MANIFOLD LEARNING, A PROMISED LAND OR WORK IN PROGRESS?

Mei-Chen Yeh[1], I-Hsiang Lee[2], Gang Wu[1], Yi Wu[1], Edward Y. Chang[1]

Departments of Electrical & Computer Engineering[1] and Computer Science[2], UC Santa Barbara

ABSTRACT

Tasks of image clustering and classification often deal with data of very high dimensions. To alleviate the dimensionality curse, several methods, such as Isomap, LLE and KPCA, have recently been proposed and applied to learn low-dimensional non-linear embedded manifolds in high-dimensional spaces. Unfortunately, the scenarios in which these methods appear to be effective are very contrived. In this work, we empirically examine these methods on a realistic but not-so-difficult dataset. We discuss the promises and limitations of these dimension-reduction schemes.

1. INTRODUCTION

Several dimensionality-reduction algorithms have recently been proposed to find nonlinear manifolds embedded in a high-dimensional space. Among the proposed methods, Isomap [5], local linear embedding (LLE) [1], and kernel PCA (KPCA) [6] have been applied to tasks of image clustering [1, 3-6] and image retrieval [7,8]. However, the scenarios in which manifold learning has been shown to be effective are rather contrived. For instance, the widely used Swiss-roll example [5] is a three-dimensional structure on which data are densely populated. Several examples of face and object images presented in [1-6] change their poses only slightly from one image to another, so manifolds can easily be discovered. In all demonstrated scenarios, noise has not been considered a factor to seriously challenge manifold learning.

Manifold learning faces at least three technical challenges [3]. First, training data must be densely populated in the intrinsic space where a manifold resides. If data are sparsely populated, or if many data instances cannot find neighboring points in a local area, then no manifold can be learned. Second, the presence of noise in a local area may prevent correctly learning the real structure. Third, when the dimension of data is high (typically higher than 30), the *dimensionality curse* aggravates the above two problems. An exponentially large number of instances are required to characterize a manifold in a very high-dimensional space. The problem of noise magnifies when data is sparsely populated, which is inevitable in a high-dimensional space.

In this paper, we report our experiments using a real-world image dataset to examine the effectiveness of Isomap, LLE and KPCA. The *1,897*-image dataset we used consists of *14* image categories. We have used this dataset in several settings, both supervised and unsupervised, and have found it to be relatively "well behaved" compared to many other real-world datasets we have used. We did not use a "harder" database because all dimension-reduction methods would have failed miserably, and we would not be able to observe, identify, and explain the limitations of manifold learning.

The rest of this paper is organized into three sections. Section 2 briefly summarizes Isomap, LLE, and KPCA. Section 3 presents the results of our empirical studies. We offer our observations and concluding remarks in Section 4.

2. MANIFOLD LEARNING

Given a set of high-dimensional training instances $O = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where $\mathbf{x}_i \in \Re^p$. Manifold learning algorithms attempt to find an embedding set $E = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ of O in a low-dimensional space $\Re^d (d < p)$, and the local manifold structure formed by O in the original space $\Re^p$ is preserved in the embedded space $\Re^d$. An underling assumption of these algorithms is that the data are "well distributed" on a manifold (M) of dimension d. In the following, we briefly review three representative manifold learning algorithms, Isomap, LLE, and kernel PCA.

