Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2013, Article ID 106867, 14 pages
http://dx.doi.org/10.1155/2013/106867

*Research Article*

# Fast Discriminative Stochastic Neighbor Embedding Analysis

**Jianwei Zheng, Hong Qiu, Xinli Xu, Wanliang Wang, and Qiongfang Huang**

*School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China*

Correspondence should be addressed to Jianwei Zheng; zjw@zjut.edu.cn

Received 9 February 2013; Accepted 22 March 2013

Academic Editor: Carlo Cattani



Feature is important for many applications in biomedical signal analysis and living system analysis. A fast discriminative stochastic neighbor embedding analysis (FDSNE) method for feature extraction is proposed in this paper by improving the existing DSNE method. The proposed algorithm adopts an alternative probability distribution model constructed based on its $K$-nearest neighbors from the interclass and intraclass samples. Furthermore, FDSNE is extended to nonlinear scenarios using the kernel trick and then kernel-based methods, that is, KFDSNE1 and KFDSNE2. FDSNE, KFDSNE1, and KFDSNE2 are evaluated in three aspects: visualization, recognition, and elapsed time. Experimental results on several datasets show that, compared with DSNE and MSNP, the proposed algorithm not only significantly enhances the computational efficiency but also obtains higher classification accuracy.

## 1. Introduction

In recent years, dimensional reduction which can reduce the curse of dimensionality [1] and remove irrelevant attributes in high-dimensional space plays an increasingly important role in many areas. It promotes the classification, visualization, and compression of the high dimensional data. In machine learning, dimension reduction is used to reduce the dimension by mapping the samples from the high-dimensional space to the low-dimensional space. There are many purposes of studying it: firstly, to reduce the amount of storage, secondly, to remove the influence of noise, thirdly, to understand data distribution easily, and last but not least, to achieve good results in classification or clustering.

Currently, many dimensional reduction methods have been proposed, and they can be classified variously from different perspectives. Based on the nature of the input data, they are broadly categorized into two classes: linear subspace methods which try to find a linear subspace as feature space so as to preserve certain kind of characteristics of observed data, and nonlinear approaches such as kernel-based techniques and geometry-based techniques; from the class labels' perspective, they are divided into supervised learning and unsupervised learning; furthermore, the purpose of the former is to maximize the recognition rate between classes while the latter is for making the minimum of information loss. In addition, judging whether samples utilize local information or global information, we divide them into local method and global method.

We briefly introduce several existing dimensional reduction techniques. In the main linear techniques, principal component analysis (PCA) [2] aims at maximizing the variance of the samples in the low-dimensional representation with a linear mapping matrix. It is global and unsupervised. Different from PCA, linear discriminant analysis (LDA) [3] learns a linear projection with the assistance of class labels. It computes the linear transformation by maximizing the amount of interclass variance relative to the amount of intraclass variance. Based on LDA, marginal fisher analysis (MFA) [4], local fisher discriminant analysis (LFDA) [5], and maxmin distance analysis (MMDA) [6] are proposed. All of the three are linear supervised dimensional reduction methods. MFA utilizes the intrinsic graph to characterize the intraclass compactness and uses meanwhile the penalty graph to characterize interclass separability. LFDA introduces the locality to the LFD algorithm and is particularly useful for samples consisting of intraclass separate clusters. MMDA considers maximizing the minimum pairwise samples of interclass.

To deal with nonlinear structural data, which can often be found in biomedical applications [7–10], a number of nonlinear approaches have been developed for dimensional reduction. Among these kernel-based techniques and geometry-based techniques are two hot issues. Kernel-based techniques

attempt to obtain the linear structure of nonlinearly distributed data by mapping the original inputs to a high-dimensional feature space. For instance, kernel principal component analysis (kernel PCA) [11] is the extension of PCA using kernel tricks. Geometry-based techniques, in general, are known as manifold learning techniques such as isometric mapping (ISOMAP) [12], locally linear embedding (LLE) [13], Laplacian eigenmap (LE) [14], Hessian LLE (HLLE) [15], and local tangent space alignment (LTSA) [16]. ISOMAP is used for manifold learning by computing the pairwise geodesic distances for input samples and extending multidimensional scaling. LLE exploits the linear reconstructions to discover nonlinear structure in high-dimensional space. LE first constructs an undirected weighted graph, and then recovers the structure of manifold by graph manipulation. HLLE is based on sparse matrix techniques. As for LTSA, it begins by computing the tangent space at every point and then optimizes to find an embedding that aligns the tangent spaces.

Recently, stochastic neighbor embedding (SNE) [17] and extensions thereof have become popular for feature extraction. The basic principle of SNE is to convert pairwise Euclidean distances into probabilities of selecting neighbors to model pairwise similarities. As extension of SNE, $t$-SNE [18] uses Student's $t$-distribution to model pairwise dissimilarities in low-dimensional space and it alleviates the optimization problems and the crowding problem of SNE by the methods below: (1) it uses a symmetrized version of the SNE cost function with simpler gradients that was briefly introduced by Cook et al. [19], and (2) it employs a heavy-tailed distribution in the low-dimensional space. Subsequently, Yang et al. [20] systematically analyze the characteristics of the heavy-tailed distribution and the solutions to crowding problem. More recently, Wu et al. [21] explored how to measure similarity on manifold more accurately and proposed a projection approach called manifold-oriented stochastic neighbor projection (MSNP) for feature extraction based on SNE and $t$-SNE. MSNP employs Cauchy distribution rather than standard Student's $t$-distribution used in $t$-SNE. In addition, for the purpose of learning the similarity on manifold with high accuracy, MSNP uses geodesic distance for characterizing data similarity. Though MSNP has many advantages in terms of feature extraction, there is still a drawback in it: MSNP is an unsupervised method and lacks the idea of class label, so it is not suitable for pattern identification. To overcome the disadvantage of MSNP, we have done some preliminary work and presented a method called discriminative stochastic neighbor embedding analysis (DSNE) [22]. DSNE effectively resolves the problems above, but since it selects all the training samples as their reference points, it has high computational cost and is thus computationally infeasible for the large-scale classification tasks with high-dimensional features [23, 24]. On the basis of our previous research, we present a method called fast discriminative stochastic neighbor embedding analysis (FDSNE) to overcome the disadvantages of DSNE in this paper.

The rest of this paper is organized as follows: in Section 2, we introduce in detail the proposed FDSNE and briefly compare it with MSNP and DSNE in Section 3. Section 4 gives the nonlinear extension of FDSNE. Furthermore, experiments on various databases are presented in Section 5. Finally, Section 6 concludes this paper and several issues for future works are described.

## 2. Fast Discriminative Stochastic Neighbor Embedding Analysis

Consider a labeled data samples matrix as

$$\mathbf{X} = \left\{\mathbf{x}_1^1, \ldots, \mathbf{x}_{N_1}^1, \mathbf{x}_1^2, \ldots, \mathbf{x}_{N_2}^2, \ldots, \mathbf{x}_1^C, \ldots, \mathbf{x}_{N_C}^C\right\}, \quad (1)$$

where $\mathbf{x}_i^c \in R^d$ is a $d$-dimensional sample and means the $i$th sample in the $c$th class. $C$ is the number of sample classes, $N_c$ is the number of samples in the $c$th class, and $N = N_1 + N_2 + \cdots + N_C$.

In fact, the basic principle of FDSNE is the same as $t$-SNE which is to convert pairwise Euclidean distances into probabilities of selecting neighbors to model pairwise similarities [18]. Since the DSNE selects all the training samples as its reference points, it has high computational cost and is thus computationally infeasible for the large-scale classification tasks with high-dimensional features. So according to the KNN classification rule, we propose an alternative probability distribution function which makes the label of target sample determined by its first $K$-nearest neighbors in FDSNE. In this paper, $\mathrm{NH}_l(\mathbf{x}_i)$ and $\mathrm{NM}_l(\mathbf{x}_i)$ are defined. They, respectively, denote the $l$th-nearest neighbor of $\mathbf{x}_i$ from the same class and the different classes in the transformed space. Mathematically, the joint probability $p_{ij}$ is given by

$$p_{ij} = \begin{cases} \dfrac{\exp\left(-d_{ij}^2/2\lambda^2\right)}{\sum_{t\in H_m}\exp\left(-d_{mt}^2/2\lambda^2\right)} & \forall j \in H_i \\ \dfrac{\exp\left(-d_{ij}^2/2\lambda^2\right)}{\sum_{t\in M_m}\exp\left(-d_{mt}^2/2\lambda^2\right)} & \forall j \in M_i \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

In formula (2), $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}$ is the Euclidian distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, the parameter $\lambda$ is the variance parameter of Gaussian which determines the value of $p_{ij}$, $H_i = \{j \mid 1 \le j \le N, 1 \le i \le N, \mathbf{x}_j = \mathrm{NH}_k(\mathbf{x}_i) \text{ and } 1 \le k \le K_1\}$, $H_m = \{t \mid 1 \le t \le N, 1 \le m \le N, \mathbf{x}_t = \mathrm{NH}_k(\mathbf{x}_m) \text{ and } 1 \le k \le K_1\}$, $M_i = \{j \mid 1 \le j \le N, 1 \le i \le N, \mathbf{x}_j = \mathrm{NH}_k(\mathbf{x}_i) \text{ and } 1 \le k \le K_2\}$, and $M_m = \{t \mid 1 \le t \le N, 1 \le m \le N, \mathbf{x}_t = \mathrm{NH}_k(\mathbf{x}_m) \text{ and } 1 \le k \le K_2\}$, and then the denominator in formula (2) means all of the reference points under selection from the same class or the different classes. In particular, the joint probability $p_{ij}$ not only keeps symmetrical characteristics of the probability distribution matrix but also makes the probability value of interclass data to be 1 and the same for intraclass data.

