Face Recognition Using Kernel Ridge Regression

Senjian An, Wanquan Liu and Svetha Venkatesh
Dept. of Computing, Curtin University of Technology
GPO Box U1987, Perth, WA 6845, Australia.
senjian, wanquan, svetha@cs.curtin.edu.au

Abstract

In this paper, we present novel ridge regression (RR) and kernel ridge regression (KRR) techniques for multivariate labels and apply the methods to the problem of face recognition. Motivated by the fact that the regular simplex vertices are separate points with highest degree of symmetry, we choose such vertices as the targets for the distinct individuals in recognition and apply RR or KRR to map the training face images into a face subspace where the training images from each individual will locate near their individual targets. We identify the new face image by mapping it into this face subspace and comparing its distance to all individual targets. An efficient cross-validation algorithm is also provided for selecting the regularization and kernel parameters. Experiments were conducted on two face databases and the results demonstrate that the proposed algorithm significantly outperforms the three popular linear face recognition techniques (Eigenfaces, Fisherfaces and Laplacianfaces) and also performs comparably with the recently developed Orthogonal Laplacianfaces with the advantage of computational speed. Experimental results also demonstrate that KRR outperforms RR as expected since KRR can utilize the nonlinear structure of the face images. Although we concentrate on face recognition in this paper, the proposed method is general and may be applied for general multi-category classification problems.

1. Introduction

Face recognition has attracted tremendous attention in the computer vision community over the past few decades and many new techniques have been developed. Among them the appearance-based method is one of the most successful and well-studied techniques. In appearance-based methods, the image is represented by a high dimensional vector of pixels. To overcome the difficulty incurred by high dimensionality, a lot of subspace methods, such as the Eigenfaces [24], the Fisherfaces [1] and its variants [27, 25, 26, 3, 16, 28, 17, 15], the Laplacianfaces [9] and orthogonal Laplacianfaces [2], have been developed. Eigenfaces applies Principle Component Analysis (PCA) to project the original n-dimensional data onto a low dimensional subspace which preserves the most of the data variations. Fisherfaces uses Linear Discriminant Analysis (LDA) to find the most discriminant eigenvectors which maximizes the ratio of between-class and within-class variances. Unlike PCA, LDA is a supervised learning algorithm and its eigenvectors are usually nonorthogonal. The Laplacianfaces and Orthogonal Laplacianfaces are proposed by using Locality Preserving Projections (LPP) [8]. LPP maps each face to a low-dimensional face subspace which is characterized by a set of feature images, called *Laplacianfaces*. Unlike Eigenfaces which seeks projections that are efficient for face representation and Fisherfaces which seeks projections that are efficient for discrimination, Laplacianfaces seeks projections to preserve the local structure of the image space [9].

Face recognition is typically a multi-category classification problem. While LDA tries to maximize the between-class distances (the sum of all the pairwise distances of any two distinct classes) and minimize the within-class distances simultaneously, the pairwise distances can be significantly unbalanced and this may result in bad performance for classes with small pairwise between-class distances in the reduced subspace. Figure 1 illustrates an example for face recognition involving 3 persons. By applying Fisherfaces, one finds a two dimensional subspace wherein the within-classes distances are approximately zero, i.e, the training images for each individual locate near one point ($*$,$\times$, or $\circ$). However, the pairwise distances may be unbalanced as shown in Figure 1 (a). The distance between class 1 ($*$) and class 2 ($\times$) is much smaller than the other two pairwise between-class distances. Figure 1 (b) presents the balanced case where all the pairwise distances are identical. If the norms of the dimension reduction matrices for these two cases are approximately equal, one can expect that case (b) will generalize and perform better with unlabeled images than case (a). The three vertices of an equilateral triangle, as shown in Figure 1 (b), are three separate

