A Local Smoothing and Geodesic Distance Based Clustering Algorithm for High Dimensional Noisy Data; Utilizing Embedded Geometric Structures

Xiaoming Huo *

March 18, 2003

Abstract

A method that utilizes embedded geometric structure is proposed, to enhance clustering. The geometric structure is identified via a combination of a local smoothing method – local linear projection (LLP) – and a computational method for the geodesic distance. It is found that comparing to existing algorithms, it is more efficient to deal with noisy and "structured" data. Simulations for both synthetic data and microarray data are reported. We list some ideas for future improvement.

1 Introduction

In point cloud data, the geometry that is embedded is of unignorable importance. To illustrate the importance of utilizing geometric information, let us consider a data example that is illustrated in Figure 1. In Figure 1, the point cloud has two components; each of them surrounds a helix. In this case, two data points could be in the same system (same helix), but are far away in Euclidean distance, e.g. the points that are at the two ends of a helix. Ideally, a clustering algorithm will assign the points, which are close to the same helix, to the same cluster. Apparently, any method that ignores the geometric structure can hardly achieve this goal.

We propose a method that has the following ingredients. First of all, a local smoothing method is applied to the data, so that the points are pressed towards the geometric structure underlying. This can be considered as a *denoising* step. We choose the method: Local Linear Projection (LLP) [10]. In the second step, the geodesic distance is computed for any two points. The idea is that if there are two points that are *not* in the same geometric structure, the geodesic distance between them is infinite (extremely large). We actually develop an equivalent and fast method to realize the above idea. Our method is similar to the agglomerative approach that has been used in hierarchical clustering, e.g. see [9].

Numerical simulations show that this approach can effectively cluster observations that has embedded non-trivial geometric structure. We also report our experiment of applying this method to a microarray data.

*This work is partially supported by NSF DMS 0140587. Address: School of Industrial & System Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205. Email: xiaoming@isye.gatech.edu. Phone: 404 385 0354 (office). Fax: 404 894 2301.

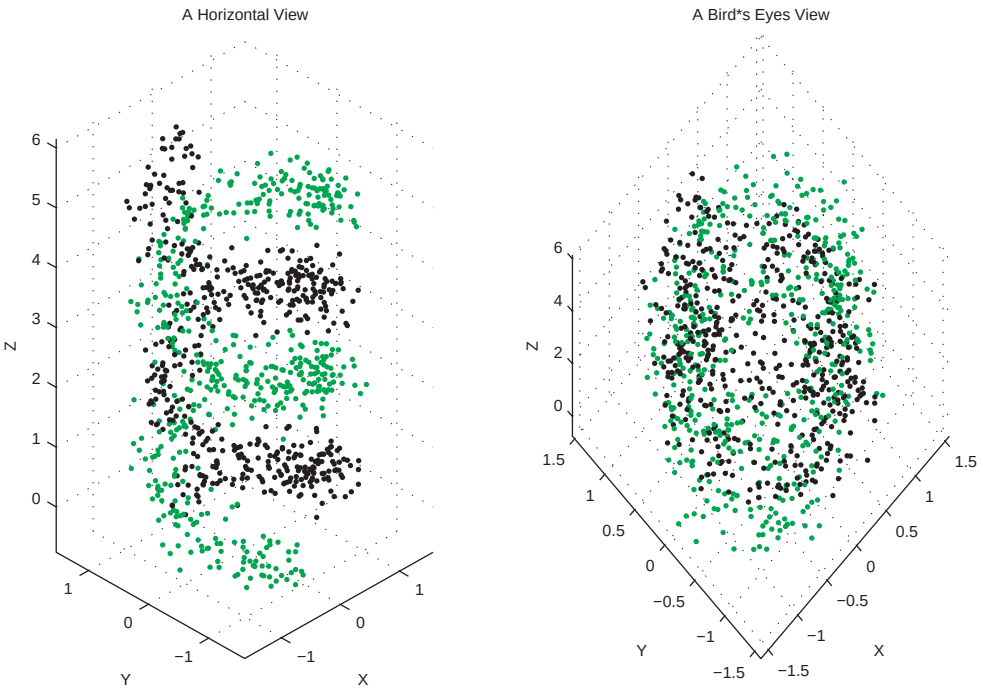

Figure 1: An artificial data, showing the need of incorporating geometric structures. The data is colored to emphasize the two helix structure that is embedded in this data.

The rest of Section Introduction is organized as follows. In Section 1.1, we review some works on computation with respect to embedded geometry. In Section 1.2, we review some methods that can globally extract subspaces. In Section 1.3, we review methods that intend to explore local geometry. Our first step – LLP – is motivated by this line of research. In Section 1.4, we address some potential areas for application. In Section 1.5, we summarize the content of the rest of the paper.

1.1 Utilizing embedded geometric structure

Many methods have been developed to render geometric objects. A fundamental question that is answered by all these methods is to compute the geodesic distance. A recent work is an approach by Mémoli and Sapiro [16]. Some numerical algorithms that have driven their method is developed in [31] and [25]. When the data is not very noisy, e.g. scanned data, efficient methods are available to recover geometry from point cloud data. When the data is very noisy, it is not clear what an efficient approach should be.

At the same time, in mathematically analysis, researchers have developed tools to 'detect' the geometric features. An example of these works include the Jone's function developed by Professor Peter Jones in solving the travelling salesman problem, referring to [11]. More specific tools are developed in some papers, e.g. [14] and [5].

We consider a simpler problem: how to incorporate the geometric information in clustering. It turns out that simple methods are available for solving the clustering problem.