For low-dimensional representations, FDSNE uses counterparts $\mathbf{y}_i$ and $\mathbf{y}_j$ of the high-dimensional datapoints $\mathbf{x}_i$ and

$\mathbf{x}_j$. It is possible to compute a similar joint probability via the following expression:

$$q_{ij} = \begin{cases} \dfrac{\left(1 + d_{ij}^2(\mathbf{A})\right)^{-1}}{\sum_{t \in H_m} \left(1 + d_{mt}^2(\mathbf{A})\right)^{-1}} & \forall j \in H_i \\ \dfrac{\left(1 + d_{ij}^2(\mathbf{A})\right)^{-1}}{\sum_{t \in M_m} \left(1 + d_{mt}^2(\mathbf{A})\right)^{-1}} & \forall j \in M_i \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In what follows, we introduce the transformation by a linear projection: $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$ ($\mathbf{A} \in \mathbf{R}^{r \times d}$) so that $d_{ij}(\mathbf{A}) = \|\mathbf{y}_i - \mathbf{y}_j\| = \|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)}$. Then by simple algebra formulation, formula (3) has the following equivalent expression:

$$q_{ij} = \begin{cases} \dfrac{\left(1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)\right)^{-1}}{\sum_{t \in H_m} \left(1 + \left(\mathbf{x}_m - \mathbf{x}_t\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_m - \mathbf{x}_t\right)\right)^{-1}} & \forall j \in H_i \\ \dfrac{\left(1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)\right)^{-1}}{\sum_{t \in M_m} \left(1 + \left(\mathbf{x}_m - \mathbf{x}_t\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_m - \mathbf{x}_t\right)\right)^{-1}} & \forall j \in M_i \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Note that all data have the intrinsic geometry distribution and there is no exception for intraclass samples and interclass samples. Then the same distribution is required to hold in feature space. Since the Kullback-Leiber divergence [25] is wildly used to quantify the proximity of two probability distributions, we choose it to build our penalty function here. Based on the above definition, the function can be formulated as:

$$\min C(\mathbf{A}) = \sum_{\forall j \in H_i} p_{ij} \log \frac{p_{ij}}{q_{ij}} + \sum_{\forall j \in M_i} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{5}$$

In this work, we use the conjugate gradient method to minimize $C(\mathbf{A})$. In order to make the derivation less cluttered, we first define four auxiliary variables $w_{ij}$, $u_{ij}$, $u_{ij}^H$, and $u_{ij}^M$ as:

$$\begin{aligned} w_{ij} &= \left[1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)\right]^{-1}, \\ u_{ij} &= \left(p_{ij} - q_{ij}\right) w_{ij}, \\ u_{ij}^H &= \begin{cases} u_{ij} & \forall j \in H_i \\ 0 & \text{otherwise,} \end{cases} \\ u_{ij}^M &= \begin{cases} u_{ij} & \forall j \in M_i \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

Then differentiating $C(\mathbf{A})$ with respect to the transformation matrix $\mathbf{A}$ gives the following gradient, which we adopt for learning:

$$\begin{aligned} &\frac{dC(\mathbf{A})}{d(\mathbf{A})} \\ &= \sum_{\forall j \in H_i} \frac{p_{ij}}{q_{ij}} \left(q_{ij}\right)' + \sum_{\forall j \in M_i} \frac{p_{ij}}{q_{ij}} \left(q_{ij}\right)' \\ &= 2\mathbf{A} \left[ \sum_{\forall j \in H_i} p_{ij} \frac{\left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T}{1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)} \right] \\ &\quad - 2\mathbf{A} \left[ \sum_{\forall j \in H_i} p_{ij} \left( \sum_{t \in H_m} \left(1 + \left(\mathbf{x}_m - \mathbf{x}_t\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_m - \mathbf{x}_t\right)\right)^{-2} \right. \right. \\ &\qquad \left. \times \left(\mathbf{x}_m - \mathbf{x}_t\right)\left(\mathbf{x}_m - \mathbf{x}_t\right)^T \right) \\ &\qquad \times \left( \sum_{t \in H_m} \left(1 + \left(\mathbf{x}_m - \mathbf{x}_t\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_m - \mathbf{x}_t\right)\right)^{-1} \right)^{-1} \Bigg] \\ &\quad + 2\mathbf{A} \left[ \sum_{\forall j \in M_i} p_{ij} \frac{\left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T}{1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)} \right] \\ &\quad - 2\mathbf{A} \left[ \sum_{\forall j \in M_i} p_{ij} \left( \sum_{t \in M_m} \left(1 + \left(\mathbf{x}_m - \mathbf{x}_t\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_m - \mathbf{x}_t\right)\right)^{-2} \right. \right. \\ &\qquad \left. \times \left(\mathbf{x}_m - \mathbf{x}_t\right)\left(\mathbf{x}_m - \mathbf{x}_t\right)^T \right) \\ &\qquad \times \left( \sum_{t \in M_m} \left(1 + \left(\mathbf{x}_m - \mathbf{x}_t\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_m - \mathbf{x}_t\right)\right)^{-1} \right)^{-1} \Bigg] \\ &= 2\mathbf{A} \left[ \sum_{\forall j \in H_i} p_{ij} w_{ij} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \right. \\ &\qquad \left. - \sum_{t \in H_m} q_{mt} w_{mt} \left(\mathbf{x}_m - \mathbf{x}_t\right)\left(\mathbf{x}_m - \mathbf{x}_t\right)^T \right] \\ &\quad + 2\mathbf{A} \left[ \sum_{\forall j \in M_i} p_{ij} w_{ij} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \right. \\ &\qquad \left. - \sum_{t \in M_m} q_{mt} w_{mt} \left(\mathbf{x}_m - \mathbf{x}_t\right)\left(\mathbf{x}_m - \mathbf{x}_t\right)^T \right] \\ &= 2\mathbf{A} \left[ \sum_{\forall j \in H_i} u_{ij} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \right. \\ &\qquad \left. + \sum_{\forall j \in M_i} u_{ij} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \right]. \end{aligned} \tag{7}$$

Let $\mathbf{U}^H$ be the $N$ order matrix with element $u_{ij}^H$, and let $\mathbf{U}^M$ be the $N$ order matrix with element $u_{ij}^M$. Note that $\mathbf{U}^H$ and $\mathbf{U}^M$ are symmetric matrices; therefore, $\mathbf{D}^H$ can be defined as a diagonal matrix that each entry is column (or row) sum of $\mathbf{U}^H$ and the same for $\mathbf{D}^M$, that is, $\mathbf{D}_{ii}^H = \sum_j \mathbf{U}_{ij}^H$ and $\mathbf{D}_{ii}^M = \sum_j \mathbf{U}_{ij}^M$. With this definition, the gradient expression (7) can be reduced to

$$\begin{aligned}
\frac{dC(\mathbf{A})}{d(\mathbf{A})} &= 2\mathbf{A}\left\{\sum_{\forall j\in H_i} u_{ij}\left(\mathbf{x}_i-\mathbf{x}_j\right)\left(\mathbf{x}_i-\mathbf{x}_j\right)^T \right.\\
&\qquad \left. +\sum_{\forall j\in M_i} u_{ij}\left(\mathbf{x}_i-\mathbf{x}_j\right)\left(\mathbf{x}_i-\mathbf{x}_j\right)^T\right\}\\
&= 2\mathbf{A}\left\{\left(\sum_{\forall j\in H_i} u_{ij}\mathbf{x}_i\mathbf{x}_i^T+\sum_{\forall j\in H_i} u_{ij}\mathbf{x}_j\mathbf{x}_j^T\right.\right.\\
&\qquad \left.-\sum_{\forall j\in H_i} u_{ij}\mathbf{x}_i\mathbf{x}_j^T-\sum_{\forall j\in H_i} u_{ij}\mathbf{x}_j\mathbf{x}_i^T\right)\\
&\qquad +\left(\sum_{\forall j\in M_i} u_{ij}\mathbf{x}_i\mathbf{x}_i^T+\sum_{\forall j\in M_i} u_{ij}\mathbf{x}_j\mathbf{x}_j^T\right.\\
&\qquad \left.\left.-\sum_{\forall j\in M_i} u_{ij}\mathbf{x}_i\mathbf{x}_j^T-\sum_{\forall j\in M_i} u_{ij}\mathbf{x}_j\mathbf{x}_i^T\right)\right\}\\
&= 4\mathbf{A}\left\{\left(\mathbf{X}\mathbf{D}^H\mathbf{X}^T-\mathbf{X}\mathbf{U}^H\mathbf{X}^T\right)\right.\\
&\qquad \left.+\left(\mathbf{X}\mathbf{D}^M\mathbf{X}^T-\mathbf{X}\mathbf{U}^M\mathbf{X}^T\right)\right\}\\
&= 4\mathbf{A}\left\{\mathbf{X}\left(\mathbf{D}^H-\mathbf{U}^H+\mathbf{D}^M-\mathbf{U}^M\right)\mathbf{X}^T\right\}.
\end{aligned} \tag{8}$$

Once the gradient is calculated, our optimal problem (5) can be solved by an iterative procedure based on the conjugate gradient method. The description of FDSNE algorithm can be given by the following.

*Step 1.* Collect the sample matrix $\mathbf{X}$ with class labels, and set $K$-nearest neighborhood parameter $K_1$, $K_2$, the variance parameter $\lambda$, and the maximum iteration times $Mt$.

*Step 2.* Compute the pairwise Euclidian distance for $\mathbf{X}$ and compute the joint probability $p_{ij}$ by utilizing formula (2) and class labels.

*Step 3* (set $t = 1 : Mt$). We search for the solution in loop: firstly, compute the joint probability $q_{ij}$ by utilizing formula (4); then, compute gradient $dC(\mathbf{A})/d(\mathbf{A})$ by utilizing formula (8); finally, update $\mathbf{A}^t$ based on $\mathbf{A}^{t-1}$ by conjugate gradient operation.

*Step 4.* Judge whether $C^t - C^{t-1} < \varepsilon$ (in this paper, we take $\varepsilon = 1e-7$) converges to a stable solution or $t$ reaches the maximum value $Mt$. If these prerequisites are met, Step 5 is performed; otherwise, we repeat Step 3.

