

SKFD-ISOMAP FOR FACE RECOGNITION

Ruifan Li¹, Cong Wang¹, and Xuyan Tu²

¹*Beijing University of Posts and Telecommunications, Beijing, China*

²*University of Science and Technology Beijing, Beijing, China*

Abstract: Recently, neuroscientists emphasized the manifold ways of perception and proposed Isomap for manifold learning. Favorable results have been achieved using Isomap for data description and visualization. However, since the unsupervised Isomap is developed based on minimizing the reconstruction error with multidimensional scaling (MDS) without using the class specific information, it may not be optimal from the perspective of pattern classification. Therefore, an improved version of Isomap, namely SKFD-Isomap, is proposed using class information to construct the neighborhood, and kernel Fisher discriminant (KFD) to achieve the nonlinear embedding. A nearest neighbor classifier is then applied in the subspace for classification. Experimental results show the effectiveness of the proposed approach.

Key words: face recognition, manifold, Isomap (isometric feature mapping), KFD

1. INTRODUCTION

Recent years, a growing interest has been shown in subspace methods for face recognition [1-5]. As facial images, represented as high-dimensional pixel arrays, often belong to a subspace of intrinsically low dimension. Eigenfaces [2] and Fisherfaces [3] are two typical examples. Eigenfaces, based on Principle component Analysis (PCA), is an unsupervised learning algorithm. PCA performs dimensionality reduction by projecting the original N -dimensional data onto the d ($d \ll N$)-dimensional linear subspace spanned by the leading eigenvectors of covariance matrix of the original data. For linearly embedding manifolds, PCA is guaranteed to produce a compact

representation for data visualization. However, without using the class information this method may be not effective for pattern classification.

Fisherfaces is based on Fisher Linear Discriminant (FLD). Contrasted to PCA, which finds a projection direction that retains maximum variance, FLD finds the optimal projection direction that maximized the distances between classes and minimized the distances within classes. In general, the FLD-based methods perform better than those of PCA-based [4]. However, Fisherfaces are confronted with the singularity of within-class matrix caused by the small samples against their high dimensions. Moreover, they both search for a linear subspace for further classification, failing to deal with nonlinear changes in face images.

As enhancements for PCA and FLD, kernel Eigenfaces and Fisherfaces [6, 7] showed expressive results in face recognition, in which the samples are first implicitly mapped onto a high-dimensional feature spaces, then PCA or FLD is applied, equivalent to extracting the most discriminant nonlinear features in the original input space. These methods can discover the nonlinear structure in the face images. However, none of them explicitly considers the structure of the manifold on which the face images possibly reside.

Recently, neuroscientists emphasized the manifold ways of perception, and showed the face images may reside on a nonlinear manifold [8]. Some manifold learning methods, i.e. Isomap (or Isometric feature mapping) [9], Locally Linear Embedding (LLE) [10], and Laplacian eigenmaps [11] are proposed. Although these methods have demonstrated excellent results in finding the embedding manifolds that best describe the data points, they are suboptimal from the classification viewpoint.

In this paper, an improved version of Isomap, namely SKFD-Isomap, is proposed for face recognition. SKFD-Isomap utilizes class information to guide the construction of neighborhood graph, and then applies Kernel Fisher Discriminant (KFD) to find an optimal projection direction using geodesic distances as feature vectors. Experimental results demonstrated the effectiveness of SKFD-Isomap for face recognition.

The rest of this paper is organized as follows. In section 2, Isomap is briefly reviewed. In section 3, SKFD-Isomap is proposed. In section 4, experiments are reported. Finally, section 5 ends with some conclusions.

2. ISOMAP

For data lying on a nonlinear manifold, the true distance between two data points is the geodesic distance on the manifold, i.e. the distance along the surface of the manifold, rather than the straight-line Euclidean distance, as shown in Figure 1. Isomap [9] builds on Multi-Dimensional Scaling (MDS)

but seeks to preserve the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The crux is estimating the geodesic distance between faraway points, given only input space distances. For neighboring points, Euclidean distance in input space provides a good approximation to geodesic distance. For faraway points, geodesic distance can be approximated by adding up a sequence of short hops between neighboring points.

