ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 193-198

Fast nonlinear dimensionality reduction with topology preserving networks

J.J. Verbeek, N. Vlassis, B. Kröse *

Computer Science Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, the Netherlands

Abstract. We present a fast alternative for the Isomap algorithm. A set of quantizers is fit to the data and a neighborhood structure based on the competitive Hebbian rule is imposed on it. This structure is used to obtain low-dimensional description of the data by means of computing geodesic distances and multi dimensional scaling. The quantization allows for faster processing of the data. The speed-up as compared to Isomap is roughly quadratic in the ratio between the number of quantizers and the number of data points. The quantizers and neighborhood structure are use to map the data to the low dimensional space.

1 Introduction

In this work we consider high dimensional data $\mathbf{X}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathbb{R}^D$ that lie on a $d \ll D$ dimensional, possibly nonlinear, manifold (plus some noise outside the manifold). The goal is to express the data in the intrinsic coordinates of the manifold. If we are able to map the data to such a low dimensional description then further processing and storage of the data is facilitated, simply because the processing and storage cost is generally at least linear in the dimensionality and also because we may avoid 'the curse of dimensionality'. We combine ideas from Topology Representing Networks [4], Isomap [7] and Local Linear Embedding (LLE) [5].

Our approach first represents the data with a Topology Representing Network (TRN). Then, pairwise distances on the graph between the nodes of the network are computed. The pairwise distances are used to obtain coordinates in $\mathbb{R}^d$ of the nodes that mimic the pairwise distances in the graph. These coordinates are obtained by means of Multi Dimensional Scaling (MDS) [1]. To map the complete data set to $\mathbb{R}^d$ local linearity of the manifold is assumed. This allows us to express the data as (convex) linear combinations of nodes of

*This research is supported by the Technology Foundation STW (project nr. AIF4997) applied science division of NWO and the technology program of the Dutch Ministry of Economic Affairs.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 193-198

the network in $\mathbb{R}^{\mathrm{D}}$. The same linear combinations are then used to reconstruct the points in $\mathbb{R}^{\mathrm{d}}$.

The virtues of this procedure as compared to Isomap are (i) The computational effort spent by Isomap on constructing the neighborhood graph and the computation of pairwise distances is greatly reduced. The speed up is roughly quadratic in the ratio between the number of quantizers and the number of data points. (ii) In situations where it is unfeasible to store all data, the TRN can be learned online (several methods exist to do so, see for example [2]). (iii) Once the TRN is learned, it can be used to map data to and from the latent space. We do not need to learn an explicit mapping, say Radial Basis Function network, from the quantizers to map the other data to the latent space.

2 Non-linear dimensionality reduction

In this section we briefly discuss three works on nonlinear dimensionality reduction from which ideas are combined in the work presented here.

Topology representing networks Martinetz and Schulten showed [4] how the simple competitive Hebbian (CHR) rule forms topology representing networks. Let us define $Q = \{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$ as a set of points, called quantizers, on a manifold $M \subset \mathbb{R}^{\mathrm{D}}$. With each quantizer $\mathbf{q}_i$ a Voronoi set V_i is associated in the following manner: $V_i = \{\mathbf{x} \in \mathbb{R}^{\mathrm{D}} : \| \mathbf{q}_{\mathrm{i}} - \mathbf{x} \| = \min_{\mathrm{j}} \| \mathbf{q}_{\mathrm{j}} - \mathbf{x} \|\}$, where $\| \cdot \|$ denotes the vector norm. The Delaunay triangulation $\mathcal{D}_Q$ associated with Q is defined as the graph that connects quantizers with adjacent Voronoi sets.[1] The masked Voronoi sets $V_i^{(M)}$ are defined as the intersection of the original Voronoi sets with the manifold M. The Delaunay triangulation $\mathcal{D}_Q^{(M)}$ on Q induced by the manifold M is the graph that connects quantizers if the intersection of their masked Voronoi sets is non-empty.

Given a set of quantizers Q and a finite data set $\mathbf{X}_n$, the CHR produces a set of edges as follows: (i) For every $\mathbf{x}_i \in \mathbf{X}_n$ determine the closest and second closest quantizer, respectively $\mathbf{q}_{i_0}$ and $\mathbf{q}_{i_1}$. (ii) Include (i_0, i_1) as an edge in E. A set of quantizers Q on M is called *dense* if for each $\mathbf{x}$ on M the triangle formed by $\mathbf{x}$ and its closest and second closest quantizer lies completely on M. Obviously, if the distribution of the quantizers over the manifold is homogeneous (the volumes of the associated Voronoi regions are equal), the quantization can be made dense simply by increasing the number of quantizers.

