ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Estimation of Tangent Planes for Neighborhood Graph Correction

Karina Zapién, Gilles Gasso and Stephane Canu

INSA de Rouen - LITIS, EA4051
1060 Av. Université, Saint Etienne du Rouvray - France

Abstract. Local algorithms for non-linear dimensionality reduction [1], [2], [3], [4], [5] and semi-supervised learning algorithms [6], [7] use spectral decomposition based on a nearest neighborhood graph. In the presence of *shortcuts* (union of two points whose distance measure along the submanifold is actually large), the resulting embbeding will be unsatisfactory. This paper proposes an algorithm to correct wrong graph connections based on the tangent subspace of the manifold at each point. This leads to the estimation of the proper and adaptive number of neighbors for each point in the dataset. Experiments show graph construction improvement.

1 Introduction

Spectral decomposition has become a broad used technique for dimensionality reduction algorithms (Isomap [1], Laplacian Eigenmaps [3], Hessian Eigenmaps [4], MVU [5]) and semi-supervised learning algorithms (LapSVM [6], [7]). These methods consist on defining a neighborhood graph, were nodes represent data points and edges indicate if two points are close to each other (therefore, nearest neighbors). Under certain conditions, satisfactory results can be obtained by using these techniques.

One important drawback is the fact that the underlying manifold can't be estimated if the nearest neighborhood graph is not properly defined and contains *shortcuts*. That is, two points are considered to be neighbors whereas they are actually far away from each other in the sense of measured distance along the manifold. This is explained by the fact that most dimensionality reduction algorithms' goal is to preserve the euclidean distance between the nearest neighbors defined by the graph. The algorithm is constrained to keep these distances in the new embedding, but these can be constraints not achievable in a smaller space.

This paper uses the idea that a manifold is differentiable and therefore a tangent subspace to the original manifold can be estimated. This will correspond to a tangent hyperplane in a lower space. All nearest neighbors should lie close to this tangent subspace, where distance will be measured not only with the euclidean distance but also with the deviation angle to the tangent subspace.

2 Background

We are willing to find the embedding manifold $\mathcal{M} \subset \mathbb{R}^D$ of input data $X = \{\mathbf{x}_i\}$, $\mathbf{x}_i \in \mathcal{M}$, $i = 1, \ldots, n$. The pursued objective is the projection of X into a subspace $\mathbb{R}^d$ $(d < D)$ that preserves the topological characteristics of $\mathcal{M}$. This will yield to dataset $Y = \{\mathbf{y}_i\},\ \mathbf{y}_i \in \mathbb{R}^d$. To achieve this goal, several spectral decomposition techniques [1], [4], [5] proceed by constructing the neighborhood graph related to the training points and then finding a transformation of this graph. Two common used techniques to determine nearest neighbors of each point in the dataset are:

- k-NN technique, which consists on choosing for each $\mathbf{x}_i$ in the dataset the closest k points as nearest neighbors of $\mathbf{x}_i$.
- ϵ-ball, where all points with distance to $\mathbf{x}_i$ less or equal than ϵ are considered nearest neighbors of $\mathbf{x}_i$.

Local approximation techniques may have good results, but if the distribution along the submanifold is not uniform or dense enough, the adjacency graph will be either not connected or shortcuts will appear.

To test our algorithm, the Isomap method is used, which is an extension of the Multidimensional Scaling [8] algorithm. Isomap is based on spectral decomposition of the geodesic distance matrix. Approximation of the geodesic distances is done by measuring the shortest path along the graph.

Choi [9] developed an algorithm to remove noisy points that produce shortcuts. For each point $\mathbf{x}_i$ the total flow is defined as the number of shortest paths passing through $\mathbf{x}_i$. This criteria for removing points, might delete useful points and might miss several edges that should be removed. A similar approach was used by Yang [10]. Other approaches are based on the construction of minimal spanning trees [11] which avoids having cycles, but will not necessarily respect intrinsic topological aspects. Finally, the work of Mekus [12] and Wang [13] are based on the estimation of the tangent subspace at each point independently.

We propose a method to correct neighborhood graphs using this last approach of tangent subspaces. One important difference is the fact that the estimation of the tangent space is done taking into consideration tangent spaces on neighboring points and therefore, respecting continuity in the manifold.

