Locality Preserving Projections with Adaptive Neighborhood Size

Wenjun Hu [a,b,c,*], Kup-Sze Choi [c], Shitong Wang [b]

[a] School of Information and Engineering, Huzhou Teachers College, Huzhou, Zhejiang, China

[b] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China

[c] Centre for Integrative Digital Health, Hong Kong Polytechnic University, Hong Kong, China

Abstract: Feature extraction methods are widely employed to reduce dimensionality of data and enhance the discriminative information. Among the methods, manifold learning approaches have been developed to detect the underlying manifold structure of the data based on local invariants, which are guaranteed by an adjacent graph of the sampled data set. The performance of the manifold learning approaches is however affected by the locality of the data. In this paper, we address this issue by proposing a method to adaptively select the neighborhood size. It is applied to the manifold learning approach Locality Preserving Projections (LPP) which is a popular linear reduction algorithm. The effectiveness of the adaptive neighborhood selection method is evaluated by performing classification and clustering experimental on the real-life data sets.

Index Terms — Dimensionality reduction, feature extraction, neighborhood size, locality preserving projections.

1 Introduction

High dimensional data are prevalent in many application domains, such as pattern recognition, information retrieval, text categorization, computer vision and data mining. Since processing the high dimensional data requires considerable computational time and storage space, it is necessary to preprocess the data before beginning the learning tasks such as classification and clustering. Two popular preprocessing methods are feature selection and feature extraction techniques.

Many feature extraction techniques have been proposed in the past few decades. The most well-known techniques include Principle Component Analysis (PCA) [5], [6] and Linear Discriminant Analysis (LDA) [7], [8]. PCA is an unsupervised dimensionality reduction technique used to find a set of orthogonal bases by which the global information of the data in the principle component space is captured. The principle component space is obtained by solving an eigenvalue problem, which is mathematically equivalent to performing Singular Value Decomposition (SVD) in the centered data matrix. On the other hand, LDA is a supervised dimensionality reduction approach used to obtain the optimal projection or transformation subspace of the data, which is obtained by minimizing the within-class scatter matrix and maximizing the between-class scatter matrix.

Recently, researchers consider that data sampled from a probability distribution may be on, or in close proximity to, a

The following publication Hu, W., Cheng, X., Jiang, Y., Choi, K. S., & Lou, J. (2017). Locality preserving projections with adaptive neighborhood size. In Proceedings of Intelligent Computing Theories and Application: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, pp. 223-234. Springer International Publishing is available at https://doi.org/10.1007/978-3-319-63309-1_21.

low-dimensional manifold of the ambient space [1], [2], [4], [9]. Many manifold learning algorithms have been proposed to deal with such kind of data, including ISOMAP [1], Laplacian Eigenmap (LE) [4], Locally Linear Embedding (LLE) [2] and Locality Preserving Projections (LPP) [3]. The former two methods are nonlinear algorithms, while the latter two are linear algorithms. Generally, most of the manifold methods make use the notion of local invariance to detect the underlying manifold structure so that the low-dimensional representations can be obtained. Besides, the local invariance is guaranteed by using an adjacent graph of the sampled data set. Here, the construction of valid adjacent graph becomes an important issue which directly impacts the performance of the manifold learning approaches. That is to say, what can be considered as local? Therefore, it is noteworthy to select the neighborhood size to construct the adjacent graph for matching the local geometry of the manifold. In this paper, we focus on this problem and propose a method to adaptively select the neighborhood size, which is validated in the original LPP method.

The rest of the paper is organized as follows. Section 2 provides a brief description of the LPP algorithm. Section 3 introduces the adaptive neighborhood selection method and Section 4 presents the experimental results of classification and clustering obtained with the proposed method. The paper is concluded in Section 5.

2 LOCALITY PRESERVING PROJECTIONS

LPP is one of the most popular manifold learning methods which are based on the spectral graph theory [10]. Generally speaking, the aim of manifold learning method is to find an optimal map to preserve the intrinsic geometry structure of the data manifold so that the geometry structure information of the data can be discovered. Let $\mathbf{y} = [y_1, \cdots, y_N]$ be the one-dimensional map of the data set $\mathbf{X}$. Given a *k*-nearest neighbor graph G with weight matrix $\mathbf{W}$, LPP attempts to obtain the optimal map by solving the following optimization criterion:

