Sammon's Nonlinear Mapping Using Geodesic Distances

Li Yang
Department of Computer Science, Western Michigan University
E-mail: li.yang@wmich.edu

Abstract

Sammon's nonlinear mapping (NLM) is an iterative procedure to project high dimensional data into low dimensional configurations. This paper discusses NLM using geodesic distances and proposes a mapping method GeoNLM. We compare its performance through experiments to the performances of NLM and Isomap. It is found that both GeoNLM and Isomap can unfold data manifolds better than NLM. GeoNLM outperforms Isomap when the short-circuit problem occurs in computing the neighborhood graph of data points. In turn, Isomap outperforms GeoNLM if the neighborhood graph is correctly constructed. These observations are discussed to reveal the features of geodesic distance estimation by graph distances.

1. Introduction

The problem of data projection is defined as follows: given a set of high dimensional data points, project them to a low space so that the result configuration performs better than the original data in further processing such as clustering, classification, indexing and searching [4, 5]. Data projection has important applications in pattern analysis, data mining, information retrieval, and neural science.

One well-known method to project data is Principal Component Analysis (PCA) which provides mean-square optimized linear projection of data. The dimensions of the low space are determined by the eigenvectors corresponding to the largest eigenvalues of the covariance matrix of the original dimensions of data. In duality to PCA, classical Multi-Dimensional Scaling (MDS) works on inter-point distances and gives a low dimensional configuration that best represents the given distances.

PCA and classical MDS geometrically models data as on hyper-planes in embedded spaces. Because high dimensional data are usually located in low-dimensional nonlinear manifolds, nonlinear projection often provides more compact representation of data in many applications.

Sammon's Non-Linear Mapping (NLM) [3] is an iterative nonlinear procedure to project high dimensional data. It starts from a random low-dimensional configuration of data and adjusts the configuration to preserve the distances between all pairs of points. Recently, a lot of research on nonlinear data projection is focused on using geodesic distances instead of Euclidean distances. A representative example is Isomap[6] which applies classical MDS to geodesic distances. The geodesic distances are calculated hop-by-hop along the data manifold. Because geodesic distances reflect intrinsic distances between data points, Isomap has the ability to unfold heavily twisted data manifolds.

This paper proposes a method, GeoNLM, which applies NLM to geodesic distances. The main idea is to put together the bests of Isomap and NLM, namely geodesic distances instead of Euclidean distances and NLM instead of classical MDS. We give experiments on artificial data and real world data. The performance of GeoNLM is compared to that of NLM and of Isomap. We find that both Isomap and GeoNLM provide more faithful configurations than NLM. We show that NLM outperforms Isomap when the short-circuit problem happens due to improper choice of neighborhood size in calculating the neighborhood graph of all data points. We also show that Isomap outperforms GeoNLM once the neighborhood graph is correctly constructed. These observations reveal interesting features of geodesic distance estimation by using graph distances.

2. Sammon's Nonlinear Mapping

Sammon's Non-Linear Mapping (NLM) projects data and tries to preserve inter-point distances in the low-dimensional configuration. Suppose that we have n data points, $\mathbf{x}_i, i = 1, \ldots, n$, in D-space and, correspondingly, we define n points, $\mathbf{y}_i, i = 1, \ldots, n$, in d-space ($d < D$). Let d^*_{ij} denote the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ and d_{ij} denote the distance between the corresponding points $\mathbf{y}_i$ and $\mathbf{y}_j$. NLM works by randomly choosing an initial d-space configuration for $\mathbf{y}_i$. The algorithm then computes all inter-point distances d_{ij} in d-space, which are used to define an

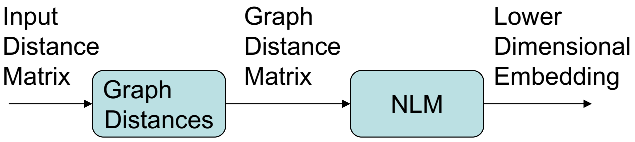


Figure 1. The two steps of GeoNLM.

error measure E as

$$E = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j}^{n} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}}$$

E is commonly referred as Sammon's stress. It represents how well the inter-point distances in the current configuration in d-space fit the inter-point distances in D-space. The next step is to adjust the d-space configuration so as to decrease E. NLM uses a steepest descent procedure to search for a minimum of E.

Sammon's stress is designed so that short distances contribute more to the value of E. In the process to minimize E, therefore, NLM puts more priority on the preservation of short distances rather than long ones. That is why NLM is capable of unfolding high dimensional data manifolds. Because the algorithm considers also long distances, however, it may fail to unfold strongly twisted patterns. One improvement over NLM is Curvilinear Component Analysis (CCA) [2] which uses a new error measure that totally ignores distances longer than a user-defined threshold.