*Step 5.* Output $\mathbf{A} = \mathbf{A}^t$.

Hereafter, we call the proposed method as fast discriminative stochastic neighbor embedding analysis (FDSNE).

## 3. Comparison with MSNP and DSNE

MSNP is derived from SNE and $t$-SNE, and it is a linear method and has nice properties, such as sensitivity to nonlinear manifold structure and convenience for feature extraction. Since the structure of MSNP is closer to that of FDSNE, we briefly compare FDSNE with MSNP and DSNE in this section.

FDSNE, MSNP, and DSNE use different probability distributions to determine the reference points. The difference can be explained in the following aspects.

Firstly, MSNP learns the similarity relationship of the high-dimensional samples by estimating neighborhood distribution based on geodesic distance metric, and the same distribution is required in feature space. Then the linear projection matrix $\mathbf{A}$ is used to discover the underlying structure of data manifold which is nonlinear. Finally, the Kullback-Leibler divergence objective function is used to keep pairwise similarities in feature space. So the probability distribution function of MSNP and its gradient used for learning are respectively given by

$$\begin{aligned}
p_{ij} &= \frac{\exp\left(-D_{ij}^{\text{geo}}/2\right)}{\sum_{k\neq i}\exp\left(-D_{ik}^{\text{geo}}/2\right)},\\
q_{ij} &= \frac{\left[\gamma^2+\left(\mathbf{x}_i-\mathbf{x}_j\right)^T\mathbf{A}^T\mathbf{A}\left(\mathbf{x}_i-\mathbf{x}_j\right)\right]^{-1}}{\sum_{k\neq l}\left[\gamma^2+\left(\mathbf{x}_k-\mathbf{x}_l\right)^T\mathbf{A}^T\mathbf{A}(\mathbf{x}_k-\mathbf{x}_l)\right]^{-1}},\\
\min C(\mathbf{A}) &= \sum_{i,j} p_{ij}\log\frac{p_{ij}}{q_{ij}},
\end{aligned} \tag{9}$$

where $D_{ij}^{\text{geo}}$ is the geodesic distance for $\mathbf{x}_i$ and $\mathbf{x}_j$ and $\gamma$ is the freedom degree parameter of Cauchy distribution.

DSNE selects the joint probability to model the pairwise similarities of input samples with class labels. It also introduces the linear projection matrix $\mathbf{A}$ as MSNP. The cost function is constructed to minimize the intraclass Kullback-Leibler divergence as well as to maximize the interclass KL divergences. Its probability distribution function and gradient are, respectively, given as by

$$p_{ij} = \begin{cases} \dfrac{\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/2\lambda^2\right)}{\sum_{c_k=c_l}\exp\left(-\left\|\mathbf{x}_k-\mathbf{x}_l\right\|^2/2\lambda^2\right)} & \text{if } c_i=c_j\\ \dfrac{\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/2\lambda^2\right)}{\sum_{c_k\neq c_m}\exp\left(-\left\|\mathbf{x}_k-\mathbf{x}_m\right\|^2/2\lambda^2\right)} & \text{else} \end{cases}$$

$$q_{ij} = \begin{cases} \dfrac{\left(1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)\right)^{-1}}{\sum_{c_k = c_l} \left(1 + \left(\mathbf{x}_k - \mathbf{x}_l\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_k - \mathbf{x}_l\right)\right)^{-1}} & \text{if } c_i = c_j \\ \dfrac{\left(1 + \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_i - \mathbf{x}_j\right)\right)^{-1}}{\sum_{c_k \neq c_m} \left(1 + \left(\mathbf{x}_k - \mathbf{x}_m\right)^T \mathbf{A}^T \mathbf{A} \left(\mathbf{x}_k - \mathbf{x}_m\right)\right)^{-1}} & \text{else,} \end{cases}$$

$$\min C(\mathbf{A}) = \sum_{c_i = c_j} p_{ij} \log \frac{p_{ij}}{q_{ij}} + \sum_{c_i \neq c_k} p_{ik} \log \frac{p_{ik}}{q_{ik}}. \tag{10}$$

Note that on the basis of the DSNE, FDSNE makes full use of class label which not only keeps symmetrical characteristics of the probability distribution matrix but also makes the probability value of interclass data and intraclass data to be 1, and it can effectively overcome large interclass confusion degree in the projected subspace.

Secondly, it is obvious that the selection of reference point in MSNP or DSNE is related to all training samples, while FDSNE only uses the first $K$-nearest neighbors of each sample from all classes. In other words, we propose an alternative probability distribution function to determine whether $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its reference point or not. Actually, the computation of gradient during the optimization process mainly determines the computational cost of MSNP and DSNE. So their computational complexity can be written as $O(2rNd+N^2d)$ in each iteration. Similarly, the computational complexity of FDSNE is $O(2rNd + KNd)$ in each iteration, where $K = K_1 + K_2$. It is obvious that $K \ll N$. Therefore, FDSNE is faster than MSNP and DSNE during each iteration.

## 4. Kernel FDSNE

As a bridge from linear to nonlinear, kernel method emerged in the early beginning of the 20th century and its applications in pattern recognition can be traced back to 1964. In recent years, kernel method has attracted wide attention and numerous researchers have proposed various theories and approaches based on it.

The principle of kernel method is a mapping of the data from the input space $R^d$ to a high-dimensional space $F$, which we will refer to as the *feature space*, by nonlinear function. Data processing is then performed in the feature space, and this can be expressed solely in terms of inner product in the feature space. Hence, the nonlinear mapping need not be explicitly constructed but can be specified by defining the form of the inner product in terms of a Mercer kernel function $\kappa$.

Obviously, FDSNE is a linear feature dimensionality reduction algorithm. So the remainder of this section is devoted to extend FDSNE to a nonlinear scenario using techniques of kernel methods. Let

$$\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\langle \varphi\left(\mathbf{x}_i\right), \varphi\left(\mathbf{x}_j\right) \right\rangle \tag{11}$$

which allows us to compute the value of the inner product in $F$ without having to carry out the map.

It should be noted that we use $\varphi_i$ to denote $\varphi(\mathbf{x}_i)$ for brevity in the following. Next, we express the transformation $\mathbf{A}$ with

$$\mathbf{A} = \left[\sum_{i=1}^{N} b_i^{(1)} \varphi_i, \ldots, \sum_{i=1}^{N} b_i^{(r)} \varphi_i\right]^T. \tag{12}$$

We define $\mathbf{B} = [b^{(1)}, \ldots, b^{(r)}]^T$ and $\mathbf{\Phi} = [\varphi_1, \ldots, \varphi_N]^T$, and then $\mathbf{A} = \mathbf{B}\mathbf{\Phi}$. Based on above definition, the Euclidian distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the $F$ space is

$$\begin{aligned} d_{ij}^F(\mathbf{A}) &= \left\|\mathbf{A}\left(\varphi_i - \varphi_j\right)\right\| = \left\|\mathbf{B}\mathbf{\Phi}\left(\varphi_i - \varphi_j\right)\right\| \\ &= \left\|\mathbf{B}\left(K_i - K_j\right)\right\| = \sqrt{\left(K_i - K_j\right)^T \mathbf{B}^T \mathbf{B} \left(K_i - K_j\right)}, \end{aligned} \tag{13}$$

where $K_i = [\kappa(\mathbf{x}_1, \mathbf{x}_i), \ldots, \kappa(\mathbf{x}_N, \mathbf{x}_i)]^T$ is a column vector. It is clear that the distance in the kernel embedding space is related to the kernel function and the matrix $\mathbf{B}$.

In this section, we propose two methods to construct the objective function. The first strategy makes $\mathbf{B}$ parameterize the objective function. Firstly, we replace $d_{ij}(\mathbf{A})$ with $d_{ij}^F(\mathbf{A})$ in formula (3) so that $p_{ij}^1$, $q_{ij}^1$ which are defined to be applied in the high dimensional space $F$ can be written as

$$p_{ij}^1 = \begin{cases} \dfrac{\exp\left(-\left(K_{ii} + K_{jj} - 2K_{ij}\right)/2\lambda^2\right)}{\sum_{t \in H_m} \exp\left(-\left(K_{mm} + K_{tt} - 2K_{mt}\right)/2\lambda^2\right)} & \forall j \in H_i \\ \dfrac{\exp\left(-\left(K_{ii} + K_{jj} - 2K_{ij}\right)/2\lambda^2\right)}{\sum_{t \in M_m} \exp\left(-\left(K_{mm} + K_{tt} - 2K_{mt}\right)/2\lambda^2\right)} & \forall j \in M_i \\ 0 & \text{otherwise,} \end{cases}$$

$$q_{ij}^1 = \begin{cases} \dfrac{\left(1 + \left(K_i - K_j\right)^T \mathbf{B}^T \mathbf{B} \left(K_i - K_j\right)\right)^{-1}}{\sum_{t \in H_m} \left(1 + \left(K_m - K_t\right)^T \mathbf{B}^T \mathbf{B} \left(K_m - K_t\right)\right)^{-1}} & \forall j \in H_i \\ \dfrac{\left(1 + \left(K_i - K_j\right)^T \mathbf{B}^T \mathbf{B} \left(K_i - K_j\right)\right)^{-1}}{\sum_{t \in M_m} \left(1 + \left(K_m - K_t\right)^T \mathbf{B}^T \mathbf{B} \left(K_m - K_t\right)\right)^{-1}} & \forall j \in M_i \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Then, we denote $C(\mathbf{B})$ by modifying $C(\mathbf{A})$ via substituting $\mathbf{A}$ with $\mathbf{B}$ into the regularization term of formula (5). Finally,


Figure 1: Sample images from COIL-20 dataset.