$$\min \sum_{i,j=1}^{m} (\mathbf{v}^T\mathbf{x}_i - \mathbf{v}^T\mathbf{x}_j)^2 \mathbf{W}_{ij} \ . \tag{1}$$

where $\mathbf{v} \in \Re^D$ is a base vector (i.e. $y_i = \mathbf{v}^T\mathbf{x}_i$) and $\mathbf{W} = [\mathbf{W}_{ij}]$ is a weight matrix which reveals the neighborhood relationship between the data points. The weight matrix $\mathbf{W}$ can be defined as:

$$\mathbf{W}_{ij} = \begin{cases} \exp(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / t) & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where $\exp(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / t)$ is the heat kernel function with a suitable constant t (t is usually called the kernel width parameter) and $N(\mathbf{x}_j)$ denotes the set of *k*-nearest neighbors or the ε neighborhoods of $\mathbf{x}_j$. For simplicity, the weight matrix can also be defined with the so-called 0-1 weight, i.e.

$$\mathbf{W}_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i), \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Clearly, the minimization criterion in (1) is an attempt to ensure that the points y_i and y_j are also close to each other if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close". That is to say, the obtained optimal map attempts to preserve the local structure of the data. In fact, the purpose of LPP is to find a group of basis vectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_d] \in \Re^{D\times d}$ and obtain a subspace to preserve the local structure of the data set. With the constraint $\mathbf{v}^T\mathbf{XDX}^T\mathbf{v} = 1$ and by some algebraic steps, the objective function in (1) can be reduced to a generalized minimum eigenvalue problem $\mathbf{XLX}^T\mathbf{v} = \lambda\mathbf{XDX}^T\mathbf{v}$, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix [3], [10], and $\mathbf{D}$ is a diagonal matrix whose entries along the diagonal are the column sum of $\mathbf{W}$, i.e. $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. Details of the solution to the eigenvalue problem can be found in [3].

3 LPP WITH ADAPTIVE NEIGHBORHOOD SIZE

Given a sample space $\Gamma \subset \Re^D$, let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \subset \Gamma$ denote the data set with $\mathbf{x}_i \in \Re^D$, which is sampled from a *d*-dimension submanifold embedded in $\Re^D$. Suppose we have known the nearest neighbor $\mathbf{z}_i$ of each sample $\mathbf{x}_i$ in Γ, then we can determine whether $\mathbf{x}_{j\neq i}$ is the neighbor of $\mathbf{x}_i$ by comparing $\|\mathbf{x}_j - \mathbf{x}_i\|$ and $\|\mathbf{z}_i - \mathbf{x}_i\|$. Clearly, $\mathbf{x}_{j\neq i} \in N(\mathbf{x}_i)$, if $\|\mathbf{x}_j - \mathbf{x}_i\| \le \|\mathbf{z}_i - \mathbf{x}_i\|$, $\mathbf{x}_{j\neq i} \notin N(\mathbf{x}_i)$; otherwise, $N(\mathbf{x}_i)$ is the set of nearest neighbors of $\mathbf{x}_i$. Hence, our goal is to estimate the nearest neighbor $\mathbf{z}_i$ of each sample $\mathbf{x}_i$ in Γ. Two effective estimation techniques are developed and will be discussed in detail in this section.

3.1 Estimation of Nearest Neighbors

Consider the nearest neighbors of a given sample $\mathbf{x}_i$ as hidden random variables. By employing the principles of the expectation-maximization algorithm [11], the nearest neighbors $\mathbf{z}_i$ can be estimated by computing the expectation of $\mathbf{x}_i$, i.e. averaging out the hidden variables, which is given by

$$\mathbf{z}_i = \mathrm{E}_{\mathbf{x}_i \sim \tilde{\mathbf{X}}_i}(\mathbf{x}_j) = \sum_{\mathbf{x}_j \in \tilde{\mathbf{X}}_i} p(\mathbf{x}_j)\mathbf{x}_j \,, \quad (4)$$

where $\tilde{\mathbf{X}}_i = [\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_N]$ and $\mathbf{x}_j \in \tilde{\mathbf{X}}_i$, $\mathrm{E}_{\mathbf{x}_i \sim \tilde{\mathbf{X}}_i}$ denotes the expectation computed with respect to $\tilde{\mathbf{X}}_i$ and $p(\mathbf{x}_j)$ is the probability of $\mathbf{x}_j$ being the nearest neighbor of $\mathbf{x}_i$. The probability can be obtained based on standard kernel density estimation as follows:

$$p(\mathbf{x}_j) = \frac{k_\sigma(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{\mathbf{x}_l \in \tilde{\mathbf{X}}_i} k_\sigma(\mathbf{x}_i, \mathbf{x}_l)}, \tag{5}$$

where $k_\sigma(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function with a width parameter σ. Here, any kernel functions with the properties described in [12] can be used for the estimation. In this study, the Gaussian kernel is adopted. It is formulated as

$$k_\sigma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}) \tag{6}$$

After the nearest neighbor $\mathbf{z}_i$ of the data $\mathbf{x}_i$ is obtained by the estimation, the samples in $\mathbf{X}$ that belong to the nearest neighbors $N(\mathbf{x}_i)$ of the data $\mathbf{x}_i$ are then determined.

3.2 Estimation of Nearest Neighborhood

Similar to the estimation method presented above, the nearest neighborhood can be computed by the expectation of $\left\|\mathbf{x}_j - \mathbf{x}_i\right\|^2$, i.e.

$$\varepsilon_i = \mathrm{E}_{\mathbf{x}_i \sim \tilde{\mathbf{X}}_i}(\left\|\mathbf{x}_j - \mathbf{x}_i\right\|^2) = \sum_{\mathbf{x}_j \in \tilde{\mathbf{X}}_i} p(\mathbf{x}_j)\left\|\mathbf{x}_j - \mathbf{x}_i\right\|^2, \tag{7}$$

where ε_i is the neighborhood size of $\mathbf{x}_i$. After obtaining the neighborhood size ε_i, the samples in $\mathbf{X}$ that satisfy the condition $\left\|\mathbf{x} - \mathbf{x}_i\right\|^2 \le \varepsilon_i$ are used to construct the set of the neighbors $N(\mathbf{x}_i)$ of the data $\mathbf{x}_i$.

For simplicity, the Euclidean distance metric in (7) is used in our experiment. In fact, other distance metrics can be used. For example, the Mahalanobis distance (MD) is a possible alternative that is affine invariant and more robust. Meanwhile, the distance metric in kernel function given in (6) can also be replaced by MD.

3.3 Adaptive Neighborhood Size for LPP

By using the two estimation techniques, the neighborhood size to LPP can be then adaptively selected. The LLP algorithm is also extended to yield a new algorithm with adaptive neighborhood size. The new LLP algorithm is denoted LPP$_{\text{ANS}}$ and described as follows:

Algorithm 1. LPP$_{\text{ANS}}$

Input: Data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \Re^{D \times N}$, kernel width parameter σ;

Output: Basis vectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_d] \in \Re^{D \times d}$.

1. Estimate the nearest neighbor of each data $\mathbf{x}_i$ using (4) or the nearest neighborhood of each data using (7);

```
2. Find the neighbors $N(\mathbf{x}_i)$ of the data $\mathbf{x}_i$ in $\mathbf{X}$;
3. Compute the corresponding weight matrix $\mathbf{W}$ using (2) or (3);
4. Compute the diagonal matrix $\mathbf{D}$ ($\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$) and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
5. Compute the eigenvectors and eigenvalues for the generalized eigenvector problem $\mathbf{XLX}^T\mathbf{v} = \lambda\mathbf{XDX}^T\mathbf{v}$;
6. Output the basis vectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_d]$ with the eigenvectors corresponding to the smallest $d$ eigenvalues .
```

Note: In Step 2, if no neighbor of the data $\mathbf{x}_i$ has been, the $k_{\min}$ points nearest to $\mathbf{x}_i$ will be considered as its neighbors $N(\mathbf{x}_i)$ for the stability of the LPP algorithm. The value of $k_{\min}$ is fixed at 2 in our experiments.

4 EXPERIMENT RESULTS

To investigate the effectiveness of the LPP algorithm with adaptively selected neighborhood size (LPP$_{\mathrm{ANS}}$), experiments are evaluate its classification and clustering performance respectively. The results are compared with that of PCA and LPP methods with different neighborhood size.