1-4244-1180-7/07/$25.00 ©2007 IEEE

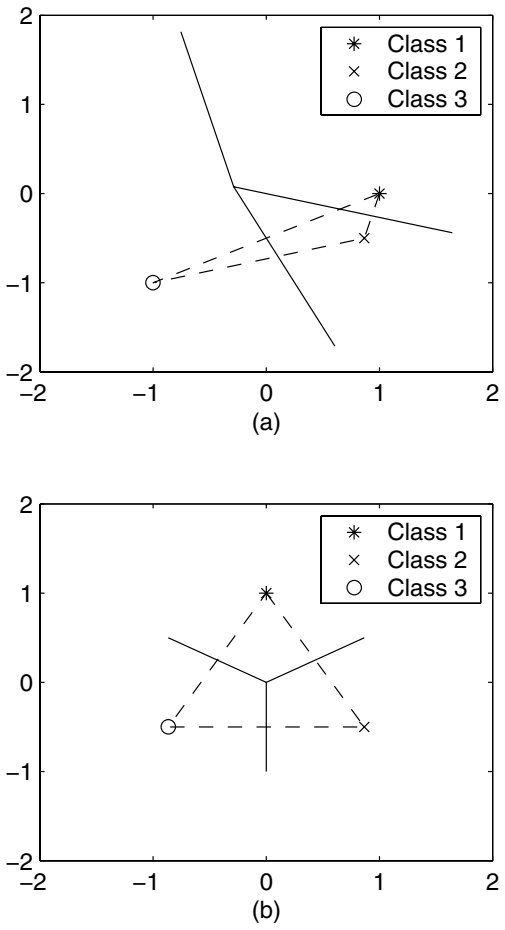

Figure 1. Illustration of an irregular Simplex (a) and a Regular Simplex (b). The hard lines denote the classification boundaries.

points in the plane with highest degree of symmetry and balance. They have equal pairwise distances and any two equilateral triangles in the same plane are congruent, i.e., they are identical under translation, rotation and reflection. The regular m-simplex [13], which is the m-dimensional analogue of a equilateral triangle, has the same property. Motivated by the fact that the m vertices of a regular m-simplex is the most balanced and symmetric separate points in the $(m-1)$-dimensional space, we choose these vertices as the targets for m distinct individuals, and apply ridge regression (RR) [11, 10] to map the training face images into the $(m-1)$-dimensional subspace. RR is a regularized least square method to model the linear dependency between covariate variables and univariate labels. The ordinary least square method minimizes the squared loss but the variance on the estimate of the linear transform may be large due to limited samples and thus not reliable. Ridge regression can reduce the variance by penalizing the norm of the linear transform and balance the bias and variance by adjusting the regularization parameter. We generalize RR for multivariate labels in order to apply it for face recognition. With the regular simplex vertices as the individual targets, the generalized RR minimizes the distances of the training images to their individual targets with a penalty on the norm of the dimension reduction matrix. The new unlabeled face image is identified by mapping into the reduced face subspace and comparing its distances to the individual targets. A nonlinear extension, which can exploit the nonlinear structure of face images, is also proposed using the kernel trick [21].

There are three major contributions in this paper. First, the original RR and kernel ridge recognition (KRR) [21] for univariate labels are generalized for multivariate labels so that they can be applied for face recognition; Second, a new face recognition technique is developed by applying the generalized RR or KRR to map the face images into a face subspace where the face images have approximately equal pairwise distances for any two distinct individuals. The proposed algorithm gains in discrimination by forcing all the training images from each individual to locate near one of the vertices of a regular simplex. Third, an efficient cross-validation algorithm is developed for selecting the regularization parameter and the kernel parameters of the generalized RR and KRR.

The layout of the rest in this paper is as follows. In Section 2, we briefly review the formulation of RR and KRR for univariate labels. Section 3 addresses the generalized RR and KRR for multi-variate labels. In Section 4, a new face recognition process based on generalized RR and KRR will be proposed. Section 5 develops an efficient cross-validation algorithm for model selection and Section 6 provides experimental results on two face databases to illustrate the performance of the proposed algorithm with comparison to some existing popular face recognition techniques.

2. Kernel Ridge Regression for Univariate Labels

In this section we briefly review RR and KRR for univariate labels. Linear ridge regression is a classical statistical problem that aims to find a linear function that models the dependencies between covariates $\{x_i\}_{i=1}^n$ in $\mathbb{R}^p$ and response variables $\{y_i\}_{i=1}^n$ in $\mathbb{R}$. The classical way is the ordinary least square (OLS) method which minimizes the squared loss:

$$\sum_i (y_i - w^T x_i)^2. \tag{1}$$

Due to limited training examples, the variance of the estimate w by OLS may be large and thus the estimate is not reliable. An effective way to overcome this problem is to penalize the norm of w as in ridge regression. Instead of minimizing squared errors, ridge regression minimizes the

following cost:

$$J(w) = \sum_i (y_i - w^T x_i)^2 + \lambda \|w\|^2 \quad (2)$$

where λ is a fixed positive number. By introducing the regularization parameter λ, the ridge regression can reduce the estimate variance at the expense of increasing training errors. The regularization parameter λ controls the trade-off between the bias and variance of the estimate. In practice, one can use cross-validation [19] to find the optimal regularization parameter that minimizes the cross-validation errors.