Martinetz and Schulten showed that if Q is dense with respect to M, the CHR produces the induced Delaunay triangulation.

Isomap The Isomap algorithm [7] finds coordinates in $\mathbb{R}^{\mathrm{d}}$ of data that lie on a d dimensional manifold embedded in a $D \gg d$ dimensional space. The aim is to preserve the topological structure of the data, i.e. the Euclidean distances in $\mathbb{R}^{\mathrm{d}}$ should correspond to the geodesic distances (distances on the manifold). The algorithm makes use of a neighborhood graph to find the topological structure of the data. The neighborhood graph can be obtained either by

[1] Two Voronoi sets are called adjacent if their intersection is non-empty.

connecting all data points that are within some small distance from each other or by connecting each data point to its k nearest neighbors. The algorithm is summarized as follows: (i) Construct neighborhood graph. (ii) Compute the graph distance[2] between all data points using a shortest path algorithm, for example Dijkstra's algorithm. (iii) Find low dimensional coordinates by applying MDS on the pairwise distances.

The run time of the Isomap algorithm is dominated by the computation of the neighborhood graph, costing $O(n^2)$, and computing the pairwise distances, which costs $O(n^2 \log n)$.

Local Linear Embedding The idea underpinning the Local Linear Embedding (LLE) algorithm [5] is the assumption that the manifold is locally linear. It follows that small patches cut out from the manifold in $\mathbb{R}^{\mathrm{D}}$ should be approximately equal (up to a rotation, translation and scaling) to small patches on the manifold in $\mathbb{R}^{\mathrm{d}}$. Therefore, local relations among data in $\mathbb{R}^{\mathrm{D}}$ that are invariant under rotation, translation and scaling should also be (approximately) valid in $\mathbb{R}^{\mathrm{d}}$. Using this principle, the procedure to find low dimensional coordinates for the data is simple: Express each data point $\mathbf{x}_i$ as a linear (possibly convex) combination of its k nearest neighbors $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$: $\mathbf{x}_i = \sum_{j=1}^{k} w_{i_j} \mathbf{x}_{i_j} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is the approximation error whose norm is minimized by the weights that are used. Then we want to find coordinates $\mathbf{y}_i \in \mathbb{R}^{\mathrm{d}}$ such that $\sum_{i=1}^{n} \parallel \mathbf{y}_i - \sum_{j=1}^{k} w_{i_j} \mathbf{y}_{i_j} \parallel^2$ is minimized. It turns out that the $\mathbf{y}_i$ can be obtained by finding d eigenvectors of a $n \times n$ matrix.

3 The proposed algorithm

Let us first briefly outline the idea: (1) The topological structure of the data is captured by a TRN. (2) Low dimensional coordinates for the TRN nodes are computed. (3) The data is mapped into the low dimensional space using the low dimensional coordinates of the nodes.

(1a) First a set of quantizers that represents the data has to be generated. We propose to use robust variants of the Generalized Lloyd Algorithm (GLA) [3] to find these. (The GLA is also known as: k-means or LBG algorithm.) (1b) Then, we construct a graph on the quantizer set according to the CHR. Note that the edges emanating from a quantizer can be thought of as indicating the directions in which the manifold extends from this quantizer. (2a) Next, the geodesic distances between the quantizers are estimated by computing graph distances between the quantizers using Dijkstra's algorithm. (2b) We then use MDS to find d dimensional coordinates given these pairwise distances.

Once low dimensional coordinates have been found for the quantizers, we have to assign d dimensional coordinates to the data. We assume that the manifold is locally linear and that the quantizers are distributed densely over the manifold. In this case the manifold is spanned up, locally around a quantizer, by linear combinations of the edges to its neighboring quantizers. Hence,