Figure 2: Samples of the cropped images from USPS dataset.

by the same argument as formula (7), we give the following gradient:

$$\begin{aligned}\frac{dC(\mathbf{B})}{d(\mathbf{B})} &= \sum_{\forall j\in M_i}\frac{p_{ij}^1}{q_{ij}^1}\left(q_{ij}^1\right)' + \sum_{\forall j\in H_i}\frac{p_{ij}^1}{q_{ij}^1}\left(q_{ij}^1\right)' \\ &= 2\mathbf{B}\left[\sum_{\forall j\in H_i}u_{ij}^1\left(K_i-K_j\right)\left(K_i-K_j\right)^T\right. \\ &\quad \left.+\sum_{\forall j\in M_i}u_{ij}^1\left(K_i-K_j\right)\left(K_i-K_j\right)^T\right].\end{aligned} \tag{15}$$

In order to make formula (15) easy to be comprehended, $w_{ij}^1$, $u_{ij}^1$, $u_{ij}^{1H}$, and $u_{ij}^{1M}$ are given by

$$\begin{aligned} w_{ij}^1 &= \left[1+\left(K_i-K_j\right)^T\mathbf{B}^T\mathbf{B}\left(K_i-K_j\right)\right]^{-1}, \\ u_{ij}^1 &= \left(p_{ij}-q_{ij}\right)w_{ij}^1, \\ u_{ij}^{1H} &= \begin{cases} u_{ij}^1 & \forall j\in H_i \\ 0 & \text{otherwise,}\end{cases} \\ u_{ij}^{1M} &= \begin{cases} u_{ij}^1 & \forall j\in M_i \\ 0 & \text{otherwise.}\end{cases}\end{aligned} \tag{16}$$

Meanwhile, the gradient expression (15) can be reduced to

$$\begin{aligned}\frac{dC(\mathbf{B})}{d(\mathbf{B})} &= 2\mathbf{B}\left\{\sum_{\forall j\in H_i}u_{ij}^1\left(K_i-K_j\right)\left(K_i-K_j\right)^T\right. \\ &\quad \left.+\sum_{\forall j\in M_i}u_{ij}^1\left(K_i-K_j\right)\left(K_i-K_j\right)^T\right\}\end{aligned}$$

Figure 3: Sample face images from ORL dataset.

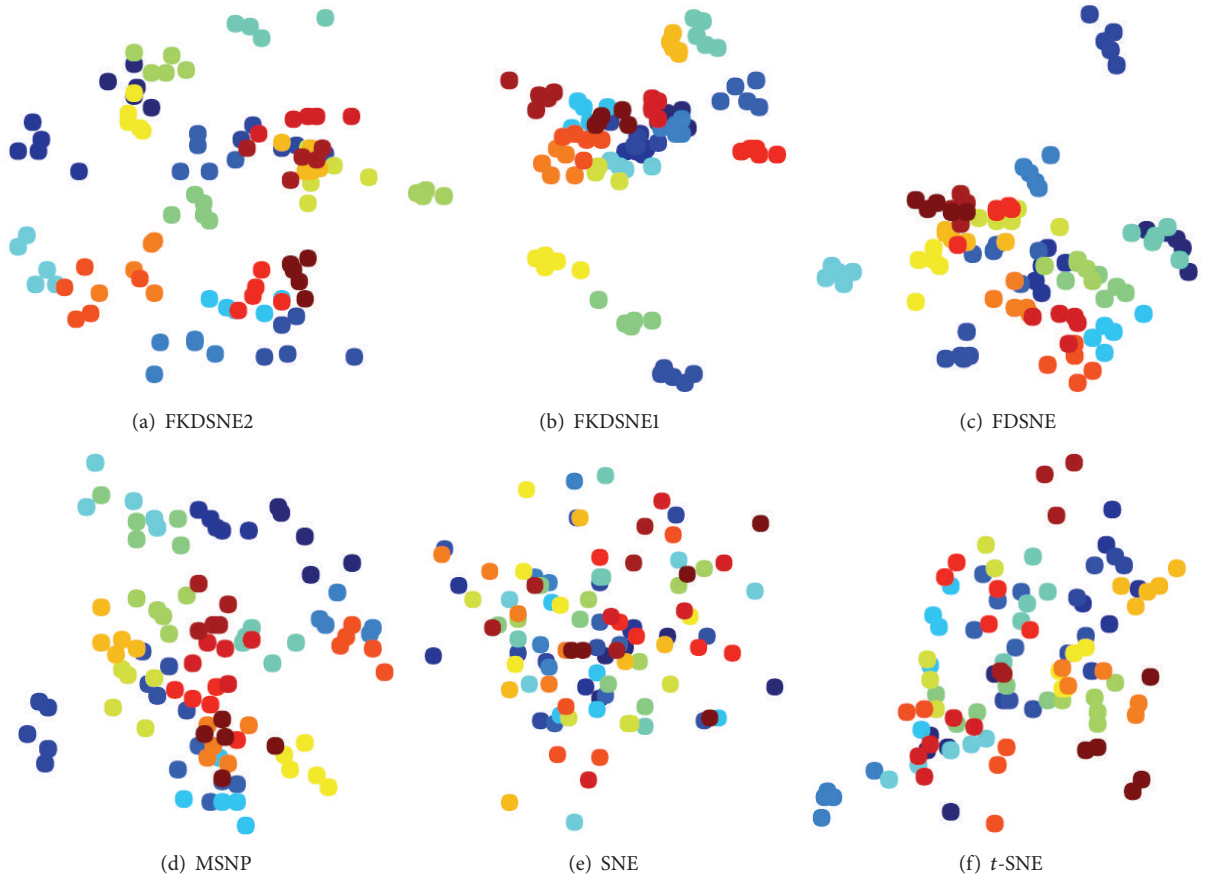
$$\begin{aligned} &= 4\mathbf{B}\left\{\left(\mathbf{K}\mathbf{D}^{1H}\mathbf{K}^T-\mathbf{K}\mathbf{U}^{1H}\mathbf{K}^T\right)\right. \\ &\quad \left.+\left(\mathbf{K}\mathbf{D}^{1M}\mathbf{K}^T-\mathbf{K}\mathbf{U}^{1M}\mathbf{K}^T\right)\right\} \\ &= 4\mathbf{B}\left\{\mathbf{K}\left(\mathbf{D}^{1H}-\mathbf{U}^{1H}+\mathbf{D}^{1M}-\mathbf{U}^{1M}\right)\mathbf{K}^T\right\},\end{aligned} \tag{17}$$

where $\mathbf{U}^{1H}$ is the $N$ order matrix with element $u_{ij}^{1H}$, and $\mathbf{U}^M$ is the $N$ order matrix with element $u_{ij}^{1M}$. Note that $\mathbf{U}^{1H}$ and $\mathbf{U}^{1M}$ are symmetric matrices; therefore, $\mathbf{D}^{1H}$ can be defined as a diagonal matrix that each entry is column (or row) sum of $\mathbf{U}^{1H}$ and the same for $\mathbf{D}^{1M}$, that is, $\mathbf{D}_{ii}^{1H} = \sum_j \mathbf{U}_{ij}^{1H}$ and $\mathbf{D}_{ii}^{1M} = \sum_j \mathbf{U}_{ij}^{1M}$.

For convenience, we name this kernel method as FKD-SNE1.

Another strategy is that we let $C^F(\mathbf{A})$ be the objective function in the embedding space $F$. So its gradient can be written as

$$\begin{aligned}&\frac{dC^F(\mathbf{A})}{d(\mathbf{A})} \\ &= \sum_{\forall j\in M_i}\frac{p_{ij}^1}{q_{ij}^1}\left(q_{ij}^1\right)' + \sum_{\forall j\in H_i}\frac{p_{ij}^1}{q_{ij}^1}\left(q_{ij}^1\right)' \\ &= 2\left[\sum_{\forall j\in H_i}p_{ij}^1\frac{\mathbf{B}\left(K_i-K_j\right)\left(\boldsymbol{\varphi}_i-\boldsymbol{\varphi}_j\right)^T}{\left(1+\left(K_i-K_j\right)^T\mathbf{B}^T\mathbf{B}\left(K_i-K_j\right)\right)}\right] \\ &\quad -2\left[\sum_{\forall j\in H_i}p_{ij}^1\left(\sum_{t\in H_m}\left(1+\left(K_m-K_t\right)^T\mathbf{B}^T\mathbf{B}\left(K_m-K_t\right)\right)^{-2}\right.\right. \\ &\qquad \left.\times\mathbf{B}\left(K_m-K_t\right)\left(\boldsymbol{\varphi}_m-\boldsymbol{\varphi}_t\right)^T\right) \\ &\qquad \left.\times\left(\sum_{t\in H_m}\left(1+\left(K_m-K_t\right)^T\mathbf{B}^T\mathbf{B}\left(K_m-K_t\right)\right)^{-1}\right)^{-1}\right] \\ &\quad +2\left[\sum_{\forall j\in M_i}p_{ij}^1\frac{\mathbf{B}\left(K_i-K_j\right)\left(\boldsymbol{\varphi}_i-\boldsymbol{\varphi}_j\right)^T}{1+\left(K_i-K_j\right)^T\mathbf{B}^T\mathbf{B}\left(K_i-K_j\right)}\right] \\ &\quad -2\left[\sum_{\forall j\in M_i}p_{ij}^1\left(\sum_{t\in M_m}\left(1+\left(K_m-K_t\right)^T\mathbf{B}^T\mathbf{B}\left(K_m-K_t\right)\right)^{-2}\right.\right. \\ &\qquad \left.\left.\times\mathbf{B}\left(K_m-K_t\right)\left(\boldsymbol{\varphi}_m-\boldsymbol{\varphi}_t\right)^T\right)\right.\end{aligned}$$



(a) FKDSNE2 (b) FKDSNE1 (c) FDSNE

(d) MSNP (e) SNE (f) $t$-SNE

Figure 4: Visualization of 100 images from COIL-20 images dataset.

