ISOMAP

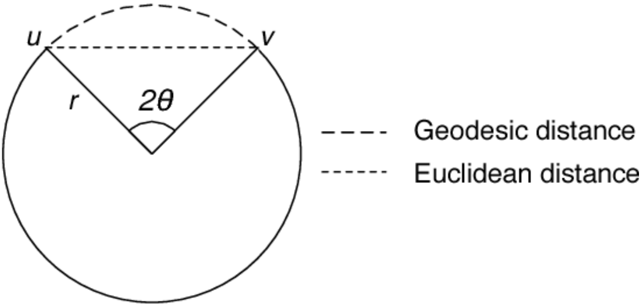
Moteu Ngoli Tatiana

AMMI-Ghana

December 18, 2019

OUTLINE

Problem??



Definitions

- Euclidean distance between points p and q is the length of the line segment connecting them denoted by $\overline{pq}$.
- In Cartesian coordinates, if
$p = (p_1, p_2, \ldots, p_n)$ *and* $q = (q_1, q_2, \ldots, q_n)$ are two points in Euclidean n-space, by using Pythagorean formula one has:

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

- The Euclidean norm

$$\begin{aligned} \|p\| &= \sqrt{p_1^2 + p_2^2 + \ldots + p_n^2} \\ &= \sqrt{p.p} \end{aligned}$$

Geodesic distance

Is the length of the shortest curve between those two points along the surface like the earth

Isomap(Manifold learning):

Isomap(Manifold learning):

- Stands for isometric mapping

Cont'd

Isomap(Manifold learning):

- Stands for isometric mapping
- Is a non-linear dimensionality reduction method

Cont'd

Isomap(Manifold learning):

- Stands for isometric mapping
- Is a non-linear dimensionality reduction method
- Preserve the geodesic distances in the lower dimension.

Isomap(Manifold learning):

- Stands for isometric mapping
- Is a non-linear dimensionality reduction method
- Preserve the geodesic distances in the lower dimension.
- If we measure the distance between two points by following the manifold, we will have a better approximation of how far or near two points are.

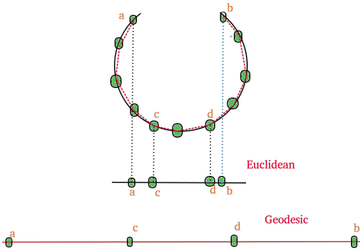


Figure: Isomap

How Isomap operates?

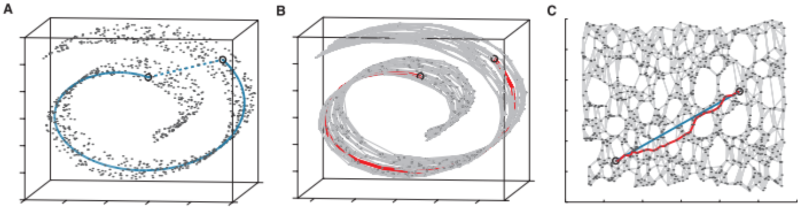


Figure: Isomap

Cont'd

How Isomap operates?

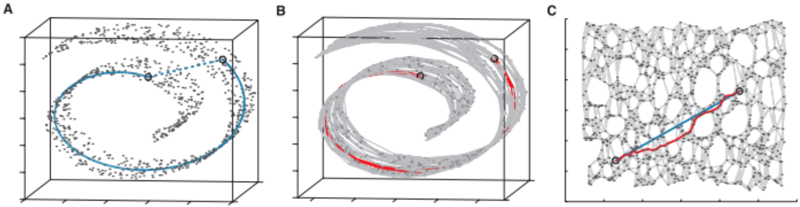


Figure: Isomap

- In A we see that two points that are close together in Euclidean Space in this "Swiss roll" dataset may not reflect the intrinsic similarity between these two points.

How Isomap operates?

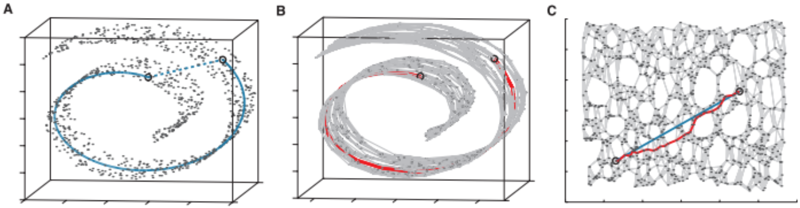


Figure: Isomap

- In A we see that two points that are close together in Euclidean Space in this "Swiss roll" dataset may not reflect the intrinsic similarity between these two points.