4.1 Data Preparation

Two real world data sets are employed for the experiments. The first data set is the COIL20 image library from the Columbia University. It is used in the classification experiment. The data set contains 1440 images generated from 20 objects. Each image is represented by a 1024-dimensional vector, and the size is 32 × 32 pixels with 256 grey levels per pixel. Further details can be found in http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

The second data set is obtained from the PIE face database of the Carnegie Mellon University (CMU) (downloadable from http://www.cad.zju.edu.cn/home/dengcai/), which is used in the clustering experiment. The PIE face images are created under different poses, illuminations and expressions. This database contains 41,368 images of 68 subjects. The image size is 32×32 pixels with 256 grey levels. In our experiment, we select 1428 images of different illuminations for clustering. In the following sections, the classification experiment is first discussed, followed by the clustering experiment.

4.2 Classification

In this experiment, we employ the 1-nearest neighbor (1-NN) classifier to evaluate the discriminating power of the features that are extracted in four different ways, including the proposed LPP$_{\mathrm{ANS}}$ algorithm, LPP with neighborhood size k=3 (LPP$_{k=3}$), LPP with neighborhood size k=10 (LPP$_{k=10}$) and the PCA method. Half of the images in the COIL20 library are selected by random and used for training. The remaining images are used for testing. The test is performed with different number of the extracted features in the range of {5, 10, 20, …, 250}. For each feature number in such range, the test is executed for 10 times to evaluate the average performance. Here, the performance metric is the classification accuracy, which is computed as

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\delta(l_o(\mathbf{x}_i), l_t(\mathbf{x}_i)),$$

where $\mathbf{x}_i$ is the test sample, $l_o(\mathbf{x}_i)$ is the true class label of $\mathbf{x}_i$, $l_t(\mathbf{x}_i)$ is the class label obtained by 1-NN, n is the size of the testing data set, and the function $\delta(l_o(\mathbf{x}_i), l_t(\mathbf{x}_i))$ equals 1 if $l_o(\mathbf{x}_i) = l_t(\mathbf{x}_i)$ and 0 otherwise. The higher the classification accuracy, the better the extracted features.

Fig.1 shows the classification results on the COIL20 data set. It can be seen that the proposed method of adaptive neighborhood size selection improves the LPP algorithm. The classification accuracy of LPP_{ANS} is higher than that of the other four methods. The performance of LPP_{ANS} also remains stable for feature number in the range of 5 to 250 features. Among the five methods, the classification accuracy of $\text{LPP}_{\text{k=10}}$ is lowest and varies considerably with the number of extracted features. When the number of extracted features is 220, the accuracy of the $\text{LPP}_{\text{k=3}}$ achieves the same classification accuracy as that of 1-NN (where all the features of the data set is used in 1-NN). The accuracy of LPP_{ANS} exceeds that of 1-NN only with the mere of 5 features, and that of PCA with the mere of 15 features.

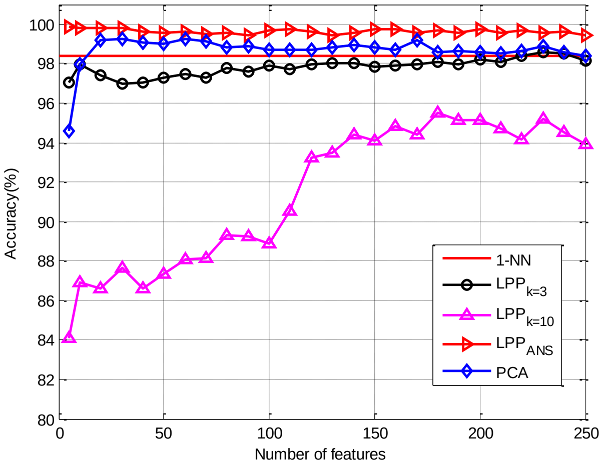


Fig. 1. Classification accuracy versus the number of extracted features on COIL20 data set

4.3 Clustering

In this experiment, we employ the K-means clustering method (KM) to evaluate the clustering performance of the LPP_{ANS} algorithm with different numbers of the extracted features. The performance metrics are the clustering accuracy and normalized mutual information (NMI). Details about these two metrics can be found in [13], [14]. The experiments are conducted repeatedly with different number of clusters K in the range of {5, 10, 20, ..., 60, 68} and with different feature number in the range of {5, 10, 20, ..., 250}. For a given value of K, the experiment is performed following the steps below:

1) Extract the best 250 features (except for K-means clustering in original space, i.e. using all the original features);

2) Randomly select K classes from the data set;

3) Set the feature number;

4) Execute the K-means clustering method for 10 times with different initialization settings and record the best results;