2.1 ISOMAP

Isomap [5] builds on classical multi-dimensional scaling (MDS) by first constructing a squared distance matrix $\mathbf{D} = [d_{ij}]_{i,j=1}^{N}$. Instead of calculating d_{ij} using the Euclidean distance, Isomap uses the *geodesic* distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ along the manifold M where the training points reside in the high-dimensional space $\Re^p$. In [5], the geodesic distance is approximated by finding the shortest path in a weighted graph G with edges of weight d_{ij} connecting neighboring data points on the

manifold M. Next, Isomap applies MDS to the geodesic distance matrix $\mathbf{D}$, embedding the p-dimensional dataset O in a d-dimensional Euclidean space $\Re^d$ that preserves the M's intrinsic geometry. More specifically, let $(\lambda_k, \mathbf{v}_k)_{k=1}^N$ be the eigenmap of the geodesic distance matrix $\mathbf{D}$. Isomap chooses the d largest λ_k with the corresponding eigenvector $\mathbf{v}_k$ and calculates the d-dimensional embedded vector $\mathbf{y}_i$ of the training point $\mathbf{x}_i$ as $\left(\sqrt{\lambda_1}\mathbf{v}_{1i},...,\sqrt{\lambda_k}\mathbf{v}_{ki},...,\sqrt{\lambda_d}\mathbf{v}_{di}\right)^T$.

2.2 LLE

The LLE algorithm [1] seeks an embedding to preserve the local manifold geometry of the neighborhood of each training point. It first constructs a sparse weight matrix $\mathbf{W}$ with its i,j^{th} component w_{ij} representing the construction ability of $\mathbf{x}_j$ on $\mathbf{x}_i$, where $\sum_j w_{ij} = 1$, and w_{ij} equals 0 if $\mathbf{x}_j$ is out of the k-nearest neighbors of $\mathbf{x}_i$. Next, LLE makes an eigendecomposition on the matrix $\mathbf{M} = \left(\mathbf{I} - \mathbf{W}^T\right)\left(\mathbf{I} - \mathbf{W}\right)$ and generates the embedding using $\mathbf{M}$'s bottom $d+1$ eigenvectors $\mathbf{v}_k$'s, corresponding to the $d+1$ smallest eigenvalues λ_k's. LLE discards the bottom $\mathbf{v}_k$ with the zero λ_k. Therefore, $\mathbf{y}_i$ is equal to $\left(v_{2i},...,v_{ki},...,v_{(d+1)i}\right)^T$.

2.3. KPCA

The KPCA algorithm [6] seeks a non-linear dimension reduction in a high-dimensional space. KPCA first maps data $\mathbf{x}_i$ into a high-dimensional Hilbert kernel space as $\phi(\mathbf{x}_i)$ using a positive semi-definite kernel $\mathbf{K}$. Next, KPCA solves an eigen problem on a centered kernel matrix $\tilde{\mathbf{K}} = \left(\mathbf{I} - \mathbf{e}\mathbf{e}^T\right)\mathbf{K}\left(\mathbf{I} - \mathbf{e}\mathbf{e}^T\right)$ where $e = m^{-1/2}(1,...,1)^T$. Finally, the $\mathbf{y}_i$ is calculated as $\left(\sum_{l=1}^N \mathbf{v}_{1l}k(\mathbf{x}_i,\mathbf{x}_l),...,\sum_{l=1}^N \mathbf{v}_{kl}k(\mathbf{x}_i,\mathbf{x}_l),...,\sum_{l=1}^N \mathbf{v}_{dl}k(\mathbf{x}_i,\mathbf{x}_l)\right)^T$

Although KPCA does not obviously consider the local manifold geometry in the algorithm, it can be related to Isomap and LLE in a kernel framework. For example, [4] argues that by taking the following "kernel"

$$\mathbf{K}_{isomap} = -\frac{1}{2}\left(\mathbf{I} - \mathbf{e}\mathbf{e}^T\right)\mathbf{D}\left(\mathbf{I} - \mathbf{e}\mathbf{e}\right),$$

the final embedding found by ISOMAP using $\mathbf{K}_{isomap}$ is identical (up to some scaling) to the projections of KPCA using the kernel. For LLE, by using the kernel $\mathbf{K}_{lle} = \lambda_{\max}\mathbf{I} - \mathbf{M}$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{M}$, the coordinates of the leading eigenvectors of KPCA performed on $\mathbf{K}_{lle}$ yield the LLE embedding.