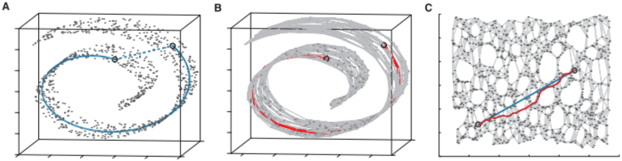


Figure: Isomap

- In B a graph is constructed with each point as n nearest neighbours. The shortest geodesic distance is then calculated by a path finding algorithm such as **Djikstra's Shortest Path**.
- In C, this is the 2D graph is recovered from applying classical MDS (Multidimensional scaling) to the matrix of graph distances. A straight line has been applied to represent a simpler and cleaner approximation to the true geodesic path shown in A.

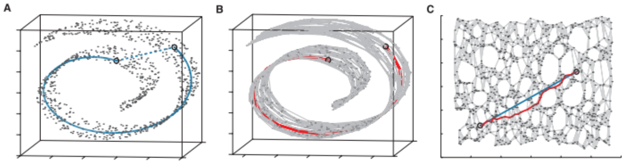


Figure: Isomap

- In B a graph is constructed with each point as n nearest neighbours. The shortest geodesic distance is then calculated by a path finding algorithm such as **Djikstra's Shortest Path**.
- In C, this is the 2D graph is recovered from applying classical MDS (Multidimensional scaling) to the matrix of graph distances. A straight line has been applied to represent a simpler and cleaner approximation to the true geodesic path shown in A.
- Isomap should be used when there is a non-linear mapping between your higher-dimensional data and your lower-dimensional manifold (e.g. data on a sphere).

Relationship with other methods

The doubly centered geodesic distance matrix K in Isomap is of the form:

$$K = -\frac{1}{2} H D^2 H$$

where $D^2 = D_{ij}^2 := (D_{ij})^2$ is the elementwise square of the geodesic distance matrix $D = [D_{ij}]$, H is the centering matrix, given by $H = I_n - \frac{1}{N} e_N e_N^T$ where $e_N = [1 \dots 1]^T \in \mathbb{R}^N$

Isomap steps

- Construct neighborhood graph G
  - Compute the matrix $D_G = d_x(i,j)$
  - $d_x(i,j)$ is the euclidean distance between neighbors

Isomap steps

- Construct neighborhood graph G
  - Compute the matrix $D_G = d_x(i,j)$
  - $d_x(i,j)$ is the euclidean distance between neighbors
- Compute shortest paths between all pairs
  - Compute the matrix $D_G = d_G(i,j)$
  - D_G = sequence of hops = approx geodesic distance

Isomap steps

- Construct neighborhood graph G
  - Compute the matrix $D_G = d_x(i,j)$
  - $d_x(i,j)$ is the euclidean distance between neighbors
- Compute shortest paths between all pairs
  - Compute the matrix $D_G = d_G(i,j)$
  - D_G = sequence of hops = approx geodesic distance
- Construct K-dimensional coordinate vectors
  - Apply MDS to D_G instead of D_x

Isomap steps

- Construct neighborhood graph G
  - Compute the matrix $D_G = d_x(i,j)$
  - $d_x(i,j)$ is the euclidean distance between neighbors
- Compute shortest paths between all pairs
  - Compute the matrix $D_G = d_G(i,j)$
  - D_G = sequence of hops = approx geodesic distance
- Construct K-dimensional coordinate vectors
  - Apply MDS to D_G instead of D_x
- Like PCA, compute the eigenvectors and the corresponding eigenvalues of the centered distance

Isomap steps

- Construct neighborhood graph G
  - Compute the matrix $D_G = d_x(i,j)$
  - $d_x(i,j)$ is the euclidean distance between neighbors
- Compute shortest paths between all pairs
  - Compute the matrix $D_G = d_G(i,j)$
  - D_G = sequence of hops = approx geodesic distance
- Construct K-dimensional coordinate vectors
  - Apply MDS to D_G instead of D_x
- Like PCA, compute the eigenvectors and the corresponding eigenvalues of the centered distance
- choose two principales components

Implementation

Pros/Cons

- Pros
  - Nonlinear
  - Non-iterative
  - Globaly optimal

Pros/Cons

- Pros
  - Nonlinear
  - Non-iterative
  - Globaly optimal
- Drawbacks
  - Graph descreteness overestimates the geodesic distance
  - K must be high to avoid linear shortcuts near regions of high surface curvature
  - Isomap performs poorly when manifold is not well sampled and contains holes. As mentioned earlier neighborhood graph creation is tricky and slightly wrong parameters can produce bad results.

References

- http://axon.cs.byu.edu/papers/gashler2007nips.pdf
- https://en.wikipedia.org/wiki/Isomap https://blog.paperspace.com/dimension-reduction-with-isomap/
- http://benalexkeen.com/isomap-for-dimensionality-reduction-in-python/
- https://www.youtube.com/watch?v=RPjPLlGefzw
- https://www.quora.com/What-kind-of-datasets-is-ISOMAP-most-effective-on