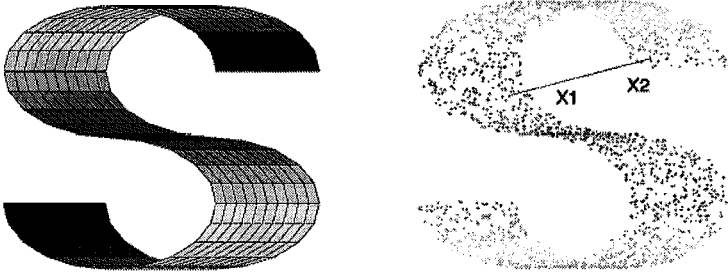


Figure 1. S-curve and geodesic distance

Given a set of n samples, $\{x_i\}_{i=1}^n$, each element $x_i \in \square^N$. Isomap can find an implicit mapping $f: \square^N \rightarrow \square^d$ ($d < N$) from the input space onto the feature space, generating the lower coordinate vector set $\{y_i\}_{i=1}^n$ of $\{x_i\}_{i=1}^n$. The procedures of Isomap are listed as follows:

Step 1, Construct neighborhood graph: First compute Euclidean distance, $d_X(x_i, x_j)$ between any two data points x_i and x_j in the input space. Next, define the graph G over all data points by connecting neighborhood points x_i and x_j , in which the neighborhood of any point can be defined either as the K nearest points or all the points within a fixed radius ε .

Step 2, Compute shortest paths: Initialize the element $d_G(x_i, x_j)$ of graph matrix G as $d_X(x_i, x_j)$ if x_i and x_j are linked by an edge; $d_G(x_i, x_j) = \infty$ otherwise. Then for each $k = 1, \dots, n$ in turn, replace all entries $d_G(x_i, x_j)$ by $\min\{d_G(x_i, x_j), d_G(x_i, x_k) + d_G(x_k, x_j)\}$. The matrix of final values $D_G = \{d_G(x_i, x_j)\}$ will contain the shortest path distances between all pairs of points in G .

Step 3, Construct d -dimensional embedding applying MDS to the matrix of graph distance D_G : Compute $\tau(D_G)$ with τ defined as $-HS/2$, where S is the matrix of squared distance $S_{ij} = D_{ij}^2$, and H is the centering matrix $\{H_{ij} = \delta_{ij} - 1/N\}$. Let λ_p be the p -th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$, and v_p^i be the i -th component of the p -th eigenvector. Then set the p -th component of the d -dimensional coordinate vector y_i equal to $\sqrt{\lambda_p} v_p^i$.

As for classification tasks, Isomap can be viewed as a feature extraction process. However, the mapping between input space and feature space is implicit. It should be learned by some nonlinear interpolation techniques, such as Generalized Regression Neural Network (GRNN) [15] for the new test samples. Moreover, MDS used in Isomap is based on minimizing the reconstruction error, thus, discriminant information cannot be extracted effectively. From this, Yang [12] proposed Ext-Isomap algorithm with FLD replacing the MDS used in the original Isomap for pattern classification. Nevertheless, in Ext-Isomap, when constructing the neighborhood graph no class-label information are used. Therefore, further modifications should be made on Isomap for pattern classification.

3. SKFD-ISOMAP

Consider a set of c disjoint subsets $X = \bigcup_{i=1}^c X_i$ with each element $x \in \mathbb{R}^N$, and each subset has n_i samples, i.e. $\sum_{i=1}^c n_i = n$. In SKFD-Isomap, the first step is to determine the neighbors of each sample x_i on the low-dimensional manifold M based on Euclidean distance metric $d_X(x_i, x_j)$. Whereas, contrasted to the unsupervised nature of Isomap, the SKFD-Isomap utilizes the class information by rescaling the Euclidean distance between two data points with a constant factor λ ($0 < \lambda < 1$) if the class labels are the same. Then, the modified distance matrix between any two data points is utilized to determine whether two points are neighbors or not. These neighborhood relationships are represented in a weighted graph G in which $d_G(x_i, x_j) = d_X(x_i, x_j)$ if x_i and x_j are neighbors, and $d_G(x_i, x_j) = \infty$ otherwise.

The next step is to estimate geodesic distance $d_M(x_i, x_j)$ between any pair of points on the manifold M . For neighboring points, input space distance provides a good approximation to geodesic distance. For faraway points, geodesic distance can be approximated by a sequence of short hops between neighboring data points. In other words, $d_M(x_i, x_j)$ is approximated by the shortest path between x_i and x_j on G , which is computed by $\min\{d_G(x_i, x_j), d_G(x_i, x_k) + d_G(x_k, x_j)\}$, for each $k = 1, \dots, n$. The shortest paths between any two points are represented in a matrix D , where $D_{ij} = d_G(x_i, x_j)$.

Thus, each data point is represented as a feature vector of its geodesic distance to any points. In other words, the feature vector of data point x_i is an n -dimensional vector $f_i = [D_{ij}]$ where $j = 1, \dots, n$ and $D_{ii} = 0$. We then apply KFD on the feature vectors find an optimal projection direction for classification.