In [21], it is shown that the predicted label (i.e., $w^T x$) of a new unlabeled example x is:

$$y^T (K + \lambda I)^{-1} \kappa \quad (3)$$

where K is the matrix of dot products of the vectors $\{x_i, i = 1, 2, \cdots, n\}$ in the training set:

$$K_{i,j} = x_i^T x_j, i, j = 1, 2, \cdots, n,$$

and κ is the vector of dot products of x and the vectors in the training set:

$$\kappa_i = x_i^T x, i = 1, 2, \cdots, n.$$

With this formulation, it is easy to generalize RR to KRR using the kernel trick [21]. The data is now replaced with the feature vectors: $x_i \rightarrow \Phi_i = \Phi(x_i)$ induced by a kernel where $k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$. By replacing K and κ with $K_{i,j} = k(x_i, x_j), \kappa_i = k(x, x_i)$ for $i = 1, 2, \cdots, n, j = 1, 2, \cdots, n$, (3) is then the KRR predictor. Here $k(\cdot, \cdot)$ is the kernel function which is typically linear $k(x_i, x_j) = x_i^T x_j$, polynomial $(x_i^T x_j + 1)^d$ or Gaussian $k(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / \sigma^2\right)$. It is important to note that, in the KRR predictor, we do not actually need to access the feature vectors as long as we can access the kernel function.

3. Kernel Ridge Regression for Multivariate Labels

In [21], KRR is investigated for univariate labels, i.e., the label y_i is a real number. In order to apply RR and KRR for face recognition, which is a multi-category classification problem, we need to generalize RR and KRR to the multivariate label case, where the labels Y_i are vectors in $\mathbb{R}^r$. The task of the generalized RR is to find a matrix $W \in \mathbb{R}^{p \times r}$ that can model the linear dependency between x_i and the label Y_i. It is natural to choose the cost as:

$$\sum_i \|Y_i - W^T x_i\|^2. \quad (4)$$

Similar to RR for univariate labels, we penalize the norm of W to reduce the variance of the estimate and the total cost is given by:

$$\begin{aligned} J(W) &= \textstyle\sum_i \|Y_i - W^T x_i\|^2 + \lambda \|W\|^2 \\ &= \mathrm{tr}\left(Y Y^T + W^T X X^T W - 2 W^T X Y^T\right) \\ &\quad + \lambda \mathrm{tr}\left(W^T W\right). \end{aligned} \quad (5)$$

where $X = [x_1, x_2, \cdots, x_n]$ and $Y = [Y_1, Y_2, \cdots, Y_n]$. In the above derivation, we use the fact that $\mathrm{tr}(ab^T) = a^T b$ for any vectors a, b with equal dimensions.

Taking derivatives and equaling them to zero, we have

$$W = (X X^T + \lambda I)^{-1} X Y^T \quad (6)$$

Now we replace the training patterns x_i with their feature vectors $x_i \rightarrow \Phi_i = \Phi(x_i)$ which is induced by a kernel function $k(\cdot, \cdot)$, i.e., $k(x_i, x_j) = \Phi_i^T \Phi_j$, and replace the cost $J(W)$ with

$$J_2(W) = \sum_i \|Y_i - W^T \Phi_i\|^2 + \lambda \|W\|^2. \quad (7)$$

Applying the formula (6), one has

$$W = (\Phi \Phi^T + \lambda I)^{-1} \Phi Y^T \quad (8)$$

where $\Phi = [\Phi_1, \cdots, \Phi_n]$.

