and compared our results with those obtained using LDA+PCA and kernel PCA. The average error rates of 50 runs are shown in Table 1. As we discussed before, MMC and LDA+PCA (the algorithm of Huang *et al.*) achieve the same error rates. [2] Besides, we observe that kernel MMC just achieves similar results as linear MMC and LDA+PCA although it has a significantly better performance than kernel PCA. The reason is that the input space has already a very high dimensionality (10304) so that the samples are scattered and are easy to classify. With the kernel trick, we just map the data into a higher dimensional space. Thus, it does not improve the performance. When the dimensionality of the input space is small, the kernel methods have the superiority as shown on the vehicle dataset and in the next experiment.

Now we will show that our method performs much better than LDA+PCA when $n-c$ is close to the dimensionality $D$. Because the amount of training data is limited, we resize the images of ORL to 168 dimensions to create such a situation. We randomly select five images per person for training and use the remaining images for test. For such a setting, $n - c = 200 - 40 = 160$, which is close to $168$. Thus, the dimensionality of the null space of $\mathbf{S}_w$ is around $168 - 160 = 8$, which is very small with respect to $c - 1 = 39$. So, the performance of LDA+PCA (both the original algorithm

[2] Although MMC, the algorithm of Huang *et al.*, and the original LDA+PCA have the same discriminatory capability in principle, we observe that MMC and the algorithm of Huang *et al.* often obtain better results than the original LDA+PCA in practice. The reason may be that, when the original LDA+PCA calculates the rank and the null space of $\mathbf{S}_w$, the cumulative error due to floating-point imprecisions is significant since $\mathbf{S}_w$ is usually huge.

and the one of Huang *et al.*) will drop significantly because it loses a lot of information when it maximizes the between-class scatter in this small null space. In fact, LDA+PCA has a $64.84\%$ average error rate. On the other hand, our method tries to maximize the between-class scatter in the whole input space and achieves a much better error rate of $12.96\%$. Besides, kernel MMC and kernel PCA could obtain the error rates of $5.29\%$ and $7.91\%$ with Gaussian kernels ($\gamma = 0.058$ for kernel MMC and $\gamma = 0.019$ for kernel PCA), respectively. This confirms our above argument that kernel methods can obtain better results than linear methods when the dimensionality of input space is relatively small.

To demonstrate that our method is effective not only for face recognition but also for other applications, we apply it to cancer classification with gene expression data. Accurate diagnosis of human cancer is essential in cancer treatment. Recently, the advances in microarray technology enable us to simultaneously observe the expression levels of many thousands of genes on the transcription level. In principle, tumor gene expression profiles can serve as molecular fingerprints for cancer (phenotype) classification. Researchers believe that gene expression profiling could be a precise, objective, and systematic approach for cancer diagnosis [11, 18]. In this experiment, we apply our method on a brain cancer gene expression dataset, which is the dataset A in [20]. This dataset contains 42 samples of 10 medulloblastomas, 10 malignant gliomas, 10 atypical teratoid/rhabdoid tumors (AT/RTs), 8 primitive neuroectodermal tumors (PNETs), and 4 normal cerebella. The brain dataset contains the expression of 5597 genes (i.e. features). Clearly, it is a typical small sample size problem. On this dataset, MMC and kernel MMC achieve $17.54\%$ and $18.31\%$ average error rates (based on 50 runs), where kernel MMC employ a Gaussian kernel with $\gamma = 0.00005$. In contrast, kernel PCA obtains a $32.31\%$ error rate with the same kernel. We also run the random forest method [3] on this dataset. The random forest algorithm is a state-of-art classification method that grows many classification trees. To classify a new sample, the final prediction is voted by all the trees in the forest. Because the random forest algorithm implicitly selects features (i.e. dimensionality reduction) when growing a classification tree, we would like to compare our method with it. On the brain cancer dataset, the random forest algorithm obtains a $23.08\%$ error rate, which is significantly higher than those of our methods. It is another evidence that our methods are able to extract the most discriminatory information (among the methods compared above) for classification.

# 6 Conclusion

In pattern recognition, feature extraction techniques are widely employed to reduce the dimensionality of data and to enhance the discriminatory information. In this paper, we have proposed some linear and nonlinear feature extractors based on the maximum margin criterion. The new methods can effectively extract the most discriminatory features and do not suffer from the small sample

size problem. The experimental results show that the new methods are effective and robust. In practice, the proposed nonlinear feature extractor could be slow when the dataset is large. It is an interesting topic to develop a fast algorithm for the proposed nonlinear feature extractor.

## Acknowledgments

We thank D. Gunopulos, C. Domeniconi, and J. Peng for valuable discussions and comments. The vehicle dataset is from the Turing Institute, Glasgow, Scotland. The ORL face dataset is from the Olivetti Research Laboratory in Cambridge, UK. The brain cancer dataset is from the Broad Institute. This work was partially supported by NSF grants CCR-9988353, ACI-0085910, and CCR-0309902.

## References

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

[2] R. E. Bellman. *Adaptive Control Precesses: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, CA, 1984.

[5] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

[6] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual of Eugenics*, 7:179–188, 1936.

[7] D. H. Foley. Considerations of sample and feature size. *IEEE Transactions on Information Theory*, 18(5):618–626, 1972.

[8] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, 1990.

[10] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[11] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 536:531–537, 1999.

[12] T. Hastie and R. Tibshirani. Penalized discriminant analysis. *The Annals of Statistics*, 23:73–102, 1995.

[13] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89:1255–1270, 1994.

[14] Z.-Q. Hong and J.-Y. Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24(4):317–324, 1991.

[15] R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the small sample size problem of LDA. In *Proceedings of 16th International Conference on Pattern Recognition*, volume 3, pages 29–32, 2002.

[16] A. K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. In P. Krishnaiah and L. Kanal, editors, *Handbook of Statistics*, volume 2, pages 835–855. Amsterdam, North Holland, 1982.

[17] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[18] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679, 2001.

[19] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[20] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. S. G, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, 2002.

[21] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10:159–203, 1948.

[22] S. J. Raudys and A. K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991.

[23] S. J. Raudys and V. Pikelis. On dimensionality, sample size, classification error, and complexity of classification algorithms in pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:243–251, 1980.

[24] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, 1994.

[25] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Compution*, 10:1299–1319, 1998.

[26] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[27] Q. Tian, M. Barbero, Z.-H. Gu, and S. H. Lee. Image classification by the foley-sammon transform. *Optical Engineering*, 25(7):834–840, 1986.

[28] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[29] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data — with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.

[30] W. Zhao, R. Chellappa, and P. Phillips. Subspace linear discriminant analysis for face recognition. Technical Report CAR-TR-914, Center for Automation Research, University of Maryland, College Park, 1999.