In KFD, each vector f_i is projected to a high-dimensional feature space, \mathbb{R}^f , by a nonlinear mapping function, $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^f$, $f > n$. Note that the dimensionality of the image space \mathbb{R}^f can be arbitrarily large. Then, we

formulate the FLD problem in the high-dimensional feature space only using dot products [13].

We assume the projection $\phi(f_i)$ are centered in \square^f , i.e. $\frac{1}{n} \sum_{i=1}^n \phi(f_i) = 0$, which could be implemented as in [14]. The between-class and within-class scatter matrix are defined as

$$S_B^\phi = \sum_{i=1}^c n_i \mu_i^\phi (\mu_i^\phi)^T$$

$$S_w^\phi = \sum_{i=1}^c \sum_{j=1}^{n_i} (\phi(f_j) - \mu_i^\phi)(\phi(f_j) - \mu_i^\phi)^T$$

, where μ_i^ϕ is the mean of class i in \square^f . Applying FLD in feature in kernel space, we need to find eigenvalues and eigenvectors w^ϕ of

$$S_B^\phi w^\phi = \lambda S_w^\phi w^\phi$$

, which can be obtained by

$$W_{OPT}^\phi = \arg \max_{W^\phi} \frac{|(W^\phi)^T S_B^\phi W^\phi|}{|(W^\phi)^T S_w^\phi W^\phi|} = [w_1^\phi, \dots, w_m^\phi]$$

, where $\{w_i^\phi\}_{i=1}^m$ is the set of generalized eigenvectors corresponding to the m largest generalized eigenvalues $\{\lambda_i\}_{i=1}^m$.

From the theory of reproducing kernels, any solution $w^\phi \in \square^f$ must lie in the span of all the mapping of training samples [8], and there exist coefficients α_i such that

$$w^\phi = \sum_{i=1}^n \alpha_i \phi(f_i)$$

We define the kernel function as $k(x, y) = \phi(x) \cdot \phi(y)$, thus the kernel matrix in high-dimensional space can be denoted as K , i.e., $K_{ij} = k(f_i, f_j) = \phi(f_i) \cdot \phi(f_j)$, which is a $n \times n$ symmetric matrix. We can also define a matrix $Z = (Z_i)_{i=1, \dots, c}$, where Z_i is a $n_i \times n_i$ matrix with terms all equal to $1/n_i$, i.e., Z is an $n \times n$ block diagonal matrix. Thus, the KFD problem becomes

$$\lambda K K \alpha = K Z K \alpha$$

We can now project the vectors in \square^f to a lower dimensional space spanned by the eigenvector. Let f be the feature vector of a test sample x , whose projection is $\phi(f)$ in \square^f , then the projection of \square^f onto the eigenvectors as follows:

$$w^\phi \cdot \phi(f) = \sum_{i=1}^n \alpha_i k(f_i, f)$$

The above procedures can learn a subspace for further classification. To summarize, the SKFD-Isomap for classification has five steps as follows: S1, Compute original distances matrix between data points, and rescaling them with class information, i.e., $d_X(x_i, x_j) = \lambda d_X(x_i, x_j)$. S2, Compute shortest paths on the manifold, and feature can be denoted as f_i . S3, Compute projection

direction with KFD applied to the matrix of shortest paths. S4, For new test sample use GRNN to approximate its mapping from input space onto the feature space. S5, Map the feature vector to kernel space and then predict its class-label using simple classifier, such as K nearest neighbor.

4. EXPERIMENTS

In this section, SKFD-Isomap is compared with kernel Fisherfaces for face recognition using the Yale face database. In those two methods, the polynomial kernel functions with second or third order are selected. The parameters for most of the methods are determined empirically to achieve the best results. For SFKD-Isomap, different values of λ between 0.1 and 0.5 are tested. For kernel Fisherfaces, the samples are projected to a subspace spanned by the $c - 1$ largest eigenvectors.

This Yale database contains 165 images, each size 320 by 243, of 15 subjects in a variety of conditions including with and without glasses, illumination variation, and changes in facial expression. Since the database has large background, to reduce the adverse effect, original images have been cropped without containing the facial contours with size 87 by 123. Furthermore, each image has been resized to 58 by 82 for computational efficiency. Figure 2 shows closely cropped images of a few subjects.



Figure 2. Some chopped face images from database

In this experiment, each image is represented as a raster scan vector of intensity values, and then normalized to be zero-mean unit-variance vectors. The training set and test set are selected randomly. When performing the experiment, for each subject, eight images are selected for training, and the other two samples are left as test samples. All the images are then projected to a reduced space and recognition is performed using a nearest neighbor classifier. To reduce the fluctuation among the results as possible, the experiments are

performed 400 times. The best experimental results with the order of kernel functions $d = 2$, $K = 40$, and $\lambda = 0.3$ are given in table 1.