By using the following formula [18] on matrix manipulations,

$$(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = P B^T (B P B^T + R)^{-1}, \quad (9)$$

we have

$$(\lambda I + \Phi \Phi^T)^{-1} \Phi = \Phi (\Phi^T \Phi + \lambda I)^{-1} \quad (10)$$

and therefore

$$W = \Phi (\Phi^T \Phi + \lambda I)^{-1} Y^T = \Phi (K + \lambda I)^{-1} Y^T \quad (11)$$

where $K_{i,j} = \Phi_i^T \Phi_j = k(x_i, x_j)$. The predicted label for a new example x is then

$$\begin{aligned} Y(x) &= W^T \Phi(x) \\ &= Y (K + \lambda I)^{-1} \Phi^T \Phi(x) \\ &= Y (K + \lambda I)^{-1} \kappa(x) \end{aligned} \quad (12)$$

where $\kappa(x) = [k(x_1, x), k(x_2, x), \cdots, k(x_n, x)]^T$.

Hence, we never need to access the feature vectors as long as we can access the kernel function. The predictor

$$Y(x) = A^T \kappa(x)$$

can be obtained by solving the following linear matrix equation:

$$(K + \lambda I) A = Y^T. \quad (13)$$

In the next section, we apply the generalized RR and KRR to face recognition.

4. Face Recognition Using Ridge Regression or Kernel Ridge Regression

Suppose we have m individuals for recognition. We choose the regular simplex vertices as the individual targets and use the individual targets as the multivariate labels of the training images. It can be proved that all regular m-simplexes in $\mathbb{R}^{m-1}$ with pairwise distance 1 are congruent [13]. That is, all regular m-simplexes with pairwise distance 1 are identical under translation, rotation and reflection. Let T_i ($\in \mathbb{R}^{m-1}$), $i = 1, 2, \cdots, m$, be the vertices of one regular m-simplex and denote $T = [T_1, T_2, \cdots, T_m]$. One can construct T as follows. First, let $T_1 = [1, 0, \cdots, 0]^T$ and let $T_{i,1} = -1/(m-1)$ for $i = 2, \cdots, m$. Hereafter, we use $T_{i,j}$ to denote the element of T in the ith row and jth column. Then, we have the first row and the first column. Now suppose we have got the first $k(\geq 1)$ rows and k columns, we compute the next row as

$$\begin{array}{rcl} T_{k+1,k+1} & = & \sqrt{1 - \sum_{i=1}^{k} T_{i,k}^2} \\ T_{k+1,j} & = & -\frac{T_{k+1,k+1}}{m-k-1}, j = k+2, \cdots, m, \end{array} \quad (14)$$

and let $T_{i,k+1} = 0$ for $(m-1) \geq i > k+1$.This procedure is repeated until $k = m-2$ and will give all the vertices T_i. It is easy to check that $\sum_i T_i = 0, T_i^T T_i = 1, i = 1, 2, \cdots, m$, and

$$\|T_i - T_j\| = 2 - 2T_i^T T_j = 2 + \frac{2}{m-1} \quad (15)$$

i.e., these targets have zero mean, unit norm and have equal pairwise distances. When $m = 3$, $T_1 = [1, 0]^T, T_2 = [-\frac{1}{2}, \frac{\sqrt{3}}{2}]^T$ and $T_3 = [-\frac{1}{2}, -\frac{\sqrt{3}}{2}]^T$. T_1, T_2, T_3 are three vertices of an equilateral triangle as shown in Figure 1 (b).

Using these targets as multivariate labels of the training images $\{x_i, i = 1, 2, \cdots, n\}$, we apply RR to find the dimensional reduction matrix W

$$W = (XX^T + \lambda I)^{-1} XY^T \quad (16)$$

which minimizes the cost $J(W)$ in (5). Here $X = [x_1, x_2, \cdots, x_n]$ and the ith column of Y equals T_j if the ith image is from individual j. The regularization parameter λ is usually a small number and then $X^T W \approx Y^T$. Therefore, the training images are mapped into a $(m-1)$-dimensional subspace where the images from each individual will locate near their individual targets.

Let x be a new image. We compare the distances between $W^T x$ and the individual targets T_i and identify x as that with minimal distance.

