San José State University

Math 250: Mathematical Data Visualization

Isometric Feature Mapping (ISOmap)

Dr. Guangliang Chen

Outline of the presentation

- Introduction to the manifold learning problem
- ISOmap
- Experiments

Introduction

Real data sets often display nonlinear geometry in Euclidean spaces such that linear dimension reduction methods (PCA, LDA) cannot work well.

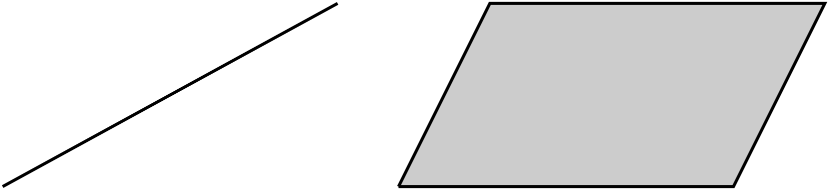
As a result, many nonlinear dimension reduction methods, which preserve different kinds of information in a manifold model, have been developed.

For example,

- **ISOmetric feature mapping (ISOmap)** $\longleftarrow$ this lecture
- **Locally linear embedding (LLE)** $\longleftarrow$ skipped (see book chapter)
- **Laplacian Eigenmaps (LM)** $\longleftarrow$ next lecture

What is a manifold?

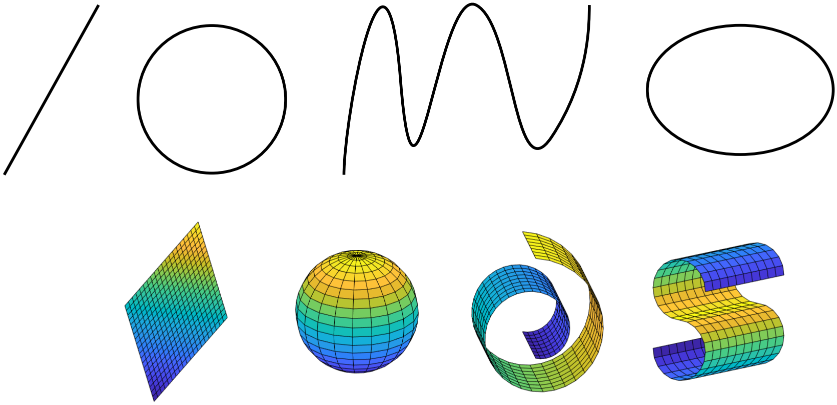
A line, plane, or any higher dimensional equivalent in a Euclidean space is a geometric object that locally has a fixed dimension and globally is flat.



If we relax the global flatness requirement and allows a plane to curve naturally in space, then we get a manifold.

ISOmap

Examples of 1-manifolds (top row) and 2-manifolds (bottom row)



ISOmap

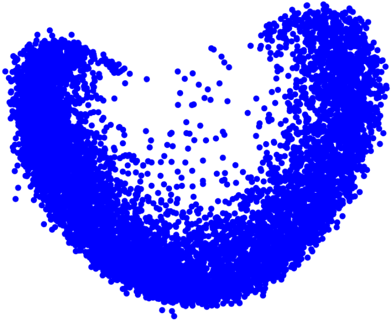
Remark.

- Manifolds can extend arbitrarily in high dimensional Euclidean spaces, spanning many dimensions and producing complex global geometries. This makes manifolds particularly suitable for modeling complex, nonlinear data that display simple local structures (i.e, flat and low dimensional).

- On the other hand, it has been observed that that many real data sets, though living in high dimensional Euclidean spaces, follow approximately along a low dimensional manifold.

ISOmap

Example: The MNIST handwritten digit 1



ISOmap

Let $\mathcal{M} \subset \mathbb{R}^d$ be a manifold with fixed local dimension k. We say that

- $\mathcal{M}$ is a k-dimensional manifold, or simply a k-manifold, in $\mathbb{R}^d$.
- k is the *intrinsic dimension* of the manifold (or the manifold dimension), and d the *ambient dimension* of the manifold.

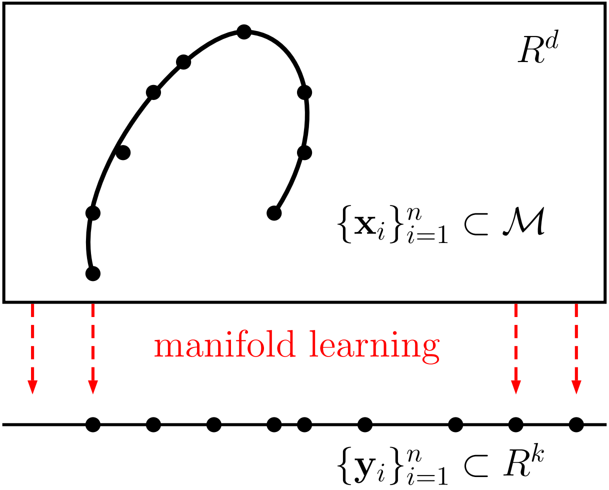
The manifold learning problem

Problem. Given a set of points along a k-manifold embedded in a high dimensional Euclidean space, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{M} \subset \mathbb{R}^d$ (where $\mathcal{M}$ is unknown), find another set of vectors in a low-dimensional Euclidean space, $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^k$, such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$ in some way.