1.2 Methods identifying global structures

Dimensionality reduction plays a significant role in *exploratory data analysis* (EDA). In many real applications, although the data may have very high dimensions, they typically embedded in manifolds (or subspaces) that are of substantially lower dimensions. Identifying these manifolds (or subspaces) are critical in understanding these data. It is also important in applications such as data visualization and modelling. In the communities of statistics, machine learning, and artificial intelligence, a substantial amount of techniques have been developed. In the following, we will give a quick review on works that are directly related to ours.

When the embedded structures are linear subspaces, linear techniques such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) can be used to identify the embedded linear subspaces. In PCA, the second order statistics (variances and covariances) of the data are considered, researchers find the directions in which the variances are maximized. SVD works on the data themselves. It finds the linear subspace that best preserves the information of the data. For both PCA and SVD, the embedded structure must be *globally* linear. In many applications, this condition is too restrictive. Multi-Dimensional Scaling (especially metric MDS) is close to PCA and SVD. PCA and SVD are to find the most significant linear subspaces. In Metric MDS, workers try to map the data into a low-dimensional space, at the same time keeping the inter-data distances [32]. Although the philosophical points are seemingly different, the underlying linear algebra are very similar.

1.3 Methods for local geometry

When the global linearity condition is abandoned, some methods that focused on finding local embedded structures have been proposed, among them, we have for example principal curves [26, 8]. Recently, we have paid attention to some methods that are dedicated to identifying local hidden manifolds, for example, ISOMAP [29] and Local Linear Embedding (LLE) [22]. In ISOMAP, instead of consider the distance between two data points, they consider the geodesic distance, which is the length of the shortest path that resides on the embedded manifold. In implementations, this idea is realized by considering the k-nearest neighbors. Later on, in order to achieve better numerical performance, researchers have proposed some numerical variations, e.g. Curvilinear Distance Analysis (CDA), [13]. In LLE, each data point is represented as a convex combination of its k-nearest neighbors; the data is then mapped into a low-D space, at the same time, the convex combinations (which is called embedding) is preserved to the best possibility. In [13, 29, 22], good examples are shown to illustrate these ideas. These examples are Swiss rolls, open boxes, and cylinders. We found them very instructive.

1.4 Applications

Due to the maturization of the human Genome project and the availability of the microarray technology, microarray data poses a new challenge to data analysts. The microarray technology allows workers to measure the levels of gene expression for tens and thousands of genes simultaneously. The dimensionality of microarray data is definitely high. It is urgent to develop efficient dimension reduction tools. As a matter of fact, many previously mentioned tools have been applied to microarray data, for example, researchers have used SVD to interpolate missing values in a microarray data [30]. ISOMAP has been used to understand the structure of a microarray

data [28]. PCA has been used to summarize microarray experiments [21]. A lot more examples can be found in the references of [20].

As an evidence to illustrate the importance of dimension reduction for microarray data, let us consider the clustering of genes. Clustering genes is to group together the genes that might be associated with identical functionalities. A nice survey on clustering methods for microarray datasets is given in [20]. An associated software is described in [27]. Many studies have been reported, e.g. [7]. Due to space, we can not enumerate all of them here. Dimension reduction can help improving the clustering result. One first project the data points to an embedded low-dimensional manifold, then compute the inter-distances between projections. The inter-distances should be more "faithful" than the inter-distance computed directly from the data. Hence a dimension reduction tool can be used as a preprocessing tool for a clustering algorithm.

A dimension reduction tool can also help to visualize the data. To visualize the data, we have to reduce the global dimensionality of the data. This is a little bit different from reducing the local dimensionality of a data. But by appending a post-processing method, it can be used to visualize the data. For example, we can look at the *local* structure of the data. In our simulation study to a synthetic data,a demo of this idea is given.

1.5 Main Idea

We consider point cloud data. We assume that the data is randomly sampled around an underlying geometric object (e.g. a 1-D curve in a high dimensional space, or a manifold). We further assume that the local geometry can be approximated well by some dimension reduction method. Moreover, the underlying structure is connected and smooth enough so that they can be concatenated locally to form a global geometric object.

Our proposed method has two key steps. In the first step, one applies LLP to denoise the data. In the second step, an agglomerative method is used to implement the clustering.

The rest of the paper is organized as follows. In Section 2, the statistical model for embedded manifolds is described. In Section 3, we describe our algorithm. In Section 4, some parameter estimation strategies are presented. In Section 5, we report simulational findings for both a synthetic data and a microarray data. In Section 6, some related works are discussed. In Section 7, questions that will be further analyzed are listed. We provide some final remarks.

2 Model

We assume an additive noise model. Suppose there are N observations, which are denoted by $y_1, y_2, \ldots, y_N$. Let p denote the dimension of each observation. We have $y_i \in \mathcal{R}^p, \forall 1 \leq i \leq N$. We assume that there is an underlying (piecewise smooth) function $f(\cdot)$ such that

$$y_i = f(x_i) + \varepsilon_i, \quad i = 1, 2, \ldots, N,$$

where variable $x_i \in \mathcal{R}^{p_0}$ is from a much lower dimensional space ($p_0 \ll p$), noises ε_i's follow a multivariate normal distribution ($\varepsilon_i \sim N(\vec{0}, \sigma^2 I_p)$, where σ is unknown).

In the above model, if the underlying function f is locally regular, or more specifically, function f can be approximated by a linear function:

$$f(x) \approx \beta_0 + \beta_1^T x,$$

where $\beta_0 \in \mathcal{R}^p$ and $\beta_0 \in \mathcal{R}^{p \times p_0}$, then locally linear projection can be applied to extract this information.