[2] The graph distance is defined as the minimum length among all paths in the graph that connect the two data points. The length of a path is the sum of the lengths its edges.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 193-198

the assumption allows us to write $\mathbf{x} = \sum_{j \in N(\mathbf{q_x})} w_j \mathbf{q}_j + \boldsymbol{\epsilon}$, where $\mathbf{q_x}$ is the quantizer closest to $\mathbf{x}$, $N(\mathbf{q})$ gives the indices of the neighbors of $\mathbf{q}$ (and $\mathbf{q}$ itself) in the TRN and $\boldsymbol{\epsilon}$ is a small error term. Note that the weights w are invariant under scaling, rotation and translation of $\mathbf{x}$ and the quantizers in $N(\mathbf{q_x})$. Since the manifold is assumed to be locally linear, the same weights should also describe the data point in the d dimensional space. Therefore, we set the d dimensional coordinates as: $\mathbf{y}_i = \sum_{j \in N(\mathbf{q}_i)} w_j^i \mathbf{z}_j$, where $\mathbf{z}_j$ is the low dimensional coordinate of the j-th quantizer and w_j^i denotes the weight of quantizer j for reconstructing $\mathbf{x}_i$.

To map new data (not present in the training data set) to the low dimensional space, we compute the optimal weights of the closest quantizer and it neighbors in $\mathbb{R}^{\mathrm{D}}$ and use the same weights to produce the d dimensional coordinates for the new data point.

4 Some analysis of the algorithm

Concerning the topology preservation in the nonlinear map provided by MDS we can state a simple result. Given a set of quantizers and a neighborhood graph on them, we define the *neighbor-hull* of a quantizer $\mathbf{q}$ as the convex hull of $\mathbf{q}$ and it neighbors in the graph, i.e. $\{\mathbf{x} \in \mathbb{R}^{\mathrm{D}} : \mathbf{x} = \sum_{\mathrm{j} \in \mathrm{N}(\mathbf{q})} \mathrm{w_j} \mathbf{q_j}, \mathrm{w_j} > 0, \sum_{\mathrm{j}} \mathrm{w_j} = 1\}$. Two neighbor-hulls are called adjacent if their intersection is non-empty. For the result to hold, we must further constrain the weights w to be non-negative, so that the reconstruction is inside the neighbor-hull, see the comments below.

Lemma 1 *Let data points $\mathbf{v}_1, \mathbf{v}_2 \in M \subset \mathbb{R}^{\mathrm{D}}$ be touching. If we map these points to their low dimensional representation with our algorithm (using a dense set of quantizers), then they are mapped into adjacent neighbor-hulls in $\mathbb{R}^{\mathrm{d}}$.*

Proof Since $\mathbf{v}_1, \mathbf{v}_2 \in M \subset \mathbb{R}^{\mathrm{D}}$ are adjacent, they fall in adjacent Voronoi regions, say V_1 and V_2 (associated with quantizers $\mathbf{q}_1$ and $\mathbf{q}_2$). Because we use a dense set of quantizers, $\mathbf{q}_1, \mathbf{q}_2$ are connected in the network. Let $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{\mathrm{d}}$ be the low dimensional coordinates of $\mathbf{q}_1, \mathbf{q}_2$. Since the quantizers are adjacent, their neighbor-hulls share the edge $(\mathbf{z}_1, \mathbf{z}_2)$ in $\mathbb{R}^{\mathrm{d}}$ and hence they are adjacent neighbor-hulls. □

By the same argument we can show that if we map from $\mathbb{R}^{\mathrm{d}}$ to $\mathbb{R}^{\mathrm{D}}$ using the network obtained by applying the CHR on the data and the quantizers in the *low* dimensional space, then neighboring points in $\mathbb{R}^{\mathrm{d}}$ are mapped into adjacent neighbor-hulls in $\mathbb{R}^{\mathrm{D}}$.

To answer the question of how well the low dimensional representation of the data respected the topological structure of the data, we have two tools. First, as already indicated in [7], we can measure the correlation between the graph distances for the quantizers and the Euclidean distances in $\mathbb{R}^{\mathrm{d}}$. Second, we can measure the similarity between the networks obtained by applying the CHR in $\mathbb{R}^{\mathrm{d}}$ and $\mathbb{R}^{\mathrm{D}}$. In the style of [4], one might then call the mapping to low dimensionality topology preserving if the two networks are identical.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 193-198

In order to obtain the weights that reconstruct a point optimally inside the convex hull of the closest quantizer and its neighbors, we have to solve a quadratic programming problem which is computationally demanding. The computational effort can be significantly reduced if also negative weights are allowed, allowing for general linear combinations instead of only convex ones. The computations of these weights is exactly the same as the computation of the weights in LLE. To obtain the optimal weights for $\mathbf{x}_i$, up to a re-scaling to satisfy the sum to one constraint, we need to solve $\mathbf{C}\mathbf{w} = \mathbf{1}$. We use $\mathbf{w}$ to denote the weight vector. $\mathbf{C}$ is a symmetric matrix with $\mathbf{C}_{jk} = (\mathbf{x} - \mathbf{q}_j)^\top (\mathbf{x} - \mathbf{q}_k)$, where $j, k \in N(\mathbf{q}_{i_0})$. By adding to $\mathbf{C}$ a small multiple of the identity matrix, the reconstructions are drawn toward the center of the convex hull and hence toward the convex hull itself.