Remark. $\{\mathbf{y}_i\}$ can be thought of internal coordinates of the points $\{\mathbf{x}_i\}$ in the manifold.

Remark. Manifold learning is also called **nonlinear dimensionality reduction** because manifolds are typically nonlinear and require nonlinear methods to identify the low dimensional intrinsic geometry.

ISOmap


R^d
$\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{M}$
manifold learning
$\{\mathbf{y}_i\}_{i=1}^n \subset R^k$

ISOmap

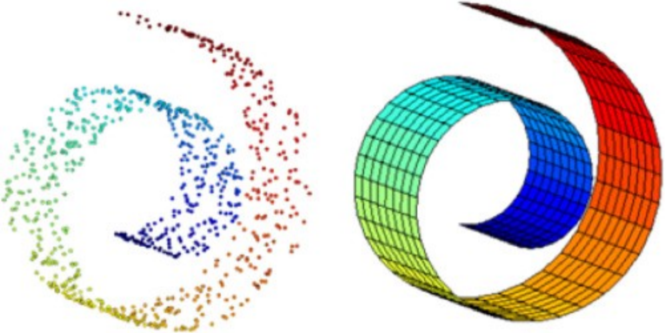
Briefly, ISOmap is MDS with a special metric, called **geodesic distance**, for reducing the dimensionality of data sampled from a smooth manifold:

- **Paper**: *A Global Geometric Framework for Nonlinear Dimensionality Reduction, J. B. Tenenbaum, V. de Silva and J. C. Langford, Science 290 (5500): 2319–2323, December 2000*
- **Webpage at Stanford**[1]

[1] https://web.archive.org/web/20161020154438/http://isomap.stanford.edu/

Motivation

Consider a sample from a manifold, e.g., Swissroll data.



(a) (b)

ISOmap

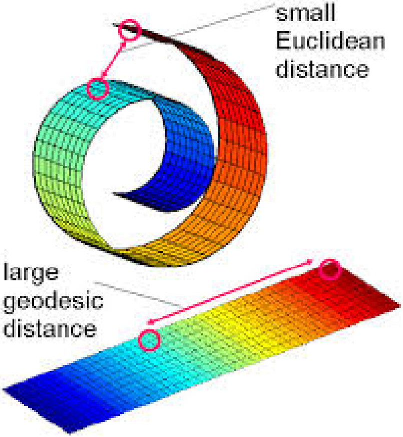
PCA has the following drawbacks:

- The PCA dimension needs to be higher, and sometimes much higher, than the manifold dimension (otherwise PCA may project faraway points along the manifold to nearby locations);
- PCA cannot capture the curved dimensions (its principal directions are generally not meaningful).

MDS + geodesic distance = ISOmap

Instead of preserving the Euclidean distance (i.e., PCA), one can apply MDS to preserve the **geodesic distance** along the manifold, which

- captures the true, nonlinear geometry corresponding to the curved dimension;
- allows to see the transitioning along the manifold (and thus the global structure).

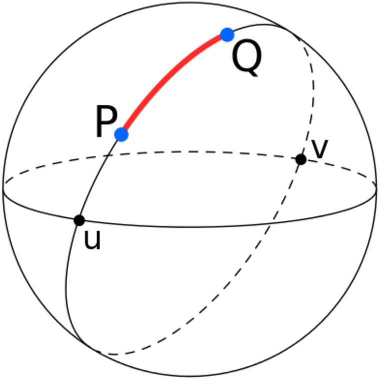


How to find geodesic distances

The geodesic distance of two data points on a manifold is the shortest distance along the manifold.

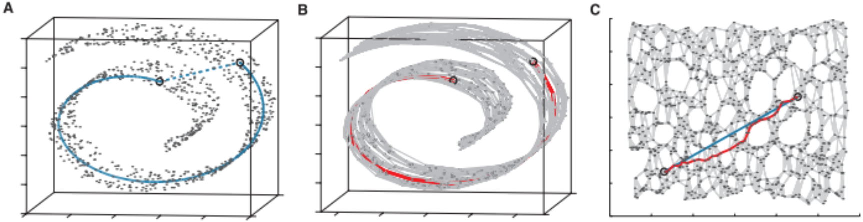
On a sphere, it is just the great-circle distance.

The exact geodesic distances are often impossible to find (unless we know the true manifold).