3 Algorithm

We describe the LLP in Section 3.1 and the clustering algorithm in Section 3.2.

3.1 Local Linear Projection

LLP can be applied to extract the local low-dimensional structure. In the first step, neighboring observations are identified. In the second step, SVD or PCA is used to estimate the local linear subspace. Finally, the observation is projected into this subspace.

ALGORITHM: LLP

for each observation $y_i, i = 1, 2, 3, \ldots, N$,

1. Find the K-nearest neighbors of y_i. The neighboring points are denoted by $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_K$.
2. Use PCA or SVD to identify the linear subspace that contains most of the information on vectors $\tilde{y}_1$, $\tilde{y}_2$, ..., $\tilde{y}_K$. Suppose the linear subspace is $\mathcal{A}_i$, and $P_{\mathcal{A}_i}(x)$ denote the projection of a vector x into this subspace. Let k_0 denote the assumed dimension of the embedded manifold, then subspace $\mathcal{A}_i$ can be viewed as a linear subspace spanned by the vectors associated with the first k_0 singular values.
3. Project y_i into the linear subspace $\mathcal{A}_i$ and let $\widehat{y_i}$ denote this projection: $\hat{y}_i = P_{\mathcal{A}_i}(y_i)$.

end.

The output of LLP, $\hat{y}_i$, $i = 1, 2, \ldots, N$, are more "faithful" to the underlying structure (if it exists) than the original observations are.

A justification to step 2. is that based on the previous model, for $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_K$, we have

$$\begin{aligned} \tilde{y}_1 &= \beta_0 + \beta_1^T \tilde{x}_1 + \tilde{\varepsilon}_1, \\ \tilde{y}_2 &= \beta_0 + \beta_1^T \tilde{x}_2 + \tilde{\varepsilon}_2, \\ &\vdots \\ \tilde{y}_K &= \beta_0 + \beta_1^T \tilde{x}_K + \tilde{\varepsilon}_K, \end{aligned}$$

where $\tilde{\varepsilon}_i \sim N(\vec{0}, \sigma^2 I_p)$, and $\tilde{x}_i \in \mathcal{R}^{p_0}$, $i = 1, 2, \ldots, K$. Hence $\tilde{y}_i$'s can be viewed as random vectors whose mean vectors are from a low-dimensional subspace. The low-dimensional subspace can be extracted via SVD of vectors $\tilde{y}_1$, $\tilde{y}_2$, ..., $\tilde{y}_K$. The dimension of this linear subspace can be estimated by analyzing the variances.

The computational complexity of LLP is roughly $C(p, k_0, K)N^2$, where constant $C(p, k_0, K)$ is a function of the dimension of the data (p), the dimension of the embedded linear subspace (k_0), and the number of nearest neighbors (K). The reasoning is as follows. First of all, to identify the

nearest neighbors, the distance matrix of the N observations need to be computed, which costs $O(pN^2)$ operations. Then each row (or column) of the distance matrix need to be sorted, which costs $O(N\log(N))$ (order of complexity for the quick sort algorithm) multiply with N (number of rows) operations. Or in other words, the sorting takes $O(N^2\log(N))$ operations. Suppose that each SVD step takes a constant amount of operations $C_2(p, k_0, K)$, so does each projection step. Overall, the order of complexity for LLP is $C(p, k_0, K)N^2$.

Note that the $O(N^2)$ complexity may inhibit the application of this method in large datasets. Many people have studied the computation of nearest neighbors. An introduction to these activities will be provided in Section 3.3. In summary, the complexity of our algorithm will be much better than $O(N^2)$.

3.2 Clustering

After computing the projections ($\hat{y}_i$'s), the inter point distance

$$\rho(i,j) = \|\hat{y}_i - \hat{y}_j\|_2, \qquad 1 \le i < j \le N,$$

can be computed. We treat distance $\rho(i,j)$ as an approximate to the geodesic distance between observations i and j. A bottom-up hierarchical clustering algorithm can be applied. Our algorithm has the following components:

- At every step, two points (sets of points, or a point and a set) that has the minimum distance is merged as one set.
- The distance between two sets, or a point and a set, is the minimum distance between two points in different sets.
- The algorithm is terminated when all the points are in one set.

This is an bottom-up clustering approach.

To determine a reasonable number of clusters. One can examine the distances at a few final merging steps. When the distance between two sets is above a prescribed threshold, the merging will be stopped.

In our simulations, we found that very likely the last few final steps are dominated by the *outliers* in the data. Hence by monitoring the numbers of points in each final clusters, one can determine which are outliers. For example, if a cluster in a final stage contains only one point, then this point is very likely to be an outlier.

3.3 Computing Nearest Neighbors

Computing nearest neighbors has been a long standing problem. A recent work [15] provides some literature pointer on this topic. In summary, by relaxing some conditions on the definition of the nearest neighbors, the order of computational complexity can be reduced to $O(N\log(N))$. Consequently, the order of complexity of the LLP will be reduced to $O(N\log(N))$ as well. The work in [2] computes the *$(1+\epsilon)$ approximate nearest neighbor* in $O((p/\epsilon)^p \cdot N\log(N))$ time and by using $O(pN)$ storage. Recall that p is the dimension of the space where the point cloud resides. Most of these works depend on a hierarchical decomposition of the state space, as a preprocessor.

For more details on algorithms, we refer to [18] and [2]. Implementations are available at [17] and in [18].