To show that the graph distances we use for the MDS mapping converge to manifold distances (how they are interpreted) the same arguments as used for the Isomap algorithm can be used. Note that in particular when the quantizers are assumed to be dense in the sense of Section 2, then is follows directly that the edges have length equal to the manifold distance between the quantizers.

5 An example on real-world data

We applied our method on data obtained from an omni-directional camera mounted on a Nomad Scout robot. The robot took pictures on 165 positions in an eight by one meter rectangular area in a corridor in our lab. The positions were distributed uniformly over the area, and the orientation of the robot was kept fixed. The omni-directional images were transformed to panoramic images of $400 \times 50 = 20000$ pixels. These images we first mapped into a 40 dimensional space by means of Principal Component Analysis, preserving 90% of the total variance in the data. Figure 1 shows the two dimensional coordinates found (by our algorithm and Isomap) for the data together with the quantizer network (connectivity obtained in the 40 dimensional space). The total run time of our algorithm (including the GLA with 30 clusters) was 1.49 seconds, where the Isomap algorithm took 9.62 seconds (five nearest neighbors we used for the neighborhood graph). Both algorithms found an embedding of the data that captures the intrinsic structure in the data and the distance correlation revealed in both cases the correct dimensionality. The speed-up is relatively small in this example, this is due to the small data set used.

6 Discussion

Advantages (i) If enough data is available and we use c times fewer quantizers as data points we obtain a speed-up, as compared to Isomap[3], of roughly c^2. (ii) In online learning (or limited memory) settings, we can apply the Dijkstra and MDS steps every once in a while on the online learned TRN, without the

[3] In [6] also a TRN is used, but a random subset of the data is used for the vertices. Also, the TRN is not used to map the data to $\mathbb{R}^d$.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 193-198

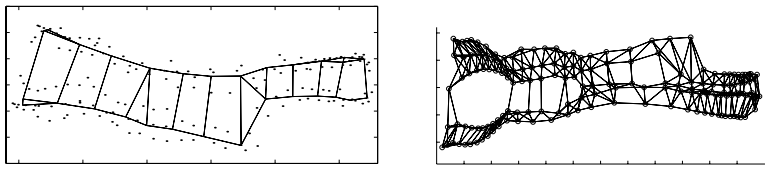

Figure 1: The $\mathbb{R}^2$ coordinates found by our algorithm (left) and Isomap (right).

need to store all data. (iii) By generating a new network on the quantizers by applying the CHR in $\mathbb{R}^d$, a new measure of topology preservation is available to assess the quality of the dimensionality reduction. (iv) New inputs are easily mapped to the low dimensional space.

Problems (i) In batch-learning settings, the applicability of the method depends on how dense the data-generating manifold is sampled. In very sparsely sampled cases, the network over the quantizers obtained with the CHR will incorrectly represent the manifold. Either we have to many quantizers as compared to data, then the network will consist of many disconnected components. Or we have to few quantizers, which will result in edges connecting areas that are relatively far away on the manifold. (ii) The number of quantizers to use introduces a parameter to be set by the user. However, with the Isomap algorithm one either has to choose the number of neighbors or a maximal distance to build the neighborhood graph, which is unnecessary in the proposed algorithm.

References

[1] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Number 59 in Monographs on statistics and applied probability. Chapman & Hall, 1994.

[2] B. Fritzke. *Handbook of Neural Computation*, chapter Unsupervised ontogenetic networks. IOP Publishing and Oxford University Press, 1996.

[3] A. Likas, N. Vlassis, and J.J. Verbeek. The global k-means clustering algorithm. Technical report, University of Amsterdam, 2001. IAS-UVA-01-02.

[4] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[5] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[6] J.B. Tenenbaum. Mapping a manifold of perceptual observations. In *Neural Information Processing Systems*, volume 10. MIT Press, 1997.

[7] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.