$$\times \left( \sum_{t \in M_m} \left(1 + (K_m - K_t)^T \mathbf{B}^T \mathbf{B} (K_m - K_t)\right)^{-1} \right)^{-1} \Bigg]$$

$$= 2 \left[ \sum_{\forall j \in H_i} p_{ij}^1 w_{ij}^1 \mathbf{B} Q_{ij}^{(K_i - K_j)} - \sum_{t \in H_m} q_{mt}^1 w_{mt}^1 \mathbf{B} Q_{mt}^{(K_m - K_t)} \right] \mathbf{\Phi}$$

$$+ 2 \left[ \sum_{\forall j \in M_i} p_{ij}^1 w_{ij}^1 \mathbf{B} Q_{ij}^{(K_i - K_j)} - \sum_{t \in M_m} q_{mt}^1 w_{mt}^1 \mathbf{B} Q_{mt}^{(K_m - K_t)} \right] \mathbf{\Phi}$$

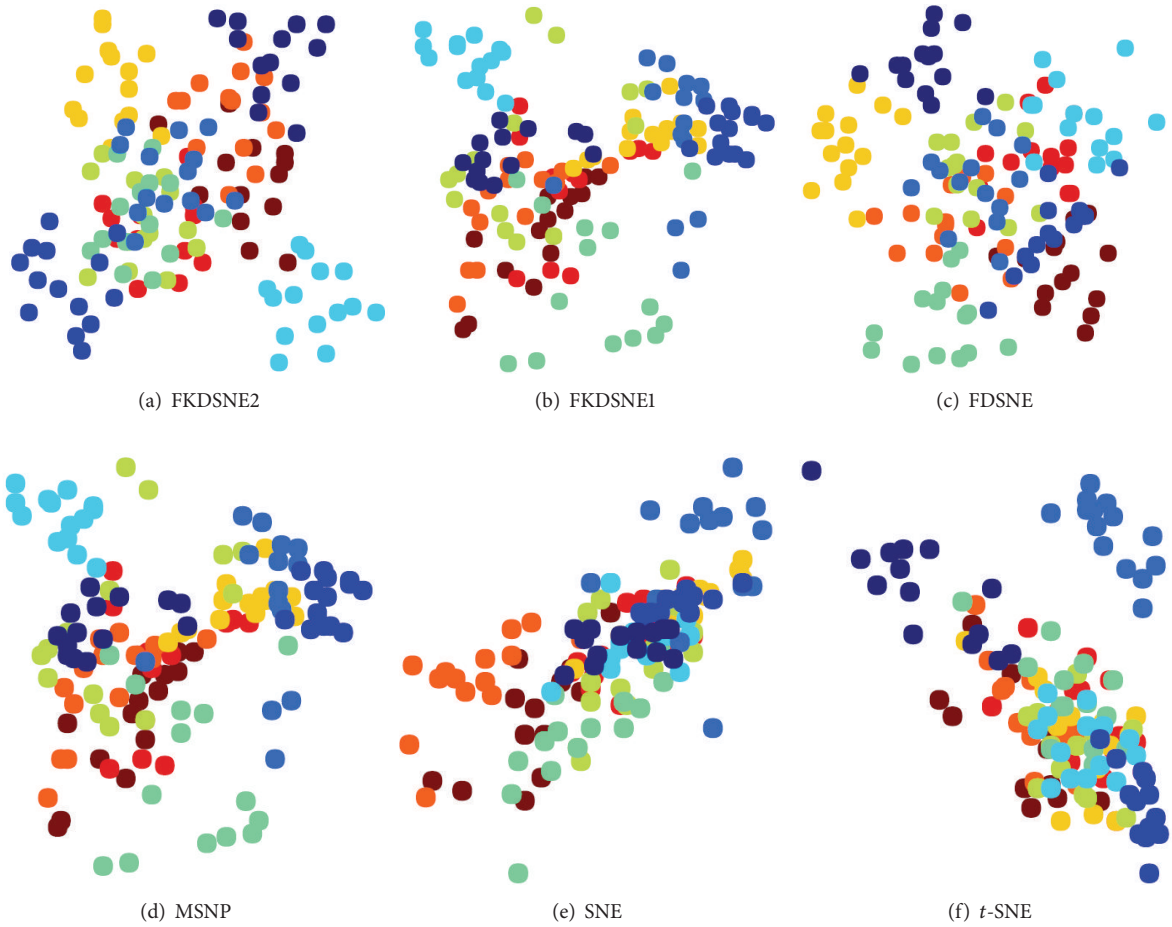
$$= 2 \left[ \sum_{\forall j \in H_i} u_{ij}^1 \mathbf{B} Q_{ij}^{(K_i - K_j)} + \sum_{\forall j \in M_i} u_{ij}^1 \mathbf{B} Q_{ij}^{(K_i - K_j)} \right] \mathbf{\Phi} \tag{18}$$

in this form, $Q_{ij}^{(K_i - K_j)}$ can be regard as the $N \times N$ matrix with vector $K_i - K_j$ in the $i$th column, and vector $K_j - K_i$ in the $j$th column and the other columns are all zeros.

This method is termed as FKDSNE2. Note that $\mathbf{\Phi}$ is a constant matrix. Furthermore, the observations of formula (18) make us know that updating the matrix $\mathbf{A}$ in the optimization only means updating the matrix $\mathbf{B}$. Additionally, $\mathbf{\Phi}$ does not need to be computed explicitly. Therefore, we do not need to explicitly perform the nonlinear map $\varphi(\mathbf{x})$ to minimize the objective function $C^F(\mathbf{A})$. The computational complexity of FKDSNE1 and FKDSNE2, is respectively, $O(2rN^2 + rNK)$ and $O(2rKN + rN^2)$ in each iteration. Hence, it is obvious that FKDSNE2 is faster than FKDSNE1 during each iteration.

## 5. Experiments

In this section, we evaluate the performance of our FDSNE, FKDSNE1, and FKDSNE2 methods for feature extraction. Three sets of experiments are carried out on Columbia Object Image Library (COIL-20) (http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php), US Postal Service (USPS) (http://www.cs.nyu.edu/~roweis/data.html), and ORL (http://www.cam-orl.co.uk) face datasets to demonstrate their good behavior on visualization, accuracy, and elapsed time. In the first set of experiments, we focus on the visualization of the proposed methods which are compared with that of the relevant algorithms, including SNE [17], $t$-SNE [18], and MSNP [21]. In the second set of experiments, we apply our methods to recognition task to verify their feature extraction capability and compare them with MSNP and DSNE [22]. Moreover, the elapsed time of FDSNE, FKDSNE1, FKDSNE2, and DSNE is compared in the third set of experiments. In particular, the Gaussian RBF kernel $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$ is chosen as the kernel function of FKDSNE1 and FKDSNE2, where $\sigma$ is set as the variance of the training sample set of $\mathbf{X}$.



(a) FKDSNE2 (b) FKDSNE1 (c) FDSNE

(d) MSNP (e) SNE (f) $t$-SNE

Figure 5: Visualization of 140 images from USPS handwritten digits dataset.

*5.1. COIL-20, USPS, and ORL Datasets.* The datasets used in our experiments are summarized as follows.

COIL-20 is a dataset of gray-scale images of 20 objects. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each image is $40 \times 40$ pixels. Figure 1 shows sample images from COIL-20 images dataset.

USPS handwritten digit dataset includes 10 digit characters and 1100 samples in total. The original data format is of $16 \times 16$ pixels. Figure 2 shows samples of the cropped images from USPS handwritten digits dataset.

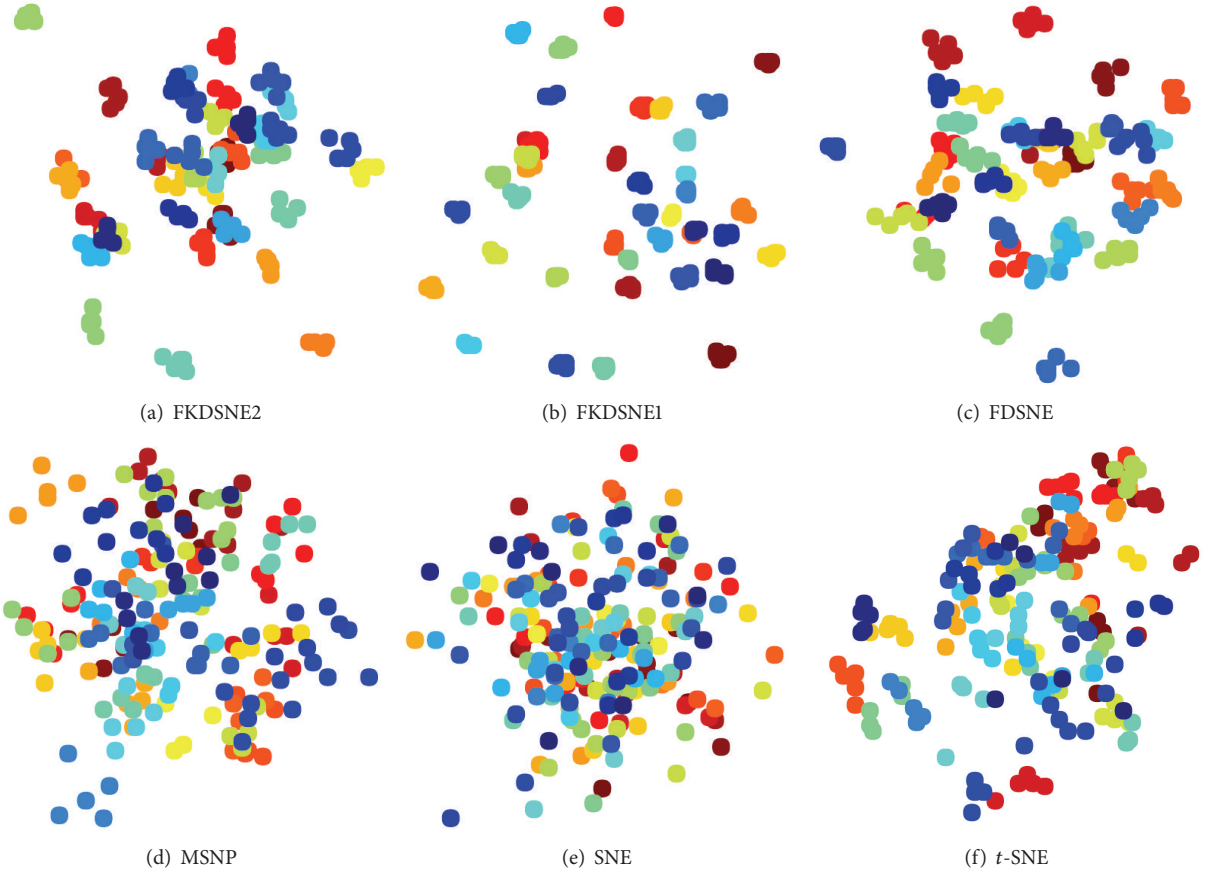
ORL consists of gray images of faces from 40 distinct subjects, with 10 pictures for each subject. For every subject, the images were taken with varied lighting condition and different facial expressions. The original size of each image is $112 \times 92$ pixels, with 256 gray levels per pixel. Figure 3 illustrates a sample subject of ORL dataset.