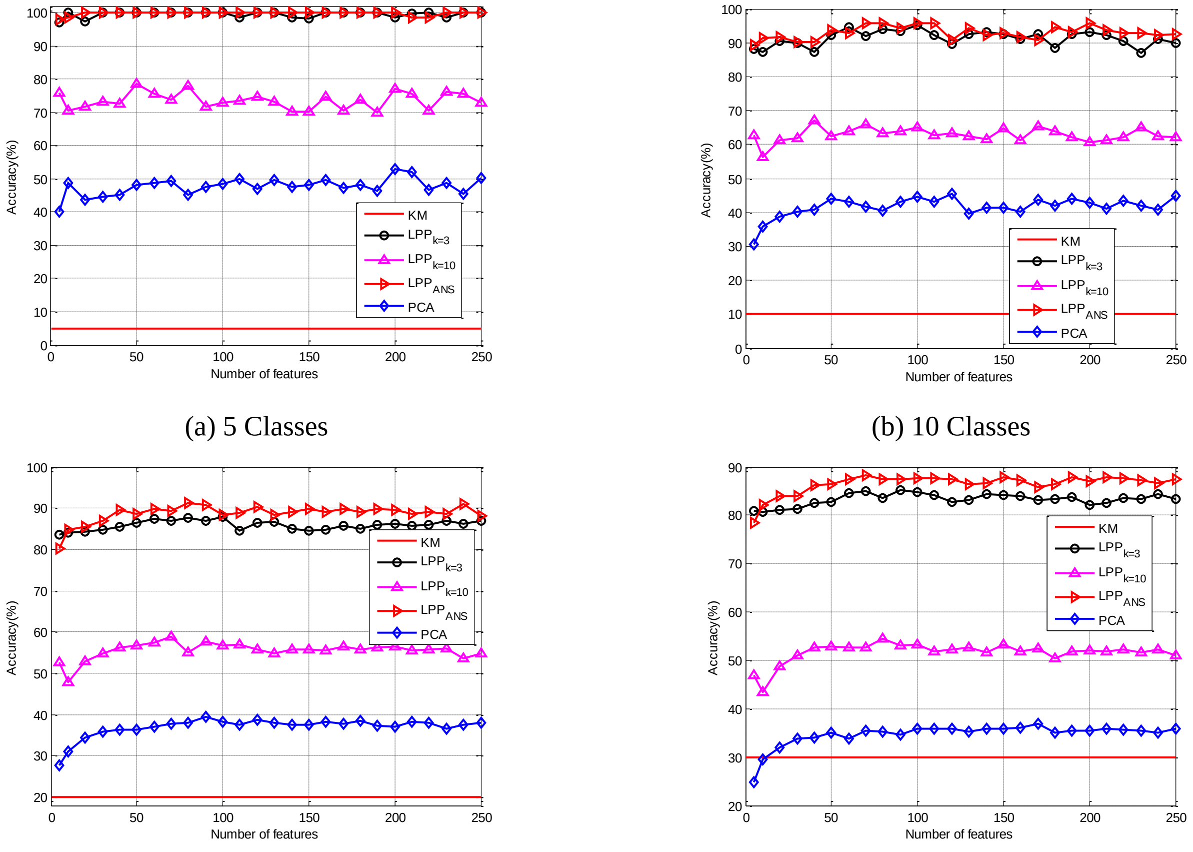
5) Repeat steps 3) and 4) until all the feature numbers are chosen;

6) Repeat steps 2) ~ 5) for 20 times (except for $K = 68$);

7) Compute the mean and standard error of performance for the given value of K;

8) Change the number of clusters K and repeat steps 2) to 7) until all the values of K are selected.

Figs. 2 and 3 show the clustering performance in terms of clustering accuracy and NMI respectively. The experimental results that the three LPP algorithms, including LPP$_{k=3}$, LPP$_{k=10}$ and LPP$_{ANS}$, achieve better performance that the PCA and KM algorithms. This demonstrates the importance of the invariant of the geometrical structure in feature extraction. Furthermore, LPP$_{ANS}$ has the best performance among the other algorithms, as a result of its ability to select the neighborhood size in an adaptively manner in the construction of the adjacent graph for learning manifolds.