3. GeoNLM

The use of geodesic distance gives a promising direction of research in data projection. One example is Isomap which applies classical MDS to geodesic distances. Because geodesic distances reflect more faithfully the underlying geometry of a data set, Isomap can compute global solutions of highly folded, twisted, or curved manifolds. The geodesic distances between all pairs of points are estimated by their shortest-path distances in the neighborhood graph of all points. Because graph distances approach geodesic distances as the number of data points increases, Isomap is guaranteed to converge asymptotically to the true intrinsic structure of data.

GeoNLM tries to combine the bests of NLM and Isomap by applying NLM to geodesic distances. The algorithm has two steps: calculate graph distances and apply NLM to the graph distances. These two steps are shown in Figure 1.

The input distances to GeoNLM can be measured either in the Euclidean metric or any other domain-specific metric. The same as in Isomap, there are two alternatives that can be used to define whether two points are neighbors: (1) if one is in the k nearest neighbors of the other (k-nearest neighbor approach); or (2) they are closer than a user-defined threshold ϵ (ϵ-nearest neighbor approach). A neighborhood graph is defined over all data points by connecting each pair of neighbor points with an edge whose length is the input distance between the pair. The algorithm then computes the graph distance between each pair of data points by calculating the shortest path between the pair on the neighbor graph using Floyd's algorithm or using repeatedly Dijkstra's shortest path algorithm. The only difference between GeoNLM and Isomap is that GeoNLM uses NLM in the second step. It applies NLM to the computed graph distances and finds a low dimensional configuration that best approximates the graph distances.

4. Artificial Examples and Discussion

Figure 2 shows the results of applying NLM, Isomap and GeoNLM to four 2D synthetic manifolds embedded in 3-space. Each test data set contains 1,000 points. The results of GeoNLM and Isomap are shown together with their neighborhood graphs. They are produced when 7 nearest neighbors of each point ($k = 7$) are used in constructing the neighborhood graph.

We can see that both Isomap and GeoNLM easily outperform NLM in unfolding data manifolds. Because geodesic distances reflect the underlying intrinsic differences between data points on a manifold, Isomap and GeoNLM are capable of finding correct intrinsic dimensionality and unfolding highly folded, twisted, or curved data distributions.

The results in Figure 2 show that Isomap and GeoNLM have similar performance. The difference between Isomap and GeoNLM is that GeoNLM uses NLM instead of classical MDS in the second step. While classical MDS is a linear projection which is equivalent to PCA, NLM is a nonlinear procedure that projects data with an emphasis on the preservation of short distances. Intuitively, NLM has more flexibility in adjusting the low dimensional configuration. Their difference becomes significant when the problem of "short-circuit" edges [1] occurs due to the improper choice of a too large neighborhood size in calculating the neighborhood graph. To illustrate this, Figure 3 gives the results of mapping the "U" shape in Figure 2 using the ϵ-neighbor approach. We can see that Isomap fails to unfold the manifold when $\epsilon = 6$ (the short-circuit problem occurs) while GeoNLM is still able to unfold the data. Figure 5 shows the Sammon's stress values of these projections against the neighborhood size ϵ. GeoNLM gives smaller stress values than Isomap and the difference becomes significant as ϵ increases.

A closer look at these results explains the behavioral difference of Isomap and GeoNLM: The distances on short-circuit edges are long distances. When the parameter k or ϵ is chosen too large so that short-circuit edges happen, all af-