*5.2. Visualization Using FDSNE, FKDSNE1, and FKDSNE2.* We apply FDSNE, FKDSNE1, and FKDSNE2 to visualization task to evaluate their capability of classification performance. The experiments are carried out, respectively, on COIL-20, USPS, and ORL datasets. For the sake of computational efficiency as well as noise filtering, we first adjust the size of each image to $32 \times 32$ pixels on ORL, and then we select five samples from each class on COIL-20, fourteen samples from each class on USPS, and five samples from each class on ORL.

The experimental procedure is to extract a 20-dimensional feature for each image by FDSNE, FKDSNE1, and FKDSNE2, respectively. Then to evaluate the quality of features through visual presentation of the first two-dimensional feature.

FDSNE, FKDSNE1, and FKDSNE2 are compared with three well known visualization methods for detecting classification performance: (1) SNE, (2) $t$-SNE, and (3) MSPN. The parameters are set as follows: the $K$-nearest neighborhood parameter of FDSNE, FKDSNE1, and FKDSNE2 methods is $K_1 = h - 1$ (let $h$ denote the number of training samples in each class), $K_2 = 40$; for SNE and $t$-SNE, the perplexity parameter is perp $= 20$ and the iteration number is $Mt = 1000$; for MSNP, the degree freedom of Cauchy distribution is $\gamma = 4$ and the iteration number is 1000 as well.

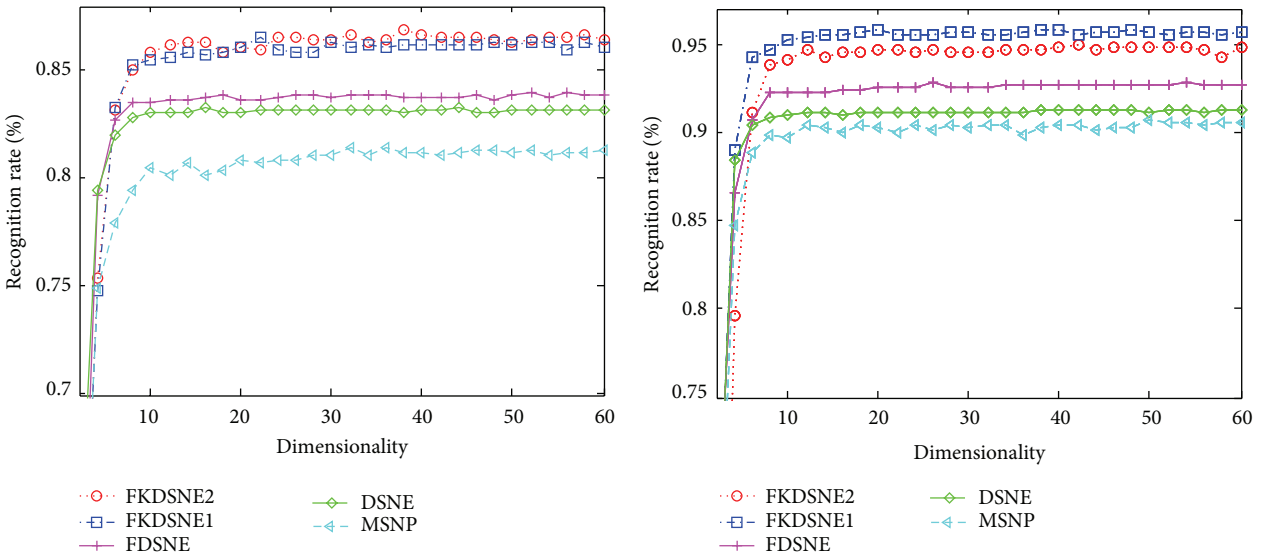
Figures 4, 5, and 6 show the visual presentation results of FDSNE, FKDSNE1, FKDSNE2, SNE, $t$-SNE, and MSNP, respectively, on COIL-20, USPS, and ORL datasets. The visual presentation is represented as a scatterplot in which a different color determines different class information. The figures reveal that the three nearest-neighbor-based methods,



(a) FKDSNE2 (b) FKDSNE1 (c) FDSNE

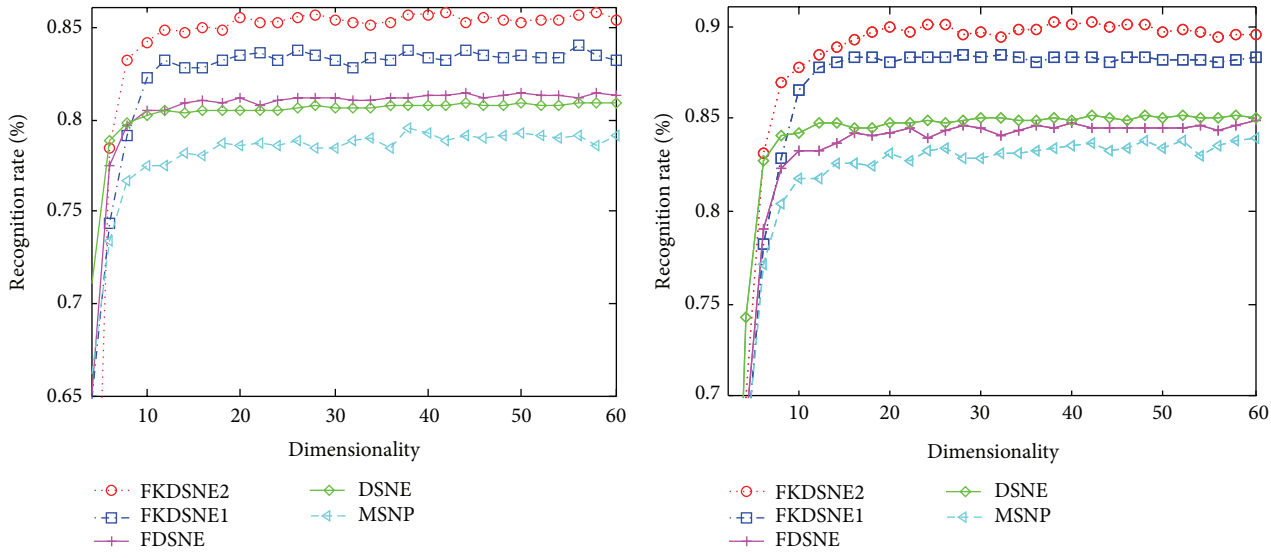
(d) MSNP (e) SNE (f) $t$-SNE

Figure 6: Visualization of 200 face images from ORL faces dataset.



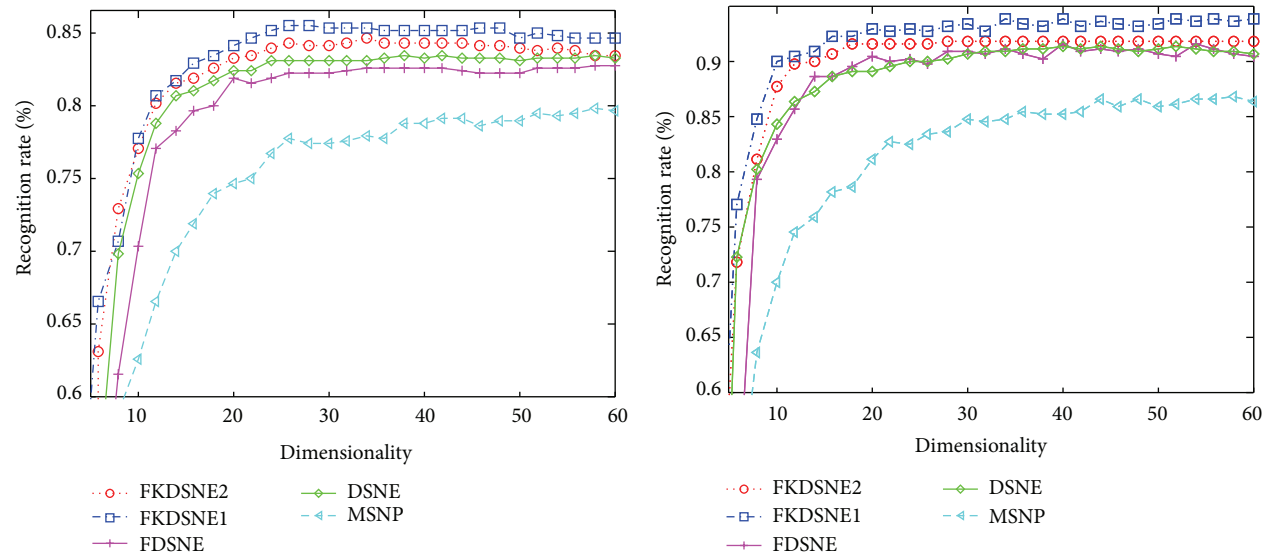

(a) $h = 5$ (b) $h = 10$

Figure 7: Recognition rate (%) versus subspace dimension on COIL-20.




(a) $h = 14$ (b) $h = 25$

Figure 8: Recognition rate (%) versus subspace dimension on USPS.