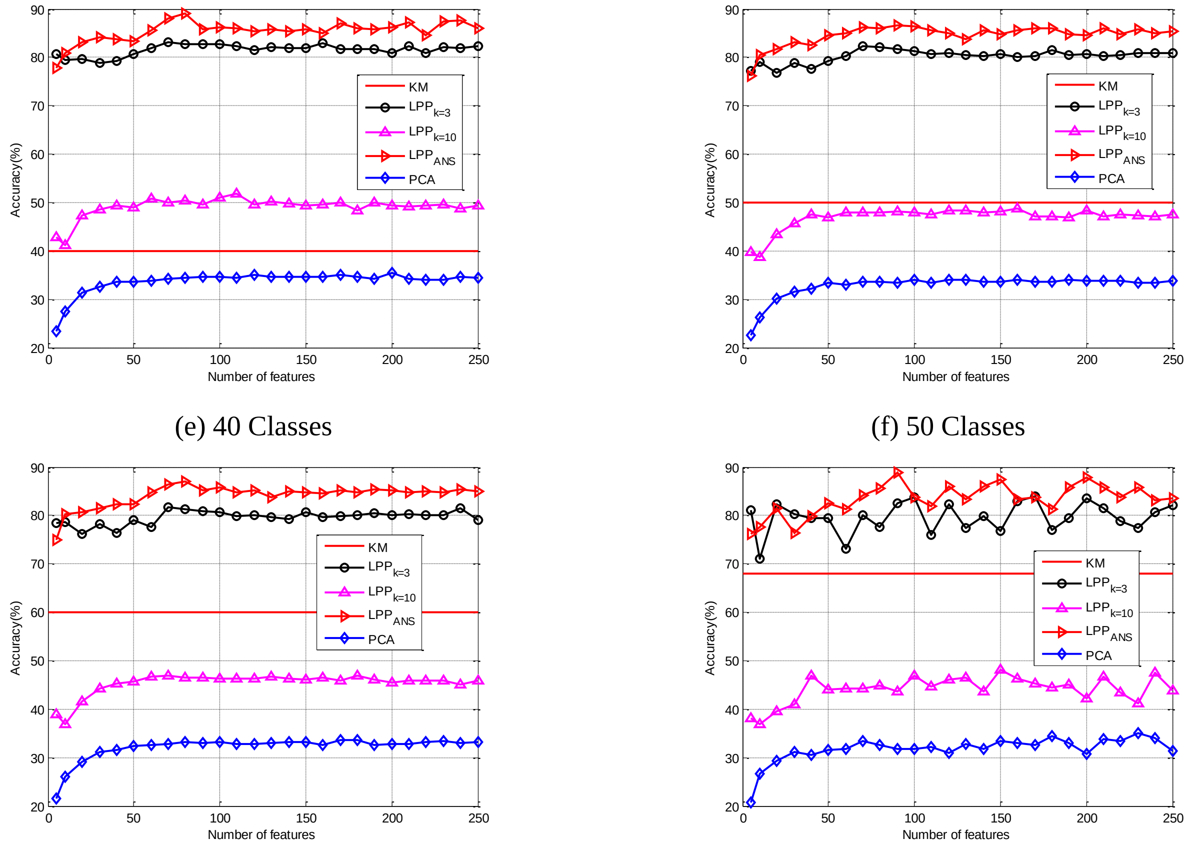


(a) 5 Classes

(b) 10 Classes

(c) 20 Classes

(d) 30 Classes



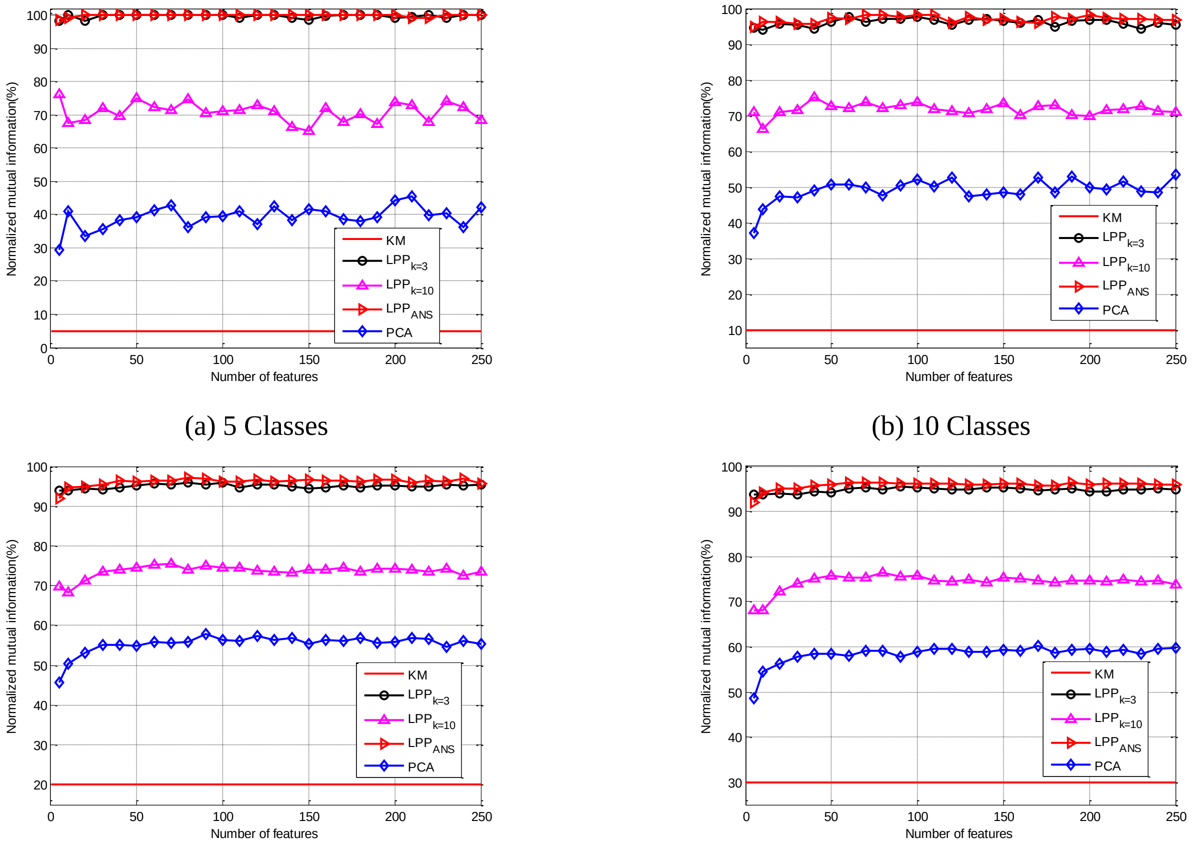
(e) 40 Classes

(f) 50 Classes

(g) 60 Classes

(h) 68 Classes

Fig. 2. Clustering accuracy versus the number of selected features on PIE data set