3 Manifold and Tangent Subspace

By definition, a manifold is a differentiable surface, therefore, if we have its k nearest neighbors, and the data lies on a connected submanifold $\mathcal{M} \in \mathbb{R}^d$, assuming that $d < k$, the tangent subspace at each point can be estimated from its k nearest neighbors. If additionally, the maximum curvature θ of the manifold is known, analysis of nearest neighbors can be done regarding the euclidean distance with respect to a point $\mathbf{x}_i$ (which should be smaller than a value ϵ deduced from θ), and the angle deviation with respect to the tangent subspace $\mathcal{P}_i$ at $\mathbf{x}_i$ (which should be smaller than θ).

Let $\mathcal{N}_i = \{\mathbf{x}_j | \mathbf{x}_j$ is nearest neighbor of $\mathbf{x}_i\}$ with $|\mathcal{N}_i| = k_i$ be the set containing the k_i nearest neighbors of $\mathbf{x}_i$.

For all neighborhood $\mathcal{N}_i$ $i = 1, ..., n$, define $I_i = \{j | \mathbf{x}_j \in \mathcal{N}_i\}$, as the set of indexes of points belonging to neighborhood $\mathcal{N}_i$, which will be also denoted as $I_i = \{i_1, i_2, ..., i_{k_i}\}$.

ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Finally, define a matrix A_i of the form $A_i = [\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, ..., \mathbf{x}_{i_{k_i}}] \in \mathbb{R}^{D \times k_i}$ for all $i = 1, ..., n$.

The implemented algorithm iterates between obtaining an estimation of the set $\mathcal{N}_i$ and the approximation of the tangent subspace $\mathcal{P}_i$ at $\mathbf{x}_i$ defined by $\mathcal{N}_i$.

3.1 Tangent subspace estimation and neighborhood estimation

If the set of nearest neighbors $\mathcal{N}_i$ for point $\mathbf{x}_i$ is well defined, that is, if the euclidean distance in the original space approximate the distance along the manifold, the desired orthogonal vector $\mathbf{w}_i$ and bias b_i that define the tangent subspace $\mathcal{P}_i$ at $\mathbf{x}_i$ are the ones that minimize $A_i^{\mathrm{T}}\mathbf{w}_i + b\mathbf{1}$, with $\mathbf{1} \in \mathbb{R}^{k_i}$ being a vector of ones. These can be estimated as follows:

$$\mathcal{P}_i(\mathcal{N}_i) = \{\mathbf{w}_i,\ b_i\}, \quad \text{with} \quad \mathbf{w}_i = \frac{(A_i^{\mathrm{T}})^{+}\mathbf{1}}{||(A_i^{\mathrm{T}})^{+}\mathbf{1}||} \text{ and } \ b_i = -\mathbf{x}_i^{\mathrm{T}}\mathbf{w}_i \tag{1}$$

where $A_i^{\mathrm{T}+}$ represents the pseudoinverse of matrix A_i^{T}.

Vice versa, if we have an adjusted tangent subspace $\mathcal{P}_i$ at $\mathbf{x}_i$ and an initial neighborhood $\mathcal{N}_i^0$, we can measure the euclidean distance ϵ_{ij} of $\mathbf{x}_j \in \mathcal{N}_i^0$ to $\mathbf{x}_i$ and the deviation angle θ_{ij} with respect to $\mathcal{P}_i$ as follows

$$\epsilon_{ij} = ||\mathbf{x}_j - \mathbf{x}_i||_2 \qquad \theta_{ij} = \arcsin\frac{|\ \langle \mathbf{w}_i, (\mathbf{x}_j - \mathbf{x}_i)\rangle|}{||\mathbf{x}_j - \mathbf{x}_i||_2}. \tag{2}$$

Then, neighborhood $\mathcal{N}_i$ can be updated as follows:

$$\mathbf{x}_j \in \mathcal{N}_i \quad \text{if} \quad \epsilon_{ij} < \epsilon \ \text{ and } \ \theta_{ij} < \theta^0 \quad \forall\ \mathbf{x}_j \in \mathcal{N}_i^0. \tag{3}$$

where θ^0 is the maximum curvature input by the user and ϵ is estimated by the algorithm (see step (4) in Section (4.1)).

3.2 Propagation of the tangent subspace estimation

Having a reliable estimation of $\mathcal{P}_i$ and $\mathcal{N}_i$ for point $\mathbf{x}_i$, this subspace is a good approximation of $\mathcal{P}_j$ for a $\mathbf{x}_j \in \mathcal{N}_i$. Then, $\mathcal{P}_i$ can be use to remove faulty nearest neighbors in $\mathcal{N}_j^0$ by using a relaxed parameter $\theta' > \theta^0$ in Eqs. (3). Remaining points will form set $\mathcal{N}_j$ which will allow us to calculate $\mathcal{P}_j(\mathcal{N}_j)$ using Eqs. (1).