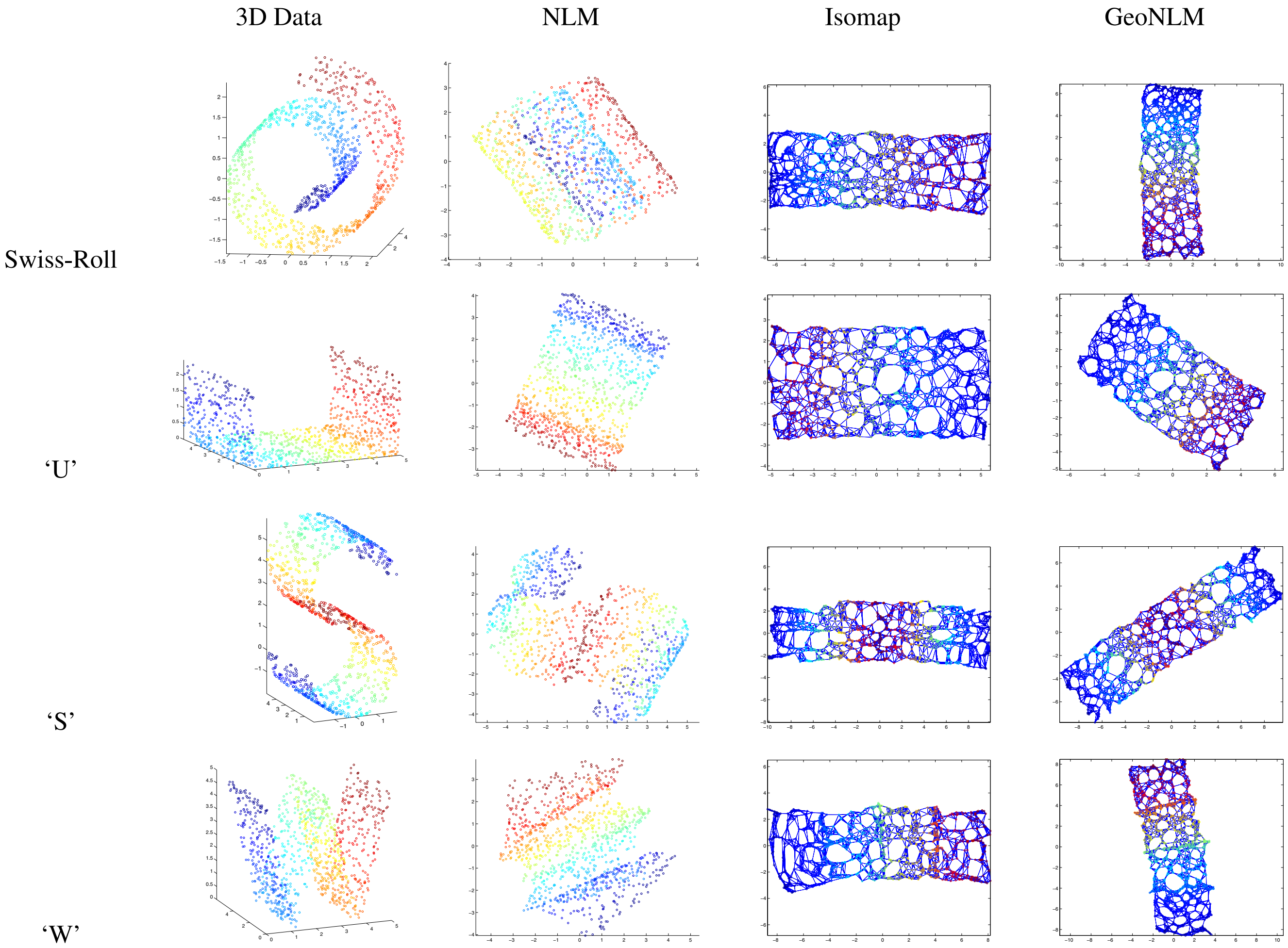


Figure 2. Experiments on artificial data sets. ($k = 7$)

fected graph distances that are measured along these edges are long distances. NLM has rightly been designed to put less emphasis on the preservation of long distances. That is why GeoNLM outperforms Isomap when the short-circuit edge problem occurs. Isomap uses classical MDS which treats equally short and long distances.

Another interesting observation (not shown in this paper due to space limitation) is that Isomap slightly outperforms GeoNLM once a correct neighborhood graph is constructed. A close look shows that a constructed neighborhood graph (for example, those shown in Figure 2) looks like a Swiss-cheese with a lot of holes inside. For the short geodesic distance between a close pair of points, therefore, graph distance between the pair may not be a good estimation of the geodesic distance. For example, the graph distance between two points on opposite banks of a hole is measured along the bank of the hole. In this case, GeoNLM performs worse than Isomap because the underlying NLM tries to best preserve short distances, some of which are over-estimations of the corresponding geodesic distances.

5. Real-World Example

We have tested GeoNLM by using an electronic nose data which consists 300 samples of the odors of six chemicals: allyl caproate, methyl salicylate, isoamyl acetate, myrcene, decanal, and diacetyl. Each sample has 32 attributes. Figure 4 shows the mapping results of the six chemicals using NLM, Isomap, and GeoNLM. We can see that GeoNLM outperforms NLM and Isomap in separating the data samples of these chemicals.