In summary, the proposed face recognition algorithm using RR includes the following three steps

1. *Compute the dimension reduction matrix W by (16);*
2. *For a new unlabeled image x, project it into the reduced subspace, that is, compute $\hat{x} = W^T x$;*
3. *Compute the distances from $\hat{x}$ to all the individual targets $T_i, i = 1, 2, \cdots, m$ and identify image x as individual j if $\|x - T_j\|$ is minimal.*

For KRR, the process is similar but it works in a kernel-induced feature space. From (13), we have

$$A = (K + \lambda I)^{-1} Y^T \quad (17)$$

where $K_{ij} = k(x_i, x_j)$. The predictor for a new image x is

$$p(x) = A^T [k(x, x_1), k(x, x_2), \cdots, k(x, x_n)]^T$$

and x is identified by comparing the distances between $p(x)$ and the individual targets T_i.

In summary, given the kernel function, say the Gaussian kernel

$$k(x, z) = e^{\frac{\|x-z\|^2}{2\sigma^2}},$$

the proposed face recognition technique through KRR includes the following four steps:

1. *Evaluate the kernel matrix K with $K_{i,j} = k(x_i, x_j)$;*
2. *Compute the dimension reduction matrix A from (17);*
3. *For a new unlabeled image x, compute $\kappa(x) = [k(x, x_1), k(x, x_2), \cdots, k(x, x_n)]^T$ and $\hat{x} = A^T \kappa(x)$;*
4. *Compute the distances from $\hat{x}$ to all the individual targets $T_i, i = 1, 2, \cdots, m$ and identify image x as individual j if $\|x - T_j\|$ is minimal.*

5. Model Selection by Cross-Validation

In the proposed face recognition techniques using RR or KRR, the model includes some hyper-parameters such as the kernel parameter and the regularization parameter that govern the generalization performance of KRR predictors. Finding the hyper-parameters with a good generalization performance is crucial for the successful application of RR and KRR [7, 5]. A popular way to estimate the generalization performance of a model is cross-validation [19]. In l-fold cross-validation, one divides the data into l subsets of (approximately) equal size and trains the classifier l times, each time leaving out one of the subsets from training, but using the omitted subset to compute the classification errors. If l equals the sample size, this is called leave-one-out cross-validation (LOO-CV).

The naive implementation of l-fold cross-validation trains a predictor for each split of the data and is thus computationally expensive if l is large, especially for LOO-CV where $l = n$. In [6], an efficient algorithm is developed for computing the leave-one-out errors of KRR for univariate labels. The algorithm computes the predicted labels directly

without training the predictors for each split and this will reduce the computational complexity to be approximately the same as that for one training. In this section, we will use the same idea to develop an efficient algorithm for general l-fold cross-validation of generalized RR and KRR with multivariate labels.

We splits the data into l subsets $\{x_{k,i}\}_{i=1}^{n_k}$ of (approximately) equal size (n_v), i.e., $n_k \approx n_v \approx n/l$, where $k = 1, 2, \cdots, l$ and $\sum_{k=1}^{l} n_k = n$. Correspondingly, we split the label Y and the solution A of (13) into l submatrices as follows:

$$Y^T = \begin{bmatrix} Y_{(1)} \\ Y_{(2)} \\ \ddots \\ Y_{(l)} \end{bmatrix}, A = \begin{bmatrix} A_{(1)} \\ A_{(2)} \\ \ddots \\ A_{(l)} \end{bmatrix} \tag{18}$$

where

$$Y_{(k)} = \begin{bmatrix} Y_{k,1}^T \\ Y_{k,2}^T \\ \ddots \\ Y_{k,n_k}^T \end{bmatrix}. \tag{19}$$

In cross-validating KRR, the predictor for each training set is not really of interest. One is only concerned with the predicted labels of the left-out examples. Next, we will derive the formula for l-fold cross-validation to directly compute the predicted labels of the left-out examples. This formula is based on the inverse of the system matrix of (13).

Let us exclude the kth group from the training patterns and train KRR on the remaining patterns. Then the predicted labels, denoted by $Y_{cv}^{(k)}$, of the kth group patterns can be computed as follows:

$$Y_{cv}^{(k)} = Y_{(k)} - C_{kk}^{-1} A_{(k)}, k = 1, 2, \cdots, l. \tag{20}$$

where

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1l} \\ C_{12}^T & C_{22} & \cdots & C_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1l}^T & C_{2l}^T & \cdots & C_{ll} \end{bmatrix} \triangleq (K + \lambda I)^{-1} \tag{21}$$

and $C_{ij} \in \mathbb{R}^{n_i \times n_j}$ for $i, j = 1, 2, \cdots, l$.