The numerical implementation of the above ideas in our proposed framework will be done in the future.

4 Model Parameter Estimation

There are two key parameters in LLP. They are the number of nearest neighbors (K) and the dimension of the local underlying subspace (k_0). The ideal number for K is the one such that the linearity assumption holds. For the dimensionality parameter k_0, it is ideal to have $k_0 = p_0$.

4.1 Number of Nearest Neighbors

Following the notations in Section 3.1, for a fixed data point y_i and its K-nearest neighbors $\tilde{y}_j$, $j = 1, 2, \ldots, K$, if the linearization model is true, the squared distances

$$d_{i,j} = \|y_i - \tilde{y}_j\|_2^2, \quad j = 1, 2, \ldots, K,$$

should approximately follow the $2\sigma^2 \cdot \chi_p^2$ distribution. These distances can be ordered:

$$d_{i,(1)} < d_{i,(2)} < \cdots < d_{i,(K)}.$$

If we calculate the differences

$$d_{i,(j+1)} - d_{i,(j)}, \quad j = 1, 2, \ldots, K-1,$$

we are going to observe a few big ones at the beginning, and then it decreases to small ones. This is because for χ^2-distributed random variables, the sequence of the differences of the order statistics is going to have the above mentioned pattern. The decreasing pattern of the differences can help to identify the appropriate number of nearest neighbors.

One can utilize the QQ-plot to examine the goodness-of-fit of K-nearest neighbors to the χ^2-distribution. More theoretically sophisticated approaches are available. For example, paper [12] gives some good references on the methods that are currently available. For readers who are more into the statistical analysis, we recommend the papers [23, 24]. Another related work on this topic is in [3]. The paper [19] creates a result on the asymptotic distribution of the statistics that are based on the nearest neighbors. This work is helpful in understanding the statistical behavior of the nearest neighbors, which consequently will help to design a good testing statistic.

4.2 Dimension of the linear subspace

Still following the notations in Section 3.1, if a fatter version of the matrix β_1 is fixed, (in our case, it is computed from SVD,) then the analysis of the appropriate dimension of the linear subspace (p_0) falls into the domain of Analysis of Variance (ANOVA). In our case, the analysis is more complicated. Since the model matrix is computed from the data as well. Intuitively, as the dimension of the subspaces increases, people would expect a quick dropping of variances at the beginning, and then a relatively steady decreasing.

5 Simulations

Two experiments are reported. The first one is for a synthetic signal. The second one is on a synthetic time series data. The last one is on a microarray data, which is also used in [9].

5.1 Synthetic Data: Two Helixes

The point cloud in Figure 1 are random samples from two underlying functions. The underlying functions are

$$\begin{aligned} f_1(t) &= (\sin(4\pi t), \quad \cos(4\pi t), \quad 6t - 0.5), \qquad \text{and} \\ f_2(t) &= (-\sin(4\pi t), \quad -\cos(4\pi t), \quad 6t - 0.5). \end{aligned}$$

The point cloud is intrinsically made by two 1-D curves. For the noisy data, the standard deviation of noises is chosen to be $\sigma = 0.20$. Based on the analysis of the differences between squared distances, we choose the number of the nearest neighbors $K = 40$. The results of applying LLP are shown in Figure 2. We can easily observe two separated curves.

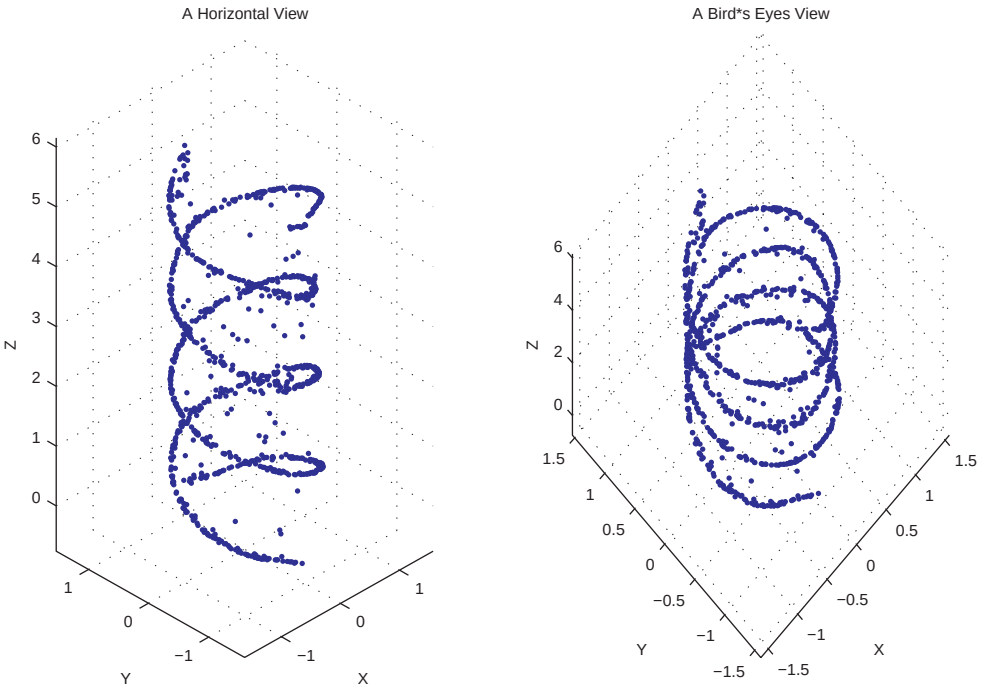

Figure 2: The resulf of applying LLP to the data in Figure 1.