(a) $h = 3$ (b) $h = 5$

Figure 9: Recognition rate (%) versus subspace dimension on ORL.

that is, FDSNE, FKDSNE1, and FKDSNE2, give considerably better classification result than SNE, $t$-SNE, and MSNP on all datasets, for the separation between classes is quite obvious. In particular, SNE and $t$-SNE not only get less separation for the interclass data but also produce larger intraclass scatter. For MSNP, it has smaller intraclass scatter, but there exists an overlapping phenomenon among classes. With regard to FDSNE, FKDSNE1, and FKDSNE2, we can find from the figures that FKDSNE1 shows the best classification performance among all the algorithms on ORL face dataset, while not on the other two datasets COIL-20 and USPS; thereinto, the classification performance of FKDSNE1 is inferior to FDSNE on COIL-20 while on USPS it is inferior to FKDSNE2. In addition, the clustering qualities and separation degree of FKDSNE1 and FKDSNE2 are obviously better than that of FDSNE.

*5.3. Recognition Using FDSNE, FKDSNE1, and FKDSNE2.* In this subsection, we apply FDSNE, FKDSNE1, and FKDSNE2 to recognition task to verify their feature extraction capability. Nonlinear dimensional reduction algorithms such as SNE and $t$-SNE lack explicit projection matrix for the out-of-sample data, which means they are not suitable for recognition. So we compare the proposed methods with DSNE and

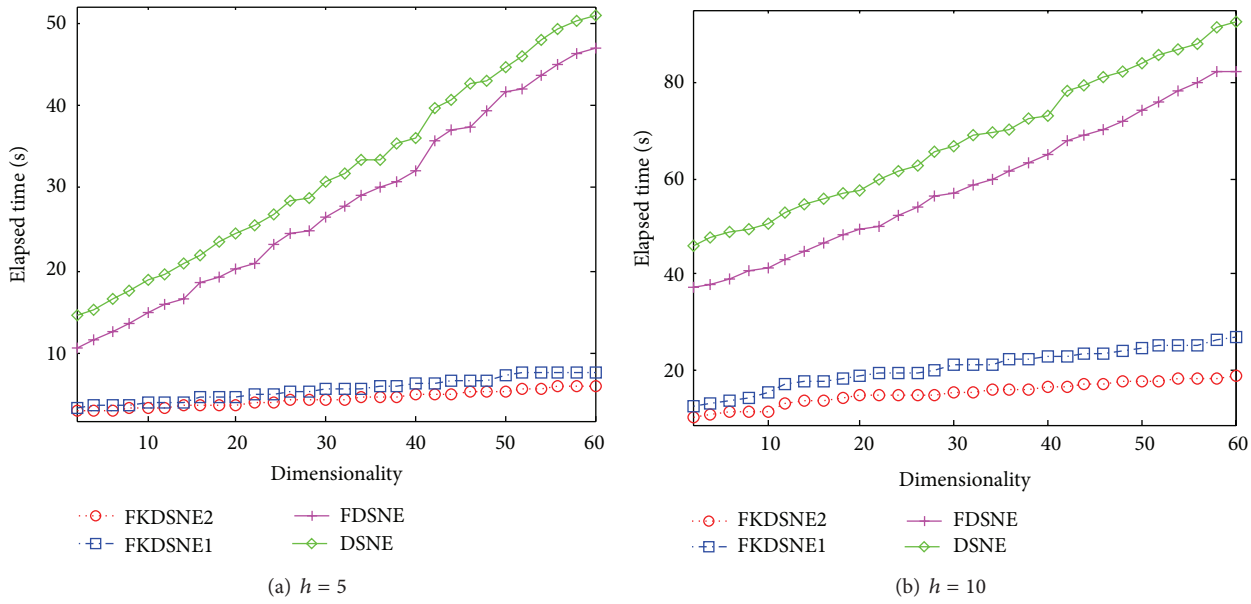



Figure 10: Elapsed time (seconds) versus subspace dimension on COIL-20.

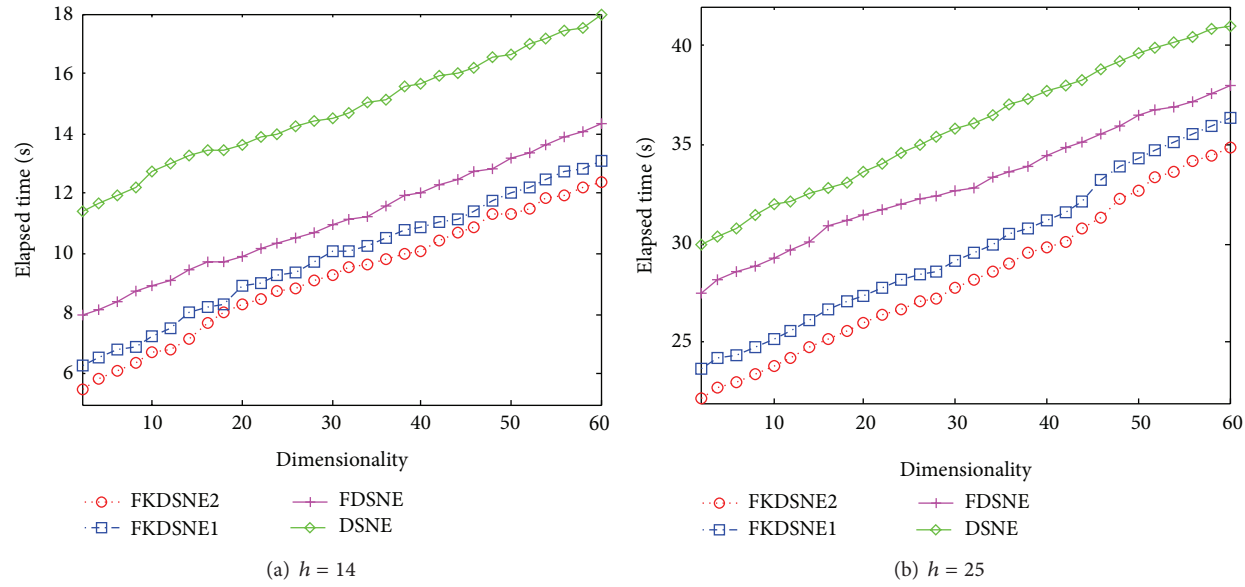
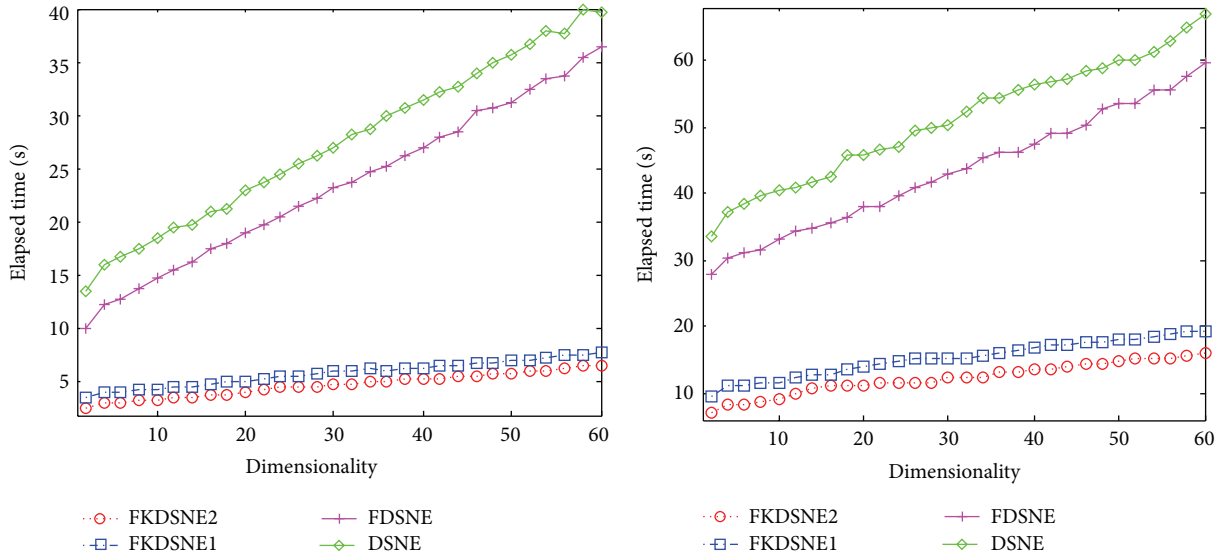



Figure 11: Elapsed time (seconds) versus subspace dimension on USPS.

MSNP, both of them are linear methods and were proved to be better than existing feature extraction algorithms such as SNE, $t$-SNE, LLTSA, LPP, and so on in [21, 22]. The procedure of recognition is described as follows: firstly, divide dataset into training sample set $\mathbf{X}_{\text{train}}$ and testing sample set $\mathbf{X}_{\text{test}}$ randomly; secondly, the training process for the optimal matrix $\mathbf{A}$ or $\mathbf{B}$ is taken for FDSNE, FKDSNE1 and FKDSNE2; thirdly, feature extraction is accomplished for all samples using $\mathbf{A}$ or $\mathbf{B}$; finally, a testing image is identified by a nearest neighbor classifier. The parameters are set as follows: the $K$-nearest neighborhood parameter $K_1, K_2$ in FDSNE, FKDSNE1, and FKDSNE2 is $K_1 = h - 1$, $K_2 = 40$; for DSNE, the perplexity parameter is $\lambda = 0.1$ and the iteration number is $Mt = 1000$; for MSNP, the freedom degree $\gamma$ of Cauchy distribution in MSNP is determined by cross validation and the iteration number is 1000 as well.

Figure 7 demonstrates the effectiveness of different subspace dimensions for COIL-20 ((a): $h = 5$, (b): $h = 10$). Figure 8 is the result of the experiment in USPS ((a): $h = 14$, (b): $h = 25$), and Figure 9 shows the recognition rate versus subspace dimension on ORL ((a): $h = 3$, (b): $h = 5$). The maximal recognition rate of each method and the corresponding dimension are given in Table 1, where the number in bold stands for the highest recognition rate. From Table 1,




(a) $h = 3$ (b) $h = 5$

Figure 12: Elapsed time (seconds) versus subspace dimension on ORL.