The proof is delegated to the appendix.

Once $(K + \lambda I)^{-1}$ is available, one can compute A via (17) and obtain C_{kk} from (21), and then $Y_{cv}^{(k)}$ is available from (20).

Note that the dimension of C_{kk} is approximately n_v which is much smaller than $(n - n_v)$ in general. Thus, (20) is generally more efficient than training the KRR predictor based on $(n - n_k)$ examples.

In summary, the proposed l-fold cross-validation algorithm includes the following steps.

1. *Evaluate the kernel matrix K and compute $(K + \lambda I)^{-1}$;*
2. *Compute A and C_{kk} from (17) and (21) respectively;*
3. *Compute the predicted response $Y_{cv}^{(k)}$ from (20);*
4. *Identify the images by comparing the distances from the predicted labels $Y_{cv}^{(k)}$ to the individual targets $\{T_i, i = 1, 2, \cdots, m\}$;*
5. *Sum up all recognition errors.*

In the naive implementation of l-fold cross-validation, one trains the KRR classifiers l times, each time leaving out one of the subsets from training, and using the omitted subset to compute the classification errors. This implementation involves the inverse of l matrices of dimensions $(n - n_k) \times (n - n_k)$. Note that the complexity of computing the inverse of an $m \times m$ matrix is m^3 [20] and $n_v \approx \frac{n}{l}$, the complexity of the naive l-fold CV is $l(n - n_v)^3 \approx \frac{(l-1)^3}{l^2} n^3$. In the special case when $n_v = 1, l = n$, this reduces to the LOO-CV and the computational complexity is $n(n-1)^3 \approx n^4$.

On the other hand, the proposed algorithm involves one inverse of an $n \times n$ matrix and the inverse of l $n_v \times n_v$ matrices and thus its complexity is $n^3 + l n_v^3 \approx [1 + \frac{1}{l^2}] n^3$. Hence, the proposed algorithm is $\frac{(l-1)^3}{1+l^2} \approx l - 3$ times as efficient as the naive implementation. In the case that $l = n$, this reduces to LOO-CV and the complexity is $n^3 + n$ which is much more efficient than the naive implementation.

6. Experimental Results

Experiments were conducted on two databases: CMU PIE [22, 23] and The Extended Yale Face Database B (YaleB) [4, 14] to test the performances of the proposed algorithm with comparisons to the most popular face recognition methods: Eigenfaces, Fisherfaces and the recently developed methods Laplacianfaces and Orthogonal Laplacianfaces. The CMU PIE face database contains 68 individuals with 41368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. The extended Yale Face Database B [14] contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions. The data format of this database is the same as the original Yale Face Database B [4]. Our experiments adopt the same procedure as that in the study by [2]. From CMU PIE, we choose the five near frontal poses (C05,C07,C09,C27,C29) and use all the 11544 images under different illuminations, lighting and expressions where each individual has 170 images except for a few bad images. From the Extended and the original Yale Face Database B,

we choose all the 2414 frontal images for 38 people. All test image data used in the experiments are manually aligned, cropped, and then re-sized to 32x32 images.

A random subset with $l(= 5, 10, 20, 30)$ images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given l, we average the results over 50 random splits and we used the same splits and the same matlab data files [1] which were used in [2]. For KRR, we used the Gaussian kernel $k(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$. The kernel parameter σ and the regularization parameter λ are selected based on Leave-one-out errors of the training images in the first 5 splits.

Table 1. Performance (error rate) comparison on CMU PIE face database.

Method	5 Train	10 Train	20 Train	30 Train
Eigen	69.9%(338)	55.7%(654)	38.1%(889)	27.9%(990)
Fisher	31.5%(67)	22.4%(67)	15.4%(67)	7.77%(67)
Lap	30.8%(67)	21.1%(134)	14.1%(146)	7.13%(131)
O-Lap	21.4%(108)	11.4%(265)	6.51%(493)	4.83%(423)
RR	25.97%(67)	14.06%(67)	7.69%(67)	5.89%(67)
KRR	26.4%(67)	13.1%(67)	5.97%(67)	4.02 %(67)

Table 2. Performance (error rate) comparison on the Extended Yale Face Database B.