3. EXPERIMENTAL RESULTS

We performed several experiments to examine if preprocessing data with Isomap, LLE, or KPCA can improve clustering performance. We employed k-means as our clustering algorithm. To conduct our experiments, we used a 14-category 1,897-image dataset, with each image being represented by a 144-dimensional feature vector [9]. We first applied k-means to the raw data to record the percentage of data that can be clustered into their correct categories. We obtained clustering accuracy of 87.36% (or 0.8736). This is the yardstick performance to which Isomap, LLE, and KPCA were compared.

3.1. Clustering Accuracy

Figure 1. Isomap Clustering Accuracy.

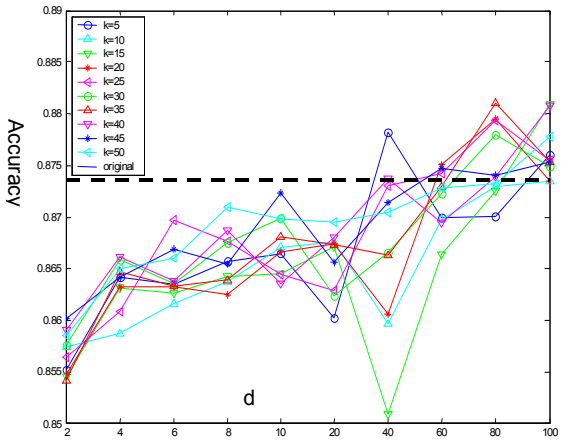

Figure 1 presents the clustering accuracy using Isomap with different values of k (number of nearest neighbors) and d (intrinsic data dimension, on the x-axis). When d is reduced from *144* to between *60* and *100* and k is set between *20* and *40*, preprocessing data with Isomap shows improvement in clustering accuracy. However, the

improvement is less than 1% even in the best case, a very insignificant amount.

Figure 2 presents the results when using LLE. The best result was obtained when *d* was set to *2* and *k* between *40* and *80*. Again, the accuracy improvement is insignificant.

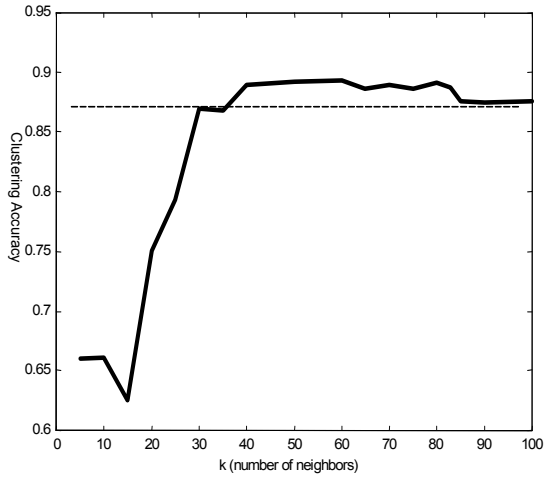

Figure 2. Clustering accuracy with LLE (d = 2).

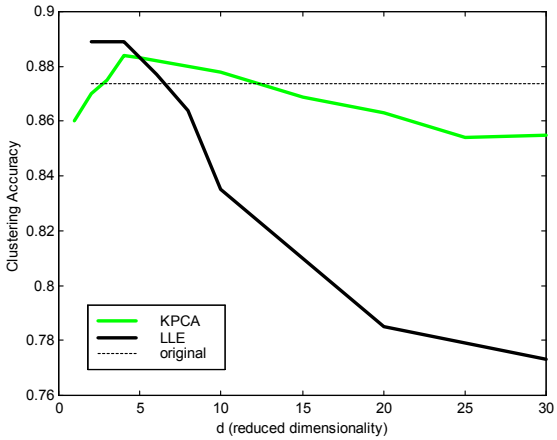

Figure 3. Clustering Accuracy using KPCA and LLE.