Figure 3 gives a result of the clustering algorithm that is described in Section 3.2. It is found that two significant clusters will be found. In this particular data, about 15 points are claimed to be outliers. Comparing the clustering result with the prior knowledge on where the data points are sampled, the misclassification rate is nearly zero.

More systematic simulation studies will be carried out in a near future.

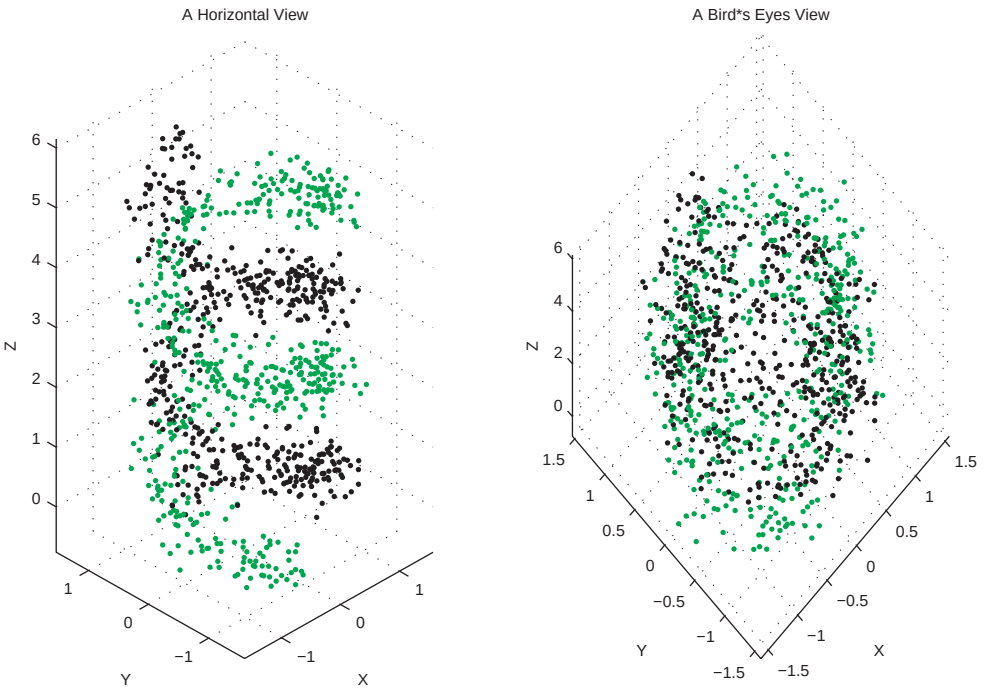

Figure 3: A result of clustering by considering the geodesic distance and local smoothing.

5.2 Time Series Data

To illustrate the application of our method in analyzing spatial data, we give the following example, on time series. A dataset contains $N = 512$ time series. Each series has $p = 64$ dimension. The time series are generated according to the following rule:

$$y_i(t) = \sin\left(\frac{2\pi t}{64} + I(i)\frac{\pi}{2}\right) + \frac{1}{2}\epsilon_{i,t}, \qquad i = 1, 2, \ldots, 512; \quad t = 1, 2, \ldots, 64,$$

where $\epsilon_{i,t} \sim N(0, 1)$ and the function $I(\cdot)$ satisfies

$$I(i) = \begin{cases} 0, & \text{if } 1 \le i \le 128, \quad \text{type-I signal,} \\ 1, & \text{if } 129 \le i \le 256, \quad \text{type-II signal,} \\ 2, & \text{if } 257 \le i \le 384, \quad \text{type-III signal,} \\ 3, & \text{if } 385 \le i \le 512, \quad \text{type-IV signal.} \end{cases}$$

Figure 4 provides an illustration of this set of data. Each plot contains time series belonging to one of the four types above.

The result of LLP based denoising is reported in Figure 5. Note that the information on how the time series are generated are removed prior to applying LLP. One can observe that the LLP recovers the underlying patterns of this set of data well.

Figure 6 illustrate the correctness of the hierarchical clustering, after applying LLP. The coloring of the asteroids indicate the clustering. Compare Figure 6 with the definition of the