Method	5 Train	10 Train	20 Train	30 Train
Eigen	63.6%(188)	46.4%(378)	30.4%(736)	22.5%(799)
Fisher	24.5%(37)	12.5%(37)	8.7%(37)	13.3%(37)
Lap	24%(37)	11.4%(76)	7.1%(193)	7.5%(251)
O-Lap	22.1%(108)	9.7%(111)	3.8%(247)	1.9%(406)
RR	23.8%(37)	12%(37)	4.77%(37)	2.28%(37)
KRR	23.9%(37)	11.04%(37)	3.67%(37)	1.43%(37)

Table 3. Computation time (seconds) comparison on the PIE database with 10 training images per individual.

Method	5 Train	10 Train	20 Train	30 Train
O-Lap	27.5	364.6	2140.8	1948.3
KRR	0.31	1.16	5.75	15.99

The performance is shown in Table 1 and Table 2. The performance for Eigenfaces (Eigen), Fisherfaces (Fisher), Laplacianfaces (Lap) and Orthogonal Laplacianfaces (O-Lap) are taken from [2] for CMU PIE database and from http://ews.uiuc.edu/ dengcai2/Data/data.html. for the Extended Yale Face Database B. The numbers in the brackets are the best dimensions for Eigenfaces, Laplacianfaces and Orthogonal Laplacianfaces. The performances of RR and KRR are significantly better than Eigenface, Fisherface and Laplacianface. Compared with the Orthogonal Laplacianfaces, the performance is comparable but our proposed algorithm is more computationally efficient as shown in Table 3. Note that the performance of Laplacianfaces and orthogonal Laplacianfaces shown in Table 1 and Table 2 are the best performances among all possible dimensions. In practice, one needs to find the best dimension in the training stage, say, via cross-validation. The orthogonal Laplacianfaces is computationally expensive because it requires the dimension reduction matrix to be orthogonal. Unlike Fisherfaces and Laplacianfaces who can compute the dimension reduction matrix by one generalized eigenvalue decomposition, the orthogonal Laplacianfaces computes its dimension reduction column by column and the computation of each column involves a generalized eigenvalue decomposition. Thus it will involve r generalized eigenvalue decompositions if the best dimension is r. From Table 1 and Table 2, one can see that the best dimensions for orthogonal Laplacianfaces are quite high. Note that, in Table 3, the computation time for 30Train is less than that for 20Train. This is due to the fact that the best dimension (423) for 30Train is less than the best dimension (493) for 20Train in this experiment. However, in order to find the best dimension, one needs to try higher dimensions than the best one and the training on 30 images per individual will be more computationally expensive than training on 20 images per individual.

[1] which were downloaded from http://ews.uiuc.edu/ dengcai2/Data/data.html

7. Conclusions

We have proposed a new face recognition technique based on the generalized RR and KRR for multivariate labels. The new technique chooses the regular simplex vertices as the targets for individuals in recognition and applies the generalized RR or KRR to minimize the training images' distances to their individual targets with a penalty on the norm of the dimension reduction matrix. An efficient cross-validation algorithm is also provided for selecting good regularization parameters and kernel parameters. Experimental results demonstrate the proposed algorithms performs well.

Although we focus on face recognition in this paper, the proposed method is general and may be applied for other type multi-category classification problems. One may also apply RR or KRR on multi-category classification problems with other type of labels, say the Error Correcting Output Coding (ECOC) labels proposed in [12], which essentially decomposes a multi-category classification problem into a set of complementary two-category problems.

References

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

[2] D. Cai, X. He, H. J., and Z. H.-J. Orthogonal laplacianfaces for face recognition. *IEEE Trans. Image Processing*, 15(11):3608–3614, 2006. 1, 5, 6

[3] 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. 1

[4] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition undervariable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2005. 5

[5] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979. 4

[6] I. Guyon. *Kernel Ridge Regression*. From http://clopinet.com/isabelle/Projects/ETH/KernelRidge.pdf, 2005. 4

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001. 4

[8] X. He and P. Niyogi. Locality preserving projections. In *Proc. Conf. Advances in Neural Information Processing Systems (NIPS'03)*, 2003. 1