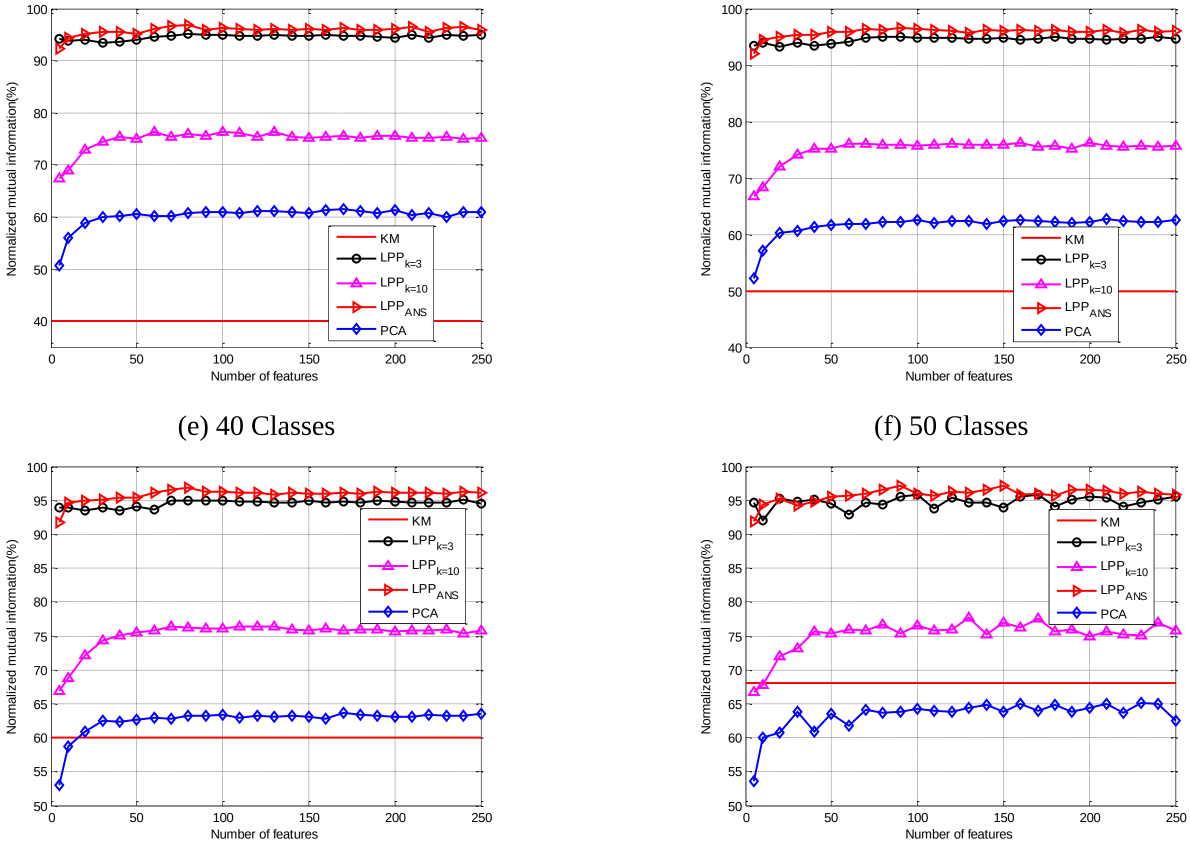


(a) 5 Classes

(b) 10 Classes

(c) 20 Classes

(d) 30 Classes



(e) 40 Classes

(f) 50 Classes

(g) 60 Classes

(h) 68 Classes

Fig. 3. Normalized mutual information versus the number of selected features on PIE data set

5 Conclusion and Future Work

In this paper, we propose two algorithms to adaptively select the neighborhood size in the construction of the adjacent graph in manifold learning. The algorithms are developed based on nearest neighbor estimation and the nearest neighborhood estimation respectively. They are applied to the LPP method to investigate the effectiveness of the algorithms in data classification and clustering. The performance of the LPP$_{ANS}$ algorithm on the PIE and COIL20 data sets is found to be superior to the conventional LLP algorithm and the PCA method. The experiment results suggest that the local structure of the data can be better captured by using the adaptive neighborhood size selection algorithms, and thus the underlying manifold structure can be discovered more effectively by the manifold learning methods. Further research will be carried out to deal two issues of the proposed algorithms. First, during the estimation of the neighbor or the neighborhood, a suitable width parameter σ is required to define the kernel density, but it is not clear about how to select this parameter theoretically and effectively. Besides, the data set sampled from a low-dimensional manifold is usually noisy. It is necessary to investigate the robustness of our proposed algorithm against the noises.

Acknowledgements

This work was supported in part by the Hong Kong Polytechnic University under Grants A-PJ38, 1-ZV6C and 87RF; the National Natural Science Foundation of China under Grants 60903100, 60975027, 90820002; and the Natural Science Foundation of Jiangsu Province under Grant BK2009067.

References

[1] J. Tenenbaum, V. de Silva, J. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319-2323, Dec. 2000.

[2] S. Roweis, L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323-2326, Dec. 2000.

[3] X. He, P. Niyogi, “Locality preserving projections,” *Proceeding in Conference Advances in Neural Information Processing Systems (NIPS)*, 2003.

[4] M. Belkin, P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” *Advances in Neural Information Processing Systems 14*, pp. 585-591, MIT Press, 2001.

[5] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *Journal of Educational Psychology*, vol. 24, pp. 417-441, 1933.

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

[7] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.

[8] C. R. Rao, “The utilization of multiple measurements in problems of biological classification,” *Journal of the Royal Statistical Society. Series B*, vol. 10, pp. 159-203, 1948.

[9] H. S. Seung, D. D. Lee, “The manifold ways of perception,” *Science*, vol. 290, no. 12, pp. 2268-2269, Dec. 2000.

[10] Fan R. K. Chung, Spectral Graph Theory, *Regional Conference Series in Mathematics*, no. 92, 1997.

[11] A. Dempster, N. Laird, D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1-38, 1977.

[12] C. Atkeson, A. Moore, S. Schaal, “Locally weighted learning,” Artificial Intelligence Review, vol. 11, no. 15, pp. 11-73, 1997.

[13] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” *Advances in Neural Information Processing Systems 18*, MIT Press, 2005.

[14] D. Cai, X. He, J. Han, “Document clustering using locality preserving indexing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, pp. 1624-1637, Dec. 2005.