Table 1. Performance comparison on the Yale Database

| Method | Parameters | Dimension of Space | Error Rate (%) |
|--------------------|--------------------------------|--------------------|----------------|
| Kernel Fisherfaces | $d = 2$ | 14 | 12.37 |
| SKFD-Isomap | $d = 2, K = 40, \lambda = 0.3$ | 14 | 8.62 |

In addition, we test the performance of SKFD-Isomap with different values of d and with different order of polynomial kernel function, i.e. $d = 2$ or 3 . Figure 3(a) shows the results, where $K = 40$. The order of polynomial kernel, $d = 2$ performs better than higher order. This phenomena may be caused by overfitting with higher order. For the neighborhood parameter, K , its performance impact on SKFD-Isomap is also tested. The results are shown in figure 3(b), where $d = 2$ and $\lambda = 0.3$. We can see that the proposed approach shows better performance compared with that of Ext-Isomap. We suppose that the class information used in SKFD-Isomap improved the feature extraction for further classification.

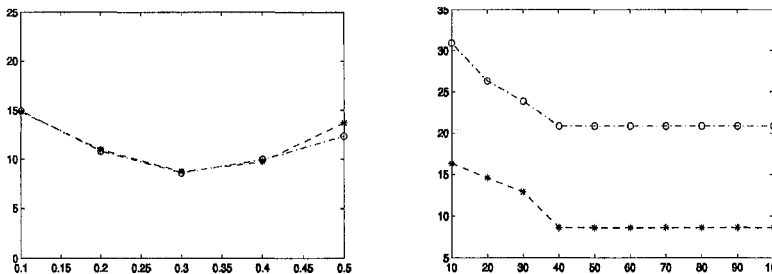


Figure 3. a) SKFD-Isomap with $d = 2$ or 3 ; b) performance of Ext-Isomap and SKFD-Isomap

5. CONCLUSION

In this paper, an improved version of Isomap, namely SKFD-Isomap, is proposed for face recognition. The SKFD-Isomap utilizes class information to guide the construction of the distances matrix between any two data points. Moreover, the geodesic distances between all pairs of points are applied as feature vectors replacing MDS with KFD to find an optimal projection direction for classification. In face recognition experiments, SKFD-Isomap serves as a feature extraction process compared with Ext-Isomap and Kernel Fisherfaces, combined with a nearest neighbor classifier. Experimental results show that SKFD-Isomap excels kernel Fisherfaces and Ext-Isomap.

ACKNOWLEDGEMENTS

The authors would like to thank AT&T laboratory Cambridge and Center for Computational Vision and Control at Yale University for their devotion of face database for public research.

REFERENCES

- [1] Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: Face recognition: A literature survey. *ACM Computing Surveys*, 35 (2003), 399–458
- [2] Turk, M.A., Pentland, A.P.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3 (1991), 72–86
- [3] Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (1997), 711–720
- [4] Martinez, Alex M.; Kak, A.C.: Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (2001), 228–233
- [5] He, X., Yan, S., Hu, Y., Zhang, H.: Learning a locality preserving subspace for visual recognition. In: *Proc. IEEE Ninth International Conference on Computer Vision (ICCV03)*, Nice, France, (2003), 385–392
- [6] Yang, M.H.: Kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods. In: *Proc. IEEE Sixth International Conference on Automatic Face and Gesture recognition (FGR02)*, Washington D.C. (2002), 215–220
- [7] Liu, Q., Huang, R., Lu, H., Ma, S.: Kernel-based nonlinear discriminant analysis for face recognition. In: *Proc. IEEE Sixth International Conference on Automatic Face and Gesture recognition (FGR02)*, Washington D.C. (2002), 788–795
- [8] Seung, H.S., Lee, D.D.: The manifold ways of perception. *Science*, 290 (2000), 2268–2269
- [9] Silva, V., Tenenbaum, J., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. *Science*, 290 (2000), 2219–2223
- [10] Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (2000), 2223–2226
- [11] Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, (2001), 788–795
- [12] Yang, M.H.: Face recognition using extended isomap. In: *Proc. IEEE International Conference on Image Processing*, Rochester, NY, (2002), 117–120
- [13] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Muller, K.: Fisher discriminant analysis with kernels. In: *Proc. IEEE Neural Networks for Signal Processing Workshop*, Madison, Wisconsin, (1999), 41–48
- [14] Scholkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10 (1998), 1299–1319
- [15] Specht, D.F.: A general regression neural network. *IEEE Transactions on Neural network*, 2 (1991), 568–576