[9] X. He, S. Yan, H. Y., N. P., and Z. H.-J. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(3):328–340, 2005. 1

[10] A. E. Hoerl and R. W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970. 2

[11] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. 2

[12] J. Kittler, R. Ghaderi, W. T., and M. G. Face recognition using error correcting output codes. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001. 6

[13] F. Lazebnik. *On a Regular Simplex in $\mathbb{R}^n$*. From http://www.math.udel.edu/ lazebnik/papers/simplex.pdf, 2006. 2, 4

[14] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005. 5

[15] D. Lin and X. Tang. Recognize high resolution faces: From macrocosm to microcosm. In *Proc. of CVPR'06*, 2006. 1

[16] C. Liu and H. Wechsler. Enhanced fisher linear discriminant moldels for face recognition. In *Proc. of ICPR'98*, 1998. 1

[17] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using LDA-based algorithms. *IEEE Trans. on Neural Networks*, 14(1):195–200, 2003. 1

[18] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. http://matrixcookbook.com/, 2006. 3

[19] M. Plutowski. *Survey: Cross-validation in Theory and in Practice*. Research Report. Dept. of Computational Science Reserach, David Sarnoff Reserach Center, Princeton, New Jersey., 1996. 3, 4

[20] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 1993. 5

[21] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proc. Of the 15th International Conference on Machine Learning (ICML98)*, pages 515–521. Madison-Wisconsin, 1998. 2, 3

[22] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 215, Washington, DC, USA, May 2002. IEEE Computer Society. 5

[23] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 25(12):1615–1618, 2003. 5

[24] M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conf. Computer Vision and Pattern Recognition*, 1991. 1

[25] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. In *Proc. of CVPR'04*, 2004. 1

[26] X. Wang and X. Tang. Random sampling LDA for face recognition. In *Proc. of CVPR'04*, 2004. 1

[27] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 26(9):1222–1227, 2004. 1

[28] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *Proc. of FGR'98*, 1998. 1

Appendix: Derivation of (20)

We only prove the case $k = l$. For other cases $k < l$, one can always permute the order of the training patterns so that the kth group moves to be the last one. Denote

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix} \tag{22}$$

where $K_{11} \in \mathbb{R}^{(n-n_l)\times(n-n_l)}$, $K_{12} \in \mathbb{R}^{(n-n_l)\times n_l}$, $K_{22} \in \mathbb{R}^{n_l \times n_l}$.

To train KRR after leaving the lth group out, one needs to solve the following linear system

$$(K_{11} + \lambda I_{n-n_l})\hat{A} = Y_{\setminus l} \tag{23}$$

where $Y_{\setminus l}$ equals Y by deleting $Y_{(l)}$. Then, the labels of the validation patterns are

$$Y_{cv}^{(l)} = K_{12}^T (K_{11} + \lambda I_{n-n_l})^{-1} Y_{\setminus l}. \tag{24}$$

Applying the well-known block inverse formula of matrices, one has

$$\begin{bmatrix} K_{11} + \lambda I & K_{12} \\ K_{12}^T & K_{22} + \lambda I \end{bmatrix}^{-1} = \begin{bmatrix} F_{11}^{-1} & -(K_{11} + \lambda I)^{-1} K_{12} F_{22}^{-1} \\ -F_{22}^{-1} K_{12}^T (K_{11} + \lambda I)^{-1} & F_{22}^{-1} \end{bmatrix} \tag{25}$$

where

$$\begin{aligned} F_{11} &= (K_{11} + \lambda I) - K_{12}(K_{22} + \lambda I)^{-1} K_{12}^T \\ F_{22} &= K_{22} + \lambda I - K_{12}^T (K_{11} + \lambda I)^{-1} K_{12}. \end{aligned} \tag{26}$$

Substituting (25) into (17) and noticing the notations (18,19), one has

$$\begin{aligned} A_{(l)} &= F_{22}^{-1} \left(Y_{(l)} - K_{12}^T (K_{11} + \lambda I)^{-1} Y_{\setminus l} \right) \\ &= F_{22}^{-1} \left(Y_{(l)} - Y_{cv}^{(l)} \right). \end{aligned} \tag{27}$$

From (25) and (21), $F_{22}^{-1} = C_{ll}$ and thus (20) is true for $k = l$.

□