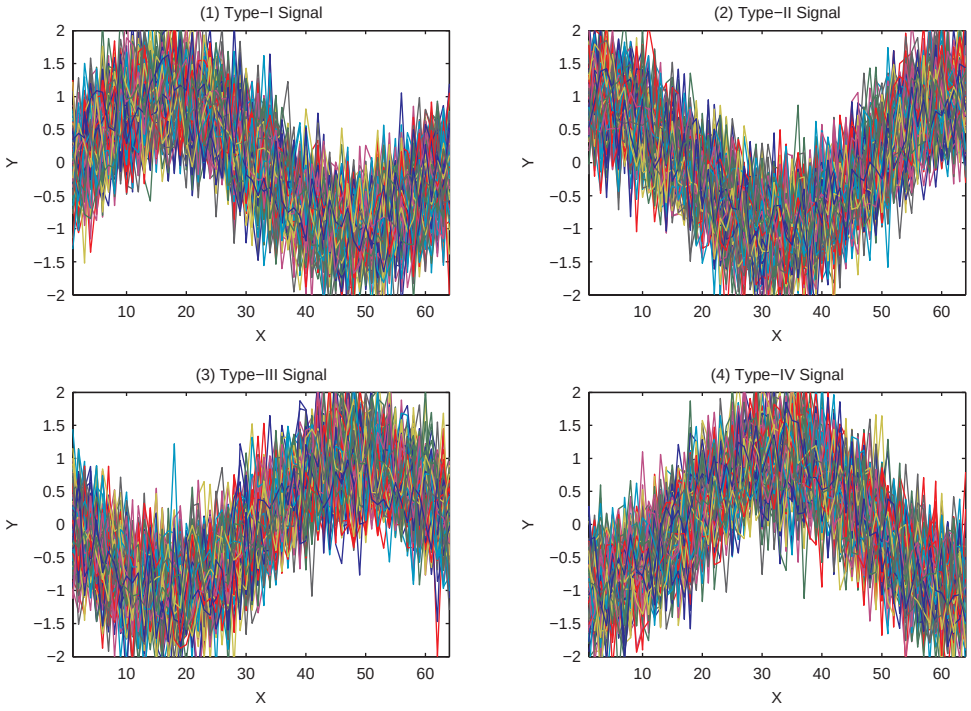

Figure 4: Another set of *noisy* time series data.

function $I(\cdot)$, we can tell that the result of the clustering is consistent with the underlying patterns of the time series.

Based on a previous analysis, we may not be so surprised to observe such a result.

5.3 Microarray Data

Our clustering method is applied to a microarray dataset, which is also used in [9]. (The dataset is downloadable on the web.) We found that the number of the nearest neighbors should be $K = 30$. Due to time limitation, we choose a subset of 1000 genes from the original data. We applied our clustering method to the data. In our simulation, we choose two values – 1 and 15 – for the local dimension, which is k_0 in Section 3.1. The motivation is to examine the effect of smoothing on the results of clustering. The smaller the value of k_0 is, the larger the smoothing effect is. After clustering, we re-plot the microarray data, using the order from the hierarchical clustering. By doing so, we put two 'similar' genes closer to each other.

Figure 7 show the results of the above experiments. See detailed explanation in the caption of the figure. It is found that when after the re-arrangement, the neighboring genes are more similar to each other. On the other hand, the level of smoothing (value of k_0) does have an effect on the re-arrangement: when the local dimension is set low, the re-arrangement appears with less uncertainty; when the local dimension is set high, the re-arrangement becomes more chaotic. We are doing more research on the genetic interpretation of such a phenomenon.

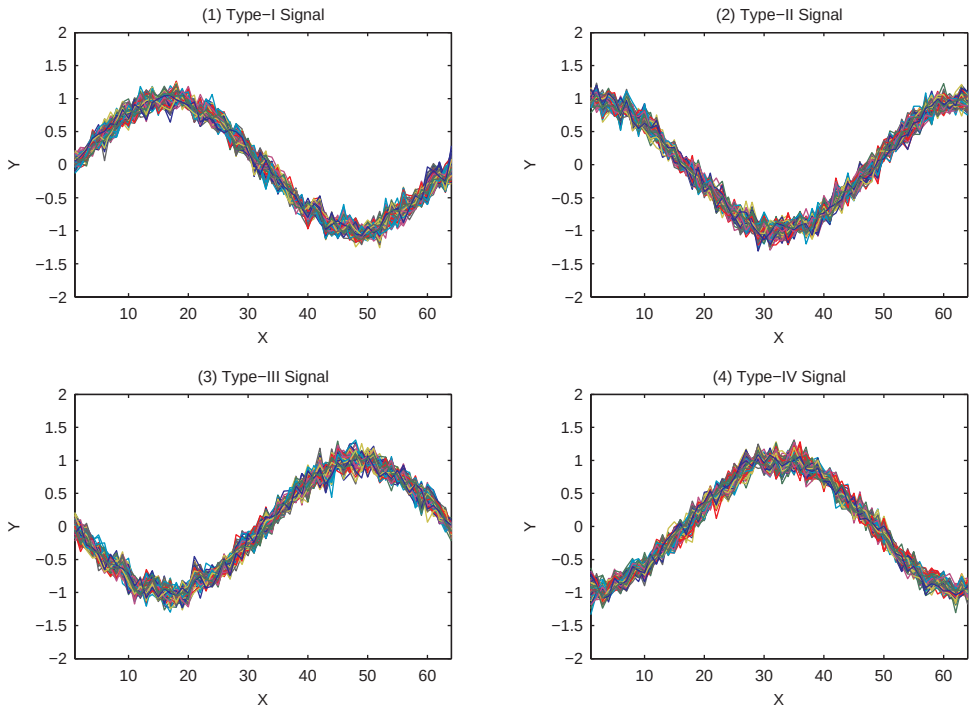

Figure 5: The denoised time series data via LLP.

6 Related Works

Geometric information has been taken into account in clustering. For example, researchers have studied the pattern of sparsity in text data, while carrying out a clustering analysis. Some examples are given in [6]. Note that in our paper, we consider numerical (instead of discrete) data. It will be interesting to explore the connection between the *geodesic distance* idea that is presented here, with the clustering method that is used in [6]. We will leave this as a future work.

The idea of using PCA or SVD in local dimension reduction has been seen in the data mining literature, e.g., [1] and [4]. Their method is named *subspace clustering.* Our method—LLP—is *not* a subspace based method. By *subspace*, the authors of [1, 4] discussed a methodology, which is based on subsets of attributes. In our framework, the underlying structure are *not* necessarily parallel to any axis. Moreover, the geodesic distance has not been taken into account in existing publications that are related to subspace clustering. The absence perhaps is due to the datasets that they have studied: the underlying structure may not be a continuous smooth geometric object.

