Isometric Feature Mapping (Isomap)

Büşra Güvenoğlu
busraguvenoglu@iyte.edu.tr

Department of Computer Engineering
Izmir Institute of Technology

05.04.2017

Outline

- Intuition
- Linear method- PCA
- Linear method- MDS
- Nonlinear method- Isomap

Dimensionality Reduction

Aim:

- to find a **small number of features** that represent a **large number of observed dimensions**.

Notations

Inputs **(high dimensional)**

- $x_1, x_2 \ldots x_n$ in R^D

Outputs **(low dimensional)**

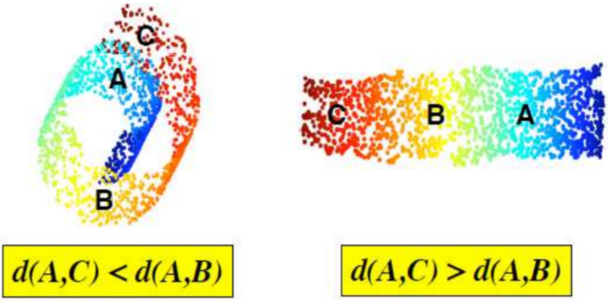
- $y_1, y_2 \ldots y_n$ in R^d(d<<D)

Goals

- Nearby points remain nearby.
- Distant points remain distant.

Motivation

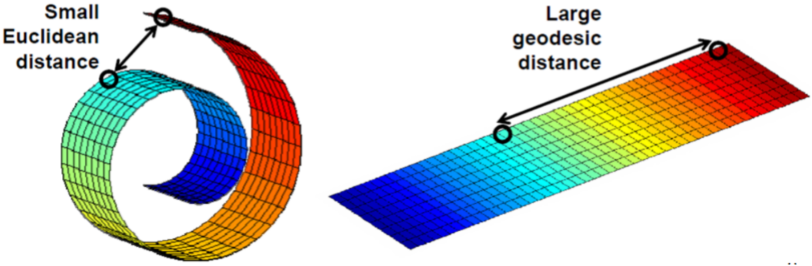
A **manifold** is a topological space which is locally Euclidean.



- Rank ordering of Euclidean distances is **NOT** preserved in "manifold learning".

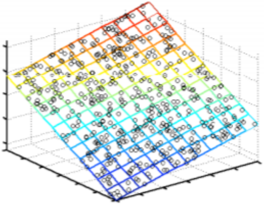
Geodesic Distance

- the **length of the shortest curve** between two points taken along the surface of a manifold



Dimensionality Reduction Techniques

- Linear
  - PCA
    - PCA finds the directions that have the most variance.
  - Classical MDS
    - Preserves Euclidean distances between points.
- Non-Linear
  - Isomap
    - Preserves geodesic distances between points.
  - LLE
    - Preserves local configurations in data



RECALL: Principal Components Analysis (PCA)

Main steps for computing PCs :

Input: $z_i \in R^D,\ i=1,..,n$ Output: $y_i \in R^d,\ i=1,..,n$

1. Subtract sample mean from the data

$$x_i = z_i - \hat{\mu}, \quad \hat{\mu} = 1/n \sum_i z_i$$

2. Compute the covariance matrix

$$C = 1/n \sum_{i=1}^{n} x_i x_i^t$$

3. Compute eigenvectors $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_d}$ corresponding to the ***d*** largest eigenvalues of C (d<<D).

4. The desired y is

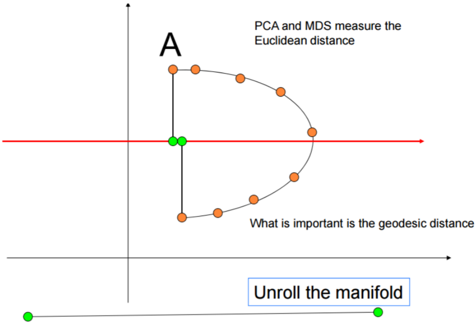
$$y = P^t x, \ P = [e_1, \ldots, e_d]$$

RECALL: Multidimensional Scaling (MDS)

Main steps:

- Compute pairwise distance matrix D
- Convert the pairwise distance matrix D into the inner product matrix B
- Decompose B into eigenvectors and eigenvalues B=XX^T
- Use top d eigenvectors and eigenvalues to form the d dimensional embedding.

Non-Linear Manifolds



- Unlike the geodesic distance, the Euclidean distance cannot reflect the geometric structure of the data points

Isometric Feature Mapping- **ISOMAP Algorithm**

The key idea:

- Use geodesic instead of Euclidean distances in MDS.
- To preserve structure preserve the geodesic distance and not the euclidean distance.
- The issue
  - How do we measure the geodesic distances on the manifold?
  - How to map points on the Euclidean space in lower dimensional space?