3.3 Finding a starting point

The calculation of the tangent subspace at each point $\mathbf{x}_i$ is done by propagation of the tangent subspace $\mathcal{P}_i$ at point $\mathbf{x}_i$ to its neighbors. This can be done only if we know that $\mathcal{P}_i$ is a reliable approximation. Therefore, the departure point is an important matter.

A first approximation to find $\mathcal{N}_i$, can be done by using the k nearest neighbors of each point, then each $\mathcal{P}_i$ will be estimated with these neighborhoods. If at a point $\mathbf{x}_i$ and at all its k-nearest neighbors in $\mathcal{N}_i$ a close approximation of the tangent subspace was found, the perpendicular vectors to their tangent subspaces should be all quite similar. Therefore, it will be taken as starting point $\mathbf{x}_{t_0}$, the one whose orthogonal vector $\mathbf{w}_{t_0}$ maximizes the cost function

$$C(\mathbf{w}_i, I_i) = \frac{1}{k_i} \sum_{j \in I_i} |\langle \mathbf{w}_i, \mathbf{w}_j \rangle|. \quad (4)$$

If there is a shortcut at point $\mathbf{x}_i$, the estimated subspace at this point will be biased, and this subspace won't have the same orientation as its nearest neighbors, giving a smaller value in Equation (4).

4 Algorithm Description

4.1 Initialization phase

This phase consists on finding a point $\mathbf{x}_{t_0}$ located in a low curvature zone. So further tangent subspace approximations based on $\mathcal{P}_{t_0}$ are reliable.

1: **Input:** curvature θ and the maximum number of neighbors k, set $k_i = k$.
Set $\mathcal{N}_i^0 \leftarrow \{\mathbf{x}_j | \mathbf{x}_j$ is a k-nearest neighbor of $\mathbf{x}_i\}$, $\forall i = 1, ..., n$.
Set $\Phi_{ij} = \infty,\ i, j = 1, ..., n$, which will be later updated with the deviation angle of $\mathbf{x}_j$ to stable tangent subspaces $\mathcal{P}_i$.
2: Estimate the initial tangent subspace $\mathcal{P}_i^0(\mathcal{N}_i^0) = \{\mathbf{w}_i, b_i\}$, $i = 1, ..., n$ using an analogous form of Eq. (1).
3: Find $\mathbf{x}_{t_0}$ with the best tangent subspace approximation, that is, find t_0 that maximizes (4)
4: Set $\epsilon = ||\mathbf{x}_{t_0} - \mathbf{x}_{t_{0\,k+1}}||$, being $\mathbf{x}_{t_{0\,k+1}}$ the $(k+1)$-nearest neighbor of $\mathbf{x}_{t_0}$. This sets for all $\mathbf{x}_i \in X$ the maximum radius of ϵ-balls for a neighborhood $\mathcal{N}_i$.
5: Set $I_{done} \leftarrow \{\mathbf{x}_{t_0}\}$. Index of points $\mathbf{x}_i \in X$ having a stable tangent space $\mathcal{P}_i$.
6: Select nearest neighbors $\mathcal{N}_{t_0} \in \mathcal{N}_{t_0}^0$ using tangent plane $\mathcal{P}_{t_0}^0$ and Eqs. (3).
7: Update $\mathcal{P}_{t_0}(\mathcal{N}_{t_0})$ using Eq. (1).
8: Set deviation angle $\Phi_{t_0 j} = \angle(\mathbf{w}_{t_0}, \mathbf{x}_j - \mathbf{x}_{t_0})$, for all $j \notin I_{done}$.

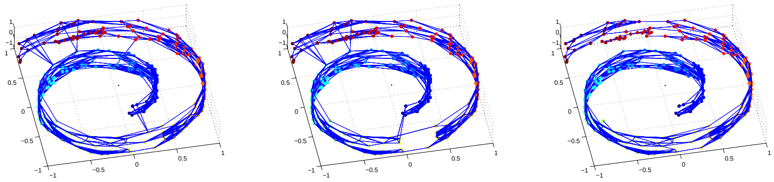
4.2 Successive Approximation phase

This phase uses reliable estimations of tangent subspaces to make approximations at other points. Graph is traversed considering deviation angles.