Figure 3 shows the clustering accuracy after using KPCA for manifold learning. The light-color curve shows the KPCA clustering accuracy on different *d* settings. When *d* is *4* or *5*, KPCA slightly outperforms the yardstick accuracy. KPCA and LLE appear to obtain similar intrinsic dimensions (*2-5*), whereas Isomap works better when *d* is between *20* and *40*.

3.2. Promising Potential

Our experimental results show that manifold learning can be helpful in two ways. First, it can (although in a very insignificant way on our dataset) improve data quality for supervised and unsupervised learning tasks. Second, if the discovered intrinsic structure is in low dimensions, one can visualize the data and gain useful insights. Figure 4 plots the image-dataset in a 3-d space after LLE has been applied. From the plot, we can identify "trouble-spot" classes or clusters, and this information can be useful for designing new features to improve cluster separation.

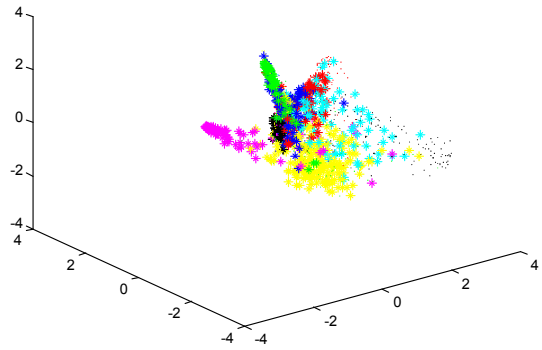

Figure 4. Image Data in a 3-d Space After LLE.

3.3. Limitations

Unfortunately, several limitations hinder manifold learning from being practical. In addition to the high data-density requirement, sensitivity to noise, and curse of dimensionality that we discussed in Section 1, we observed during our empirical study one major *chicken-and-egg* problem: without knowing the structure of the data, turning parameters (such as *d* and *k* for LLE) is merely shooting in the dark. When conducting our experiments, since we had the labels of the images, we could measure the effectiveness of the learned manifolds and identify the best parameter combinations. However, in a realistic setting when little or no ground-truth is given, there is no way of knowing what parameter setting might yield improved results. More specifically, let us use Figures 1-3 to explain. Without prior knowledge of the image categories, we would not have been able to evaluate clustering accuracy. Thus, we cannot predict whether a parameter combination would be helpful or counter-productive.

At first glance, it would seem that combining supervised learning with manifold learning might alleviate this parameter-setting problem. Our experimental results, however, do not show that this path can work effectively. We randomly set aside 50% of the data as training data. We intended to learn the best parameter setting(s) using the training data, and then apply that setting(s) to the

entire dataset to learn manifolds. We conducted this training experiment for both Isomap and LLE with four sets of randomly sampled datasets. Unfortunately, from the four different training datasets, we obtained vastly different parameter settings. Figure 5 plots the distribution of the "good" settings that can yield improved clustering accuracy for Isomap. Notice that the "good" parameters obtained with different training datasets are all quite different; moreover, the parameters learned do not correlate with the best parameters for the entire dataset, as shown in Figure 1. Similarly, Figure 6 plots the same poor parameter pattern we obtained from the LLE training experiment. We believe that since manifold learning is very sensitive to data distribution, even slightly different sets of data can lead to very different manifold structures. Both the high training variance and the inability to generalize the training result to unseen data render this semi-supervised path unusable or unhelpful.