7 Future Works and Conclusion

LLP has been useful in identifying locally low-dimensional embedded subspaces. It is an optimal dimension reduction tool. We use LLP as a preprocessing tool in a newly proposed clustering algorithm. Geometric information has been utilized. Simulational studies provide some initial

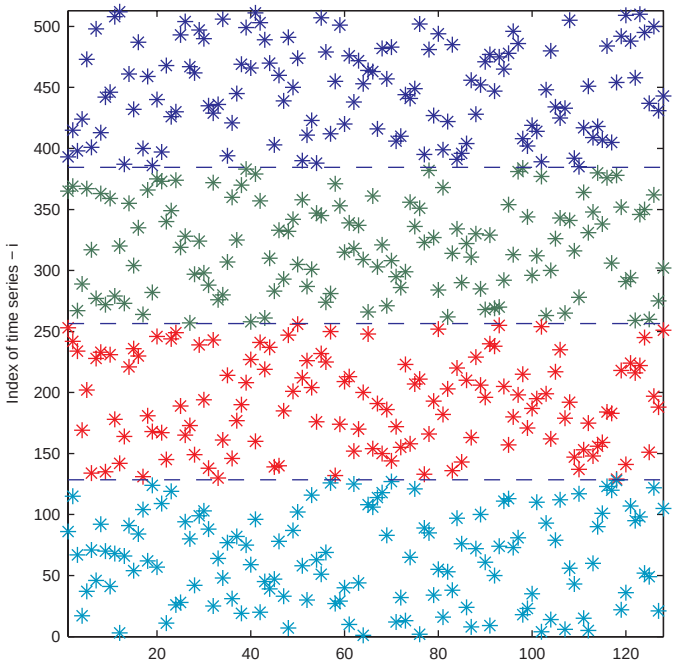

Figure 6: The indices of the last four groups in the hierarchical clustering. Note that it is consistent with the labeling scheme that is applied in our data.

successes.

More systematic simulational studies will be reported in the future. Analytical study of the choice of local dimensionality will be reported as well. We will compare our methods with other approaches.

Acknowledgment

The comments from anonymous referees helped improving the presentation of this paper.

References

[1] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998) Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Eds., Haas, L. M. and Tiwary, A., *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data,* June 2-4, 1998, Seattle, Washington, USA.

[2] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM 45* (6):891-923, November 1998. *Tech. Report CS-TR-3568, Univ. of Maryland, College Park, Dept. of Computer Science,* December 1995. In *Proc. 5th ACM-SIAM Sympos.*, Discrete Algorithms, pages 573–582, 1994.

[3] Bartoszynski R, Pearl DK, Lawrence J. (1997) A multidimensional goodness-of-fit test based on interpoint distances. *Journal of the American Statistical Association, 92*(438): 577-586 June.

[4] Cheng, C.H., Fu, A.W., and Zhang, Y. (1999) Entropy-based Subspace Clustering for Mining Numerical Data. *In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99)*, San Diego, August.

[5] David, G. and Semmes, S. (1993) *Analysis of and on uniformly rectifiable sets*, volume 38 of *Math. Surveys and Monographs*. Amer. Math. Soc.

[6] Dhillon, I. S. and Modha, D. S. (2001) Concept decompositions for large sparse text data using clustering. *Machine Learning, 42*(1-2): 143-175.

[7] Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, *Proc. Natl Acad. Sci. USA*, vol 95: 14863-14868.

[8] Hastie, T., and Stuetzle, W. (1989) Principal Curves, *Journal of the American Statistical Association,* 84 (406): 502-516, June.

[9] Hastie, T., Tibshirani, R., and Friedman, J. H. (2001) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, *Springer series in statistics,* New York.

[10] Huo, X. and Chen, J. (2002). Local Linear Projection (LLP) *First IEEE Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, Raleigh, NC. October 11-13. `http://www.gensips.gatech.edu/proceedings/.`

[11] Jones, P. W. (1990). Rectifiable sets and the traveling salesman problem. *Inventiones Mathematicae*, 102:1–15.

[12] LÉcuyer, P., Cordeau, J.F., and Simard, R. (2000) Close-point spatial tests and their application to random number generators. *Operations Research, 48*(2): 308-317, March-April.

[13] Lee, J.A., Lendasse, A. and Verleysen, M. (2000) Curvilinear distance analysis versus isomap, *submitted to ESANN'02, Bruges.*

[14] Lerman, G. (2000). *Geometric Transcriptions of Sets and Their Applications to Data Analysis.* Ph.D. Thesis, Yale University Department of Mathematics.

[15] Maneewongvatana, S. and Mount, D. M. (2001) On the efficiency of nearest neighbor searching with data clustered in lower dimensions. *International Conference on Computational Sciences (ICCS 2001),* Springer Lecture Notes LNCS 2073: 842-851. Full version: *Univ. of Maryland, Dept. of Computer Science Technical Report CS-TR-4209.*

[16] Mémoli, F. and Sapiro, G. (2003) Distance Functions and Geodesics on Points Clouds. Preprint.

[17] Mount, D. M. ANN: Library for Approximate Nearest Neighbor Searching. `http://www.cs.umd.edu/~mount/ANN/.`

[18] Nene, S.A. and Nayar, S.K. (1997) A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19* (9): 989-1003, September.

[19] Penrose M. D. (2000) Central limit theorems for k-nearest neighbour distances. *Stochastic Processes and their Applications, 85*(2): 295-320, February.

[20] Quackenbush, J. (2001) Computational analysis of microarray data, *Nat Rev Genet*, 6: 418-427.