Table 1: The maximal recognition rates (%) versus the subspace dimension.

| | COIL-20 $h = 5$ | COIL-20 $h = 10$ | USPS $h = 14$ | USPS $h = 25$ | ORL $h = 3$ | ORL $h = 5$ |
|---|---|---|---|---|---|---|
| MSNP | 0.8149 (32) | 0.9063 (50) | 0.7958 (38) | 0.8395 (58) | 0.7989 (59) | 0.8690 (58) |
| DSNE | 0.8325 (36) | 0.9130 (54) | 0.8093 (50) | 0.8522 (42) | 0.8357 (42) | 0.9150 (39) |
| FDSNE | 0.8396 (52) | 0.9277 (54) | 0.8150 (58) | 0.8489 (59) | 0.8279 (58) | 0.9160 (39) |
| FKDSNE1 | 0.8651 (22) | **0.9575 (20)** | 0.8409 (26) | 0.8848 (26) | **0.8550 (26)** | **0.9405 (24)** |
| FKDSNE2 | **0.8689 (28)** | 0.9491 (22) | **0.8585 (22)** | **0.9021 (28)** | 0.8470 (24) | 0.9193 (20) |

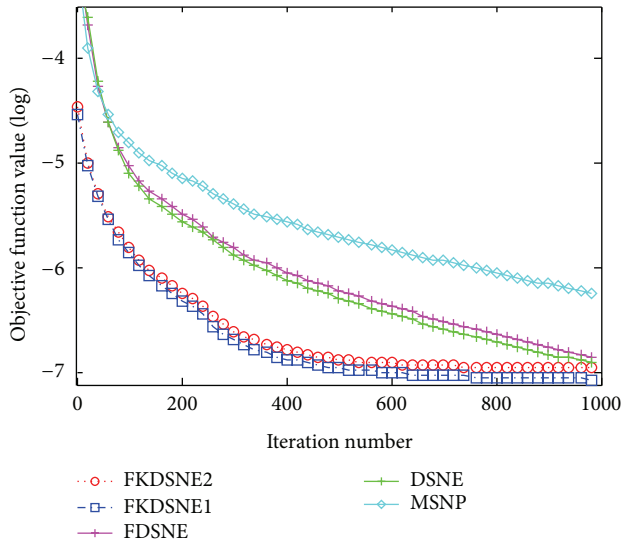



Figure 13: Objective function value (log) versus iterative number on ORL dataset.

we can find that FKDSNE1 and FKDSNE2 outperform MSNP, DSNE, and FDSNE on COIL-20, USPS, and ORL. As can be seen, FKDSNE1 and FKDSNE2 enhance the maximal recognition rate for at least 2% compared with other three methods. Besides, FKDSNE1 and FKDSNE2 achieve considerable recognition accuracy when feature dimension is 20 on the three datasets. It indicates that FKDSNE1 and FKDSNE2 grasp the key character of face images relative to identification with a few features. Though the maximal recognition rate of DSNE and FDSNE is closer to that of FKDSNE1 and FKDSNE2 on ORL dataset, the corresponding dimension of FKDSNE1 and FKDSNE2 is 20 while that of DSNE and FDSNE exceeds 30. From the essence of dimensional reduction, this result demonstrates that FDSNE and DSNE are inferior to FKDSNE1 and FKDSNE2.

*5.4. Analysis of Elapsed Time.* In this subsection, we further compare the computational efficiency of DSNE, FKDSNE, FKDSNE1, and FKDSNE2. The algorithm MSPN is not compared since its recognition rate is obviously worse than other algorithms. The parameters of the experiment are the same to Section 5.3. Figures 10, 11, and 12, respectively, show the elapsed time of four algorithms under different subspace dimensions on the three datasets. It can be observed from the figures that FKDSNE2 has the lowest computational cost among the four algorithms while DSNE is much inferior to other nearest-neighbor-based algorithms on all datasets. Particularly, on the COIL-20 dataset, the elapsed time of FKDSNE2 is more than 2 times faster than DSNE. As for DSNE

and FDSNE, the former is obviously slower than the latter. Besides, for the two kernel methods, FKDSNE2 is notably faster than FKDSNE1, which confirms our discussion in Section 4.

Furthermore, kernel-based algorithms FKDSNE1 and FKDSNE2 can effectively indicate the linear structure on high-dimensional space. Their objective function can achieve better values on desirable dimensions. For instance, Figure 13 illustrates the objective function value of MSNP, DSNE, FKDSNE, FKDSNE1, and FKDSNE2 versus iterative number on ORL dataset. It can be found that FKDSNE2 and FKDSNE1 is close to the convergence value $1e-7$ while FDSNE and DSNE only achieve $1e-6$ and MSNP achieves $1e-5.4$ when the iterative number is 400. It means that FKDSNE1 and FKDSNE2 can get the more precise objective function value with less iterative number compared with DSNE and FDSNE; that is to say that, FKDSNE1 and FKDSNE2 can achieve the same value by using forty percent of the elapsed time of DSNE and FDSNE.

## 6. Conclusion

On the basis of DSNE, we present a method called fast discriminative stochastic neighbor embedding analysis (FDSNE) which chooses the reference points in $K$-nearest neighbors of the target sample from the same class and the different classes instead of the total training samples and thus has much lower computational complexity than that of DSNE. Furthermore, since FDSNE is a linear feature dimensionality reduction algorithm, we extend FDSNE to a nonlinear scenario using techniques of kernel trick and present two kernel-based methods: FKDSNE1 and FKDSNE2. Experimental results on COIL-20, USPS, and ORL datasets show the superior performance of the proposed methods. Our future work might include further empirical studies on the learning speed and robustness of FDSNE by using more extensive, especially large-scale, experiments. It also remains important to investigate acceleration techniques in both initialization and long-run stages of the learning.

## Acknowledgment

This project was partially supported by Zhejiang Provincial Natural Science Foundation of China (nos. LQ12F03011 and LQ12F03005).

## References

[1] E. Cherchi and C. A. Guevara, "A Monte Carlo experiment to analyze the curse of dimensionality in estimating random coefficients models with a full variance-covariance matrix," *Transportation Research B*, vol. 46, no. 2, pp. 321–332, 2012.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces versus fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[5] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *Journal of Machine Learning Research*, vol. 8, pp. 1027–1061, 2007.

[6] W. Bian and D. Tao, "Max-min distance analysis by using sequential SDP relaxation for dimension reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 1037–1050, 2011.

[7] Z. Teng, J. He et al., "Critical mechanical conditions around neovessels in carotid atherosclerotic plaque may promote intraplaque hemorrhage," *Atherosclerosis*, vol. 223, no. 2, pp. 321–326, 2012.

[8] Z. Teng, A. J. Degnan, U. Sadat et al., "Characterization of healing following atherosclerotic carotid plaque rupture in acutely symptomatic patients: an exploratory study using in vivo cardiovascular magnetic resonance," *Journal of Cardiovascular Magnetic Resonance*, vol. 13, article 64, 2011.

[9] C. E. Hann, I. Singh-Levett, B. L. Deam, J. B. Mander, and J. G. Chase, "Real-time system identification of a nonlinear four-story steel frame structure-application to structural health monitoring," *IEEE Sensors Journal*, vol. 9, no. 11, pp. 1339–1346, 2009.

[10] A. Segui, J. P. Lebaron, and R. Leverge, "Biomedical engineering approach of pharmacokinetic problems: computer-aided design in pharmacokinetics and bioprocessing," *IEE Proceedings D*, vol. 133, no. 5, pp. 217–225, 1986.

[11] F. Wu, Y. Zhong, and Q. Y. Wu, "Online classification framework for data stream based on incremental kernel principal component analysis," *Acta Automatica Sinica*, vol. 36, no. 4, pp. 534–542, 2010.

[12] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[15] H. Li, H. Jiang, R. Barrio, X. Liao, L. Cheng, and F. Su, "Incremental manifold learning by spectral embedding methods," *Pattern Recognition Letters*, vol. 32, no. 10, pp. 1447–1455, 2011.

[16] P. Zhang, H. Qiao, and B. Zhang, "An improved local tangent space alignment method for manifold learning," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 181–189, 2011.

[17] G. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in Neural Information Processing Systems*, vol. 15, pp. 833–840, 2002.

[18] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[19] J. A. Cook, I. Sutskever, A. Mnih, and G. E. Hinton, "Visualizing similarity data with a mixture of maps," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, vol. 2, pp. 67–74, 2007.

[20] Z. R. Yang, I. King, Z. L. Xu, and E. Oja, "Heavy-tailed symmetric stochastic neighbor embedding," *Advances in Neural Information Processing Systems*, vol. 22, pp. 2169–2177, 2009.

[21] S. Wu, M. Sun, and J. Yang, "Stochastic neighbor projection on manifold for feature extraction," *Neurocomputing*, vol. 74, no. 17, pp. 2780–2789, 2011.

[22] J. W. Zheng, H. Qiu, Y. B. Jiang, and W. L. Wang, "Discriminative stochastic neighbor embedding analysis method," *Computer-Aided Design & Computer Graphics*, vol. 24, no. 11, pp. 1477–1484, 2012.

[23] C. Cattani, R. Badea, S. Chen, and M. Crisan, "Biomedical signal processing and modeling complexity of living systems," *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 298634, 2 pages, 2012.

[24] X. Zhang, Y. Zhang, J. Zhang et al., "Unsupervised clustering for logo images using singular values region covariance matrices on Lie groups," *Optical Engineering*, vol. 51, no. 4, Article ID 047005, 8 pages, 2012.

[25] P. J. Moreno, P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," *Advances in Neural Information Processing Systems*, vol. 16, pp. 1385–1393, 2003.