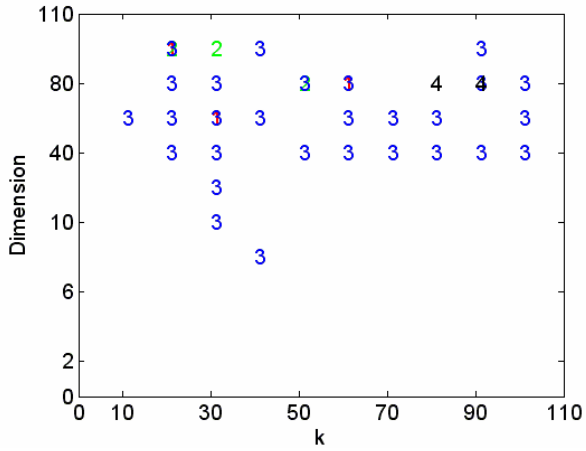

Figure 5. Parameter Settings for ISOMAP.

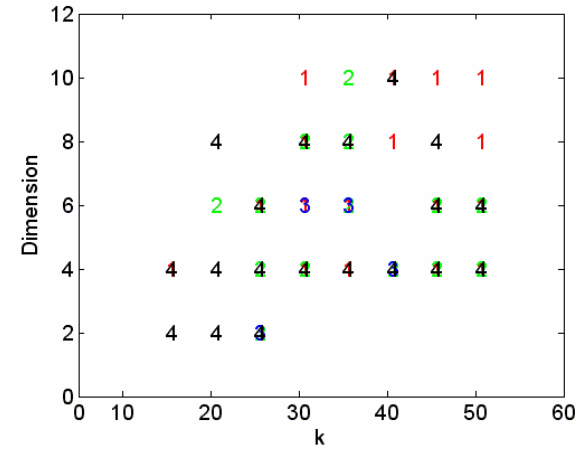

Figure 6. Parameter Settings for LLE.

4. CONCLUSIONS

In this paper, we have explained how we performed an empirical study on the representative manifold learning methods Isomap, LLE, and KPCA. Our results show that when the data is "well behaved" and we have an oracle to provide us with good parameter settings (k and d), then manifold learning can improve clustering accuracy somewhat. However, we concur with [3] that when a dataset is noisy and high dimensional, manifold learning is generally ineffective. Furthermore, we found a practical *chicken-and-egg* problem -- that it is impossible to obtain good parameter settings for manifold learning without prior knowledge of the data characteristics. In addition, we are not fully convinced that manifolds learned on a visible set of data can be generalized to unseen data. Despite some recent claims of success in image retrieval [7-8], we remain skeptical about the practical use of manifold learning at our current level of knowledge. In a recent IPAM meeting [10], which two authors of this paper and the inventors of Isomap and LLE attended, the consensus of the presenters and participants was that manifold learning remains a work-in-progress area for research.

5. REFERENCES

[1] L. K. Saul and S. T. Roweis, "*Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds*," Journal of Machine Learning Research, vol. 4, 2003.

[2] Viren Jain and Lawrence L. Saul, "*Exploratory analysis and visualization of speech and music by locally linear embedding*," Proc. ICASS, vol. 3, 2004.

[3] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Quimet, "*Out-of Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering*," Neural Information and Processing System (NIPS), 2004.

[4] J. Ham, D. D. Lee, S. Mika, and B. Scholkopf, "*A Kernel View of Dimensionality Reduction of Manifolds*," International Conference on Machine Learning (ICML), 2004.

[5] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "*A Global Geometric Framework for Nonlinear Dimensionality Reduction*," Science, 290(5500): 2319-2323, 2000.

[6] B. Scholkopf, A. J. Smola, and K. R. M ü ller, "*Nonlinear Component Analysis as a Kernel Eigenvalue Problem*," Neural Computation (NIPS), vol. 10, 1998.

[7] X. He, "*Incremental Semi-supervised Subspace Learning for Image Retrieval*," ACM Multimedia, pp. 2-8, 2004.

[8] X. He, W.-Y. Mar, H.-J. Zhang, "*Learning an Image Manifold for Retrieval*," ACM Multimedia Conf., 2004.

[9] S. Tong and E. Chang, Support Vector Machine Active Learning for Image Retrieval, ACM Multimedia, 2001.

[10] IPAM High-dimensional Data Workshop, September 2004.