[21] Raychaudhuri, S., Stuart, J.M. and Altman, R.B. (2000) Principal components analysis to summarize microarray experiments: application to sporulation time series. *Pac. Symp. Biocomput. 2000,* 455-466.

[22] Roweis, S.T. and Saul, L.K. (2000) Nonlinear dimensionality reduction by locally linear embedding, *Science*, vol 290: 2323-2326.

[23] Schilling, Mark F. (1983) Goodness of fit testing in R^m based on the weighted empirical distribution of certain nearest neighbor statistics. *Ann. Statist. 11*(1): 1–12.

[24] Schilling, Mark F. (1983) An infinite-dimensional approximation for nearest neighbor goodness of fit tests. *Ann. Statist. 11*(1): 13–24.

[25] Sethian, J.A. (1996) A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA 93*: 1591-1595, February, category: Applied Mathematics.

[26] Stanford, D.C. and Raftery A.E. (2000) Finding curvilinear features in spatial point patterns: Principal curve clustering with noise, *IEEE Trans. PAMI,* 22 (6): 601-609, June.

[27] Sturn, A., Quackenbush, J. and Trajanoski, Z. (2002) Genesis: cluster analysis of microarray data, *Bioinfomatics*, 207-208.

[28] Tamayo, P. *et al.* (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application, *Proc. Natl Acad. Sci. USA*, vol 96: 2907-2912.

[29] Tenenbaum, J.B., Silva, V., and Langford, J.C. (2000) A global geometric framework for nonlinear dimensionality reduction, *Science*, vol 290: 2319-2323.

[30] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tishirani, R., Bostein, D., and Altman, R.B. (2001) Missing value estimation methods for DNA microarrays, *Bioinformatics,* Vol.17(6): 520-525.

[31] Tsitsiklis, J. N. (1995) Efficient Algorithms for Globally Optimal Trajectories. *IEEE Transactions on Automatic Control, 40* (9): 1528-1538, September.

[32] Young, F. (1981) Introduction to Multidimensional Scaling: Theory, Methods, and Applications, *Academic Press,* New York.

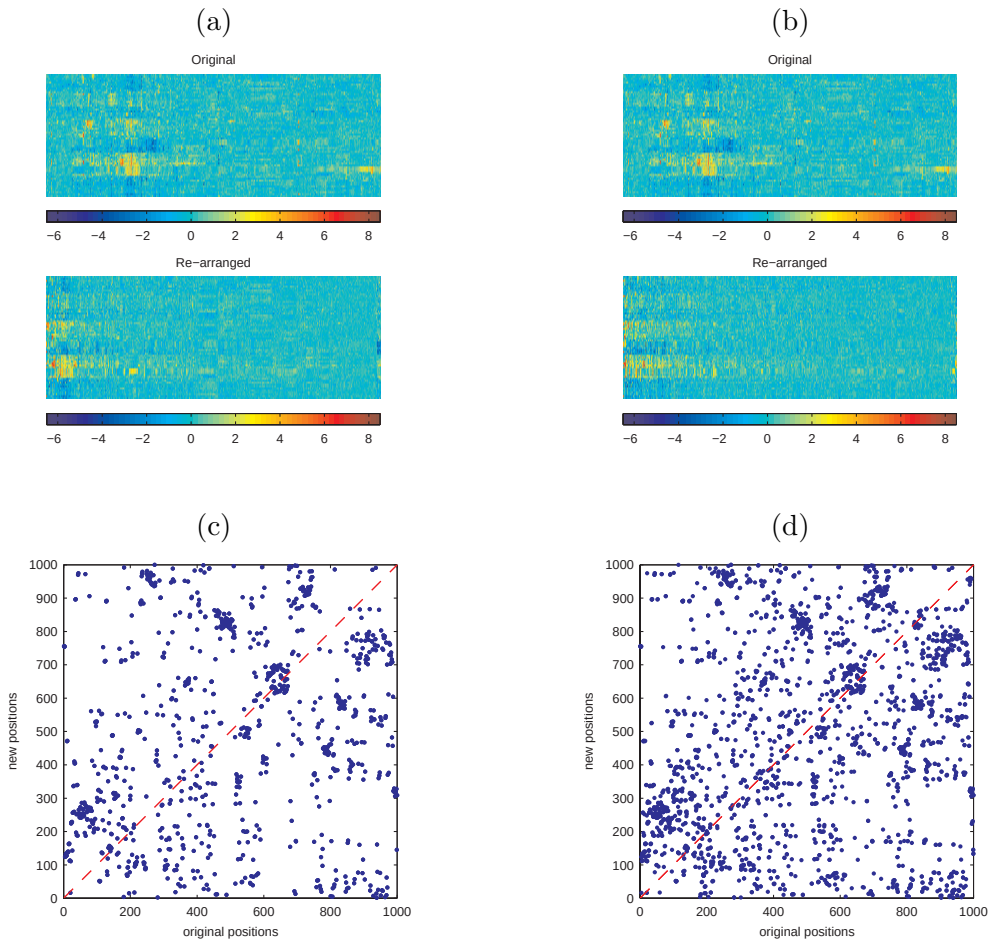

Figure 7: The results of clustering for a microarray data. The left two figures ((a) and (c)) are when the local dimension is 1. The right two figures ((b) and (d)) are for the cases when the local dimension is 15. We found that figure (d) is more 'random' than figure (c). Since the points are more spread out in (d). The figure (a) and (b) give the original microarray data and its re-arrangements based on the clustering.