Isomap Algorithm is based on two simple ideas

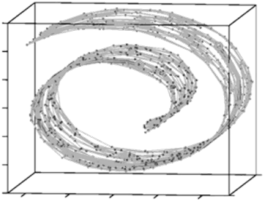
- For neighboring samples, Euclidean distance provides a good approximation to geodesic distance
- For distant points, geodesic distance can be approximated with a sequence of steps between clusters of neighboring points

Isomap operates in three steps

- Find nearest neighbors to each sample
- Find shortest path(e.g., Dijkstra)
- Apply MDS

Isomap Algorithm

- Determine the neighbors.
  - Connect each point to all points within a fixed radius ε
  - Connect each point to all of its K nearest neighbors
- These neighborhoods relation are represented as weighted graph G
  - each edge of weight $d_X(i, j)$ between neighboring points
- Result:



Isomap Algorithm

- Estimate the geodesic distances $d_M(i,j)$ between all pair of points on the manifold M by computing their shortest path distance $d_G(i,j)$ in the graph
  - Dijkstra Algorithm
  - Floyd Warshall Algorithm
- Find d dimensional embeddings that best preserves the manifold's estimated embeddings
  - apply classical MDS to the matrix of graph distances $d_G(i,j)$

Isomap Algorithm

(1) **Construct neighborhood graph**

(a) Define graph G by connecting points i and j if they are [as measured by $d_x(i,j)$]

closer than epsilon (*epsilon -Isomap*), or
if i is one of the K nearest neighbors of j (*K-Isomap*).

(b) Set edge lengths equal to $d_x(i,j)$.

(2) **Compute shortest paths**

(a) Initialize

$d_G(i,j) = d_X(i,j)$ if i,j are linked by an edge;
$d_G(i,j) = \infty$ otherwise.

(b) For $k = 1, 2, \ldots, N$, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$.

(c) Matrix $D_G = \{d_G(i,j)\}$ will contain the shortest path distances between all pairs of points in G.

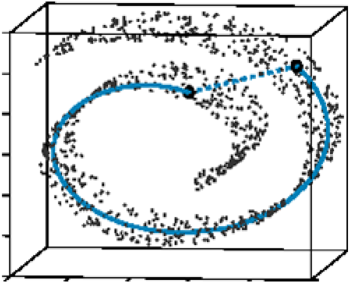
(3) **Construct d-dimensional embedding**

(a) Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$, and v_p^i be the *i-th* component of the *p-th* eigenvector.

(b) Set the *p-th* component of the *d*-dimensional coordinate vector y_i equal to $\mathrm{sqrt}(\lambda_p)v_p^i$.

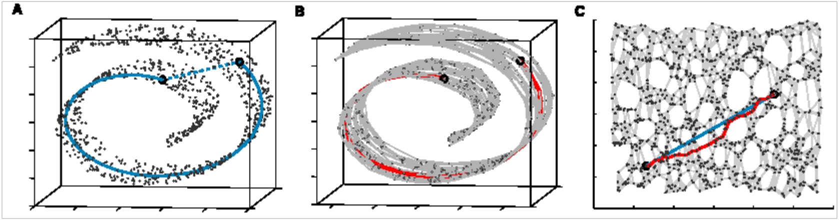
Examples-Sample points with Swiss Roll

- Altogether there are 20,000 points in the "Swiss roll" data set. We sample 1000 out of 20,000.



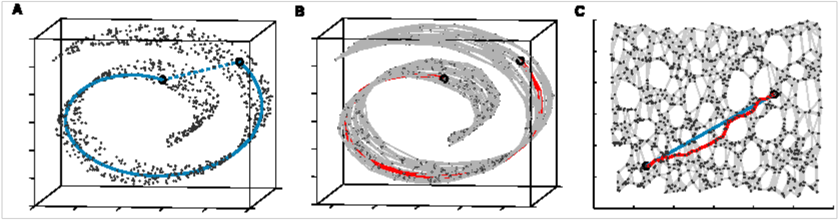
Examples-Construct neighborhood graph G

- K- nearest neighborhood (K=7) D_G is 1000 by 1000 (Euclidean) distance matrix of two neighbors (figure A)



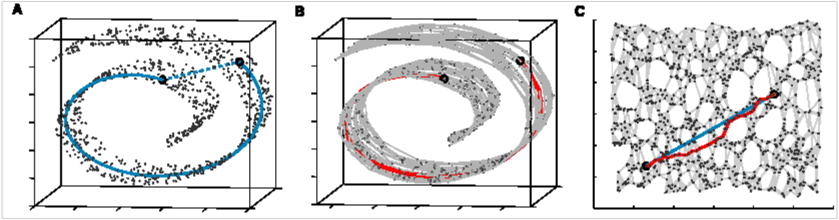
Examples-Compute all-points shortest path in G

- Now D_G is 1000 by 1000 geodesic distance matrix of two arbitrary points along the manifold (figure B)



Examples-Use MDS to embed graph in R^D

- Find a d-dimensional Euclidean space Y (Figure c) to preserve the pairwise distances.



Conclusions

- ISOMAP is a nonlinear MDS method to recovering intrinsic manifold in a low dimensional space via minimum distortion embedding.
- It has three steps:
  - construct a neighbourhood graph,
  - compute shortest path and
  - construct low-dimensional embedding.

References

- http://studentnet.cs.manchester.ac.uk/pgt/COMP61021/lectures/ISOMAP.pdf
- http://www.cs.haifa.ac.il/~rita/uml_course/lectures/Isomap_LLE_Lap.pdf
- http://www.math.pku.edu.cn/teachers/yaoy/Spring2011/lecture08.pdf

Question and Answers

THANK YOU!