6. Conclusion

This paper has presented a data projection method, GeoNLM, which applies NLM to the matrix of graph-based geodesic distances. Its performance in unfolding data manifolds has been compared to the performances of NLM and

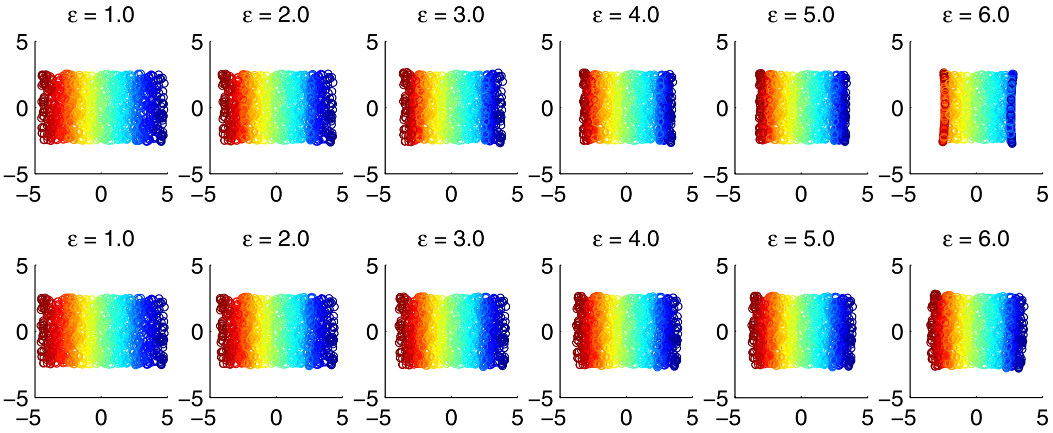
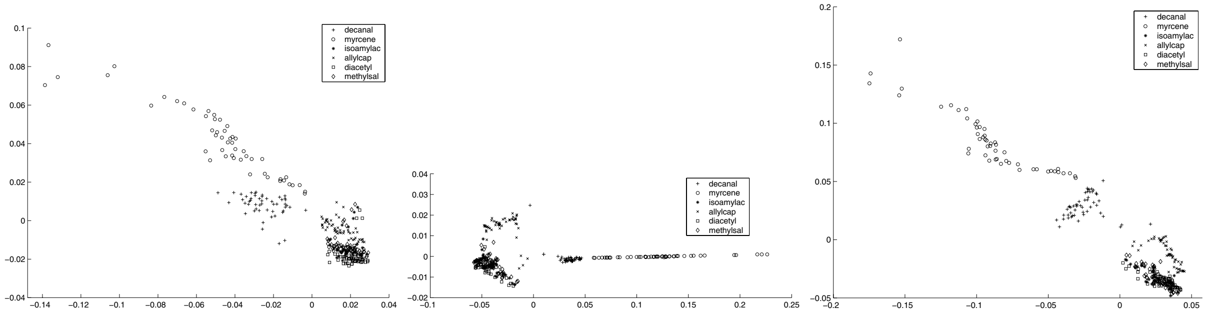


Figure 3. Projections of the 'U' shape using Isomap (above) and GeoNLM (below).



(a) (b) (c)

Figure 4. Projections of the electronic nose data by using (a) NLM, (b) Isomap, and (c) GeoNLM.

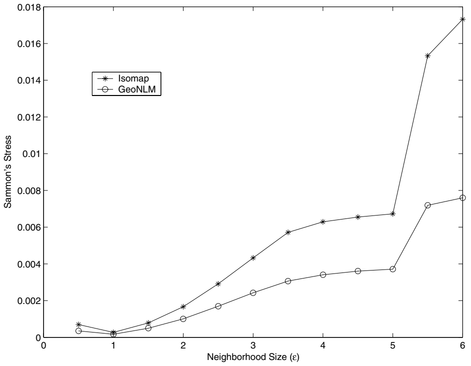


Figure 5. Sammon's stresses of the projections in Figure 3.

Isomap. Experiments show that both GeoNLM and Isomap outperform NLM in unfolding data manifolds. GeoNLM outperforms Isomap when the short-circuit problem occurs due to the improper choice of a too large neighborhood size in calculating the neighborhood graph. Isomap works better than GeoNLM once a correct neighborhood graph is constructed. These observations reveals interesting features of geodesic distance estimation by using graph distances.

References

[1] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford. The Isomap algorithm and topological stability. *Science*, 295:7a, Jan. 2002.

[2] P. Demartines and J. Herault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans. Neural Networks*, 8(1):148–154, Jan. 1997.

[3] J. J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Computer*, C-18(5):401–409, May 1969.

[4] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[5] A. K. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):4–37, Jan. 2000.

[6] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, Dec. 2000.