1: Choose $t = \text{argmin}_j\{\Phi_{ij}|\ i \in I_{done},\ j \in I_i\}$, so that for a point $\mathbf{x}_s,\ s \in I_{done},\ \mathbf{x}_t \in \mathcal{N}_s$ has the minimum deviation to tangent subspace $\mathcal{P}_s$, let $s = \text{argmin}_i\{\Phi_{it},\ i \in I_{done}\}$.
2: Select nearest neighbors $\mathcal{N}_t \in \mathcal{N}_t^0$ using Eqs. (3) with $\mathcal{P}_t = \mathcal{P}_s$ and $\theta = \theta + \angle(\mathbf{x}_t - \mathbf{x}_s, \mathcal{P}_s)$.
3: Update $\mathcal{P}_t(\mathcal{N}_t)$ with Eqs. (1).
4: Set $I_{done} = I_{done} \cup \{\mathbf{x}_t\}$.
5: Set $\Phi_{tj} = \angle(\mathbf{w}_t, \mathbf{x}_j - \mathbf{x}_t),\ j \notin I_{done}$. Set $\Phi_{jt} = \infty,\ j \in I_{done}$ to avoid selecting again $\mathbf{x}_t$.
6: If $|I_{done}| < n$, go to step (1), else go to (7).
7: If the graph is not connected, make a single component based on $\frac{\Phi_{ij}}{\epsilon ij}$.

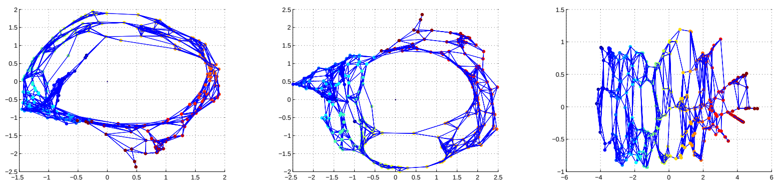
5 Experimental Results

To validate our algorithm, we applied it on a swissroll example which real underlying dimensionality lies in $\mathbb{R}^2$. We generated a lightly sparse dataset with 300 points and compared our results to k-NN and the total flows algorithm [9]. Parameter k was set to values between 8 and 16, giving in our case similar results. Curvature θ was set to 20 degrees.



(a) k-NN, $k = 8$ (b) Flows use (c) Tangents Propagation

Fig. 1: Resulting NN graph



(a) k-NN, $k = 8$ (b) Flows use (c) Tangents Propagation

Fig. 2: Resulting embedding

If the tangent subspace at each point is estimated and the neighborhood graph is built using this subspace, we are able to reduce the shortcuts in the original graph construction with the k-NN method. Additionally experiments showed that if we add Gaussian noise with zero mean and .1 variance to this dataset, our algorithm is robust.

6 Conclusions

The estimation of tangent subspaces at each point can improve the structure of a initial graph given a maximum number of nearest neighbors. Use of neighborhood information helps to improve the estimation of tangent subspaces. We propose an algorithm that helps to detect the presence of shortcuts in a graph. Future work includes automatic estimation of the manifold curvature and use of this technique in real data.

References

[1] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[2] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323 – 2326, December 2000. DOI: 10.1126/science.290.5500.2323.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. The MIT Press, 2002.

[4] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. In Stanley Falkow, editor, *Proceedings of the National Academy of Sciences of the United States of America*, volume 100, pages 5591–5596, May 2003.

[5] Kilian Weinberger and Lawrence Saul. Unsupervised learning of image manifolds by semidefinite programming. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04) Washington D.C.*, volume 20, 2004.

[6] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from examples. August 2004.

[7] A. Argyriou, M. Herbster, and M. Pontil. Combining graph laplacians for semi-supervised learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 67–74. MIT Press, Cambridge, MA, 2005.

[8] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, London, 1994.

[9] Heeyoul Choi and Seungjin Choi. Robust kernel isomap. *The journal of the pattern recognition society*, 40, 2007.

[10] Li Yang. Building k-connected neighborhood graphs for isometric data embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):827–831, 2006.

[11] Dongfang Zhao and Li Yang. Incremental construction of neighborhood graphs for nonlinear dimensionality reduction. In *ICPR (3)*, pages 177–180. IEEE Computer Society, 2006.

[12] Nathan Mekuz and John K. Tsotsos. Parameterless isomap with adaptive neighborhood selection. In K. Franke et al., editor, *DAGM 2006*. Springer-Verlag Berlin Heidelberg, 2006.

[13] Jing Wang, Zhenyue Zhang, and Hongyuan Zha. Adaptive manifold learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1473–1480. MIT Press, Cambridge, MA, 2005.