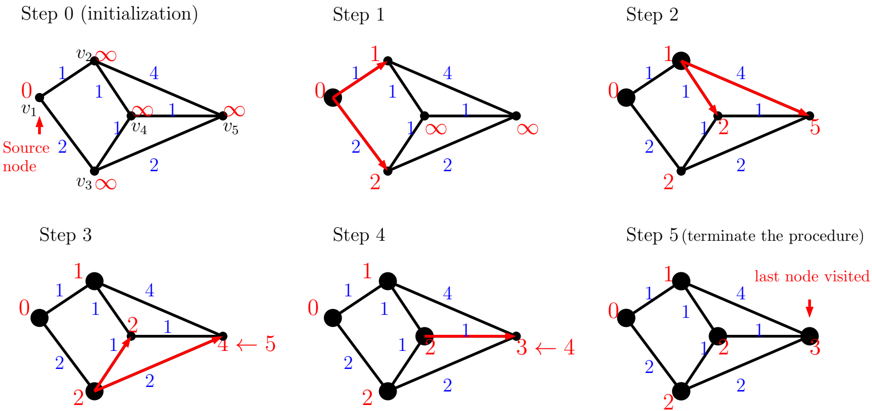
Strategy

In practical settings where we are only given a data set X sampled from an unknown manifold $\mathcal{M}$, we can approximate the true geodesic distances $d_{\mathcal{M}}(i,j)$ by the shortest-path distances $d_G(i,j)$ on a nearest-neighbor graph G built on the data set.



ISOmap

Dijkstra's algorithm for finding shortest-path distances:


Step 0 (initialization)

Step 1

Step 2

Step 3

Step 4

Step 5 (terminate the procedure)

Detailed steps

1. Build a neighborhood graph G from the given data by connecting only "nearby points" with edges weighted by their Euclidean distances, i.e.,

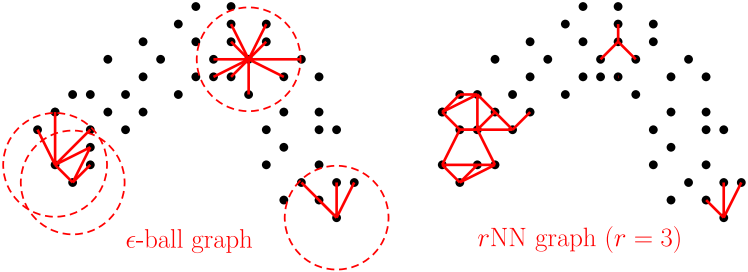
$$d_X(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\| \quad \text{if } \mathbf{x}_i, \mathbf{x}_j \text{ are "close" (and 0 otherwise)}$$

   where "closeness" is defined in one of the following ways:

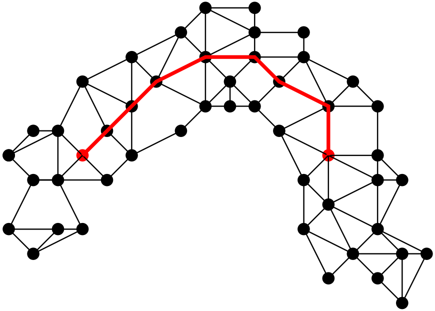
   - ϵ**-ball approach**: For each $\mathbf{x}_i$, another point $\mathbf{x}_j$ is close if and only if $\|\mathbf{x}_i - \mathbf{x}_j\| \le \epsilon$, or

ISOmap

- k**NN approach**: For each point $\mathbf{x}_i$, $\mathbf{x}_j$ is close if it is among the k nearest neighbors of $\mathbf{x}_i$.

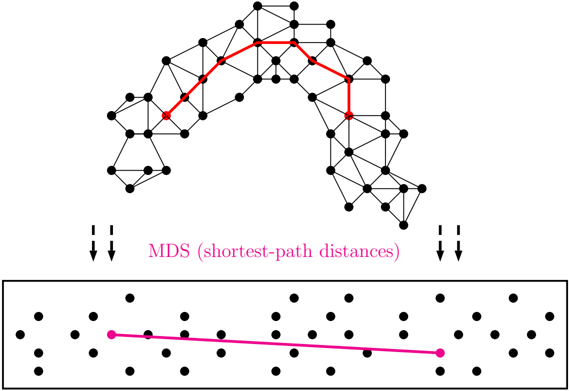


2. Apply Dijkstra's algorithm with the nearest neighbor graph G (constructed by either method) to find the shortest-path distances for all pairs of data points ($d_G(i,j)$).



ISOmap

3. Apply MDS with the shortest-path distances ($d_G(i,j)$) to find an embedding for the original data.



The ISOmap algorithm

Input: Pairwise distances $d_X(i,j)$ of data points in the input space, embedding dimension $k \geq 1$, neighborhood graph method (ϵ-ball or rNN)

Output: A k-dimensional representation of the data $\mathbf{Y} \in \mathbb{R}^{n\times k}$.

1. Construct a neighborhood graph G from the given distances $d_X(i,j)$ using the specified method
2. Compute the shortest-path distances $d_G(i,j)$ between all vertices of G by using Dijkstra's algorithm.
3. Apply MDS with $d_G(i,j)$ as input distances to find a k-dimensional representation $\mathbf{Y}$ of the original data

Implementations

MATLAB code by the authors is available from the ISOmap website.[2]

There are a few small spacing errors; I have fixed them and uploaded the corrected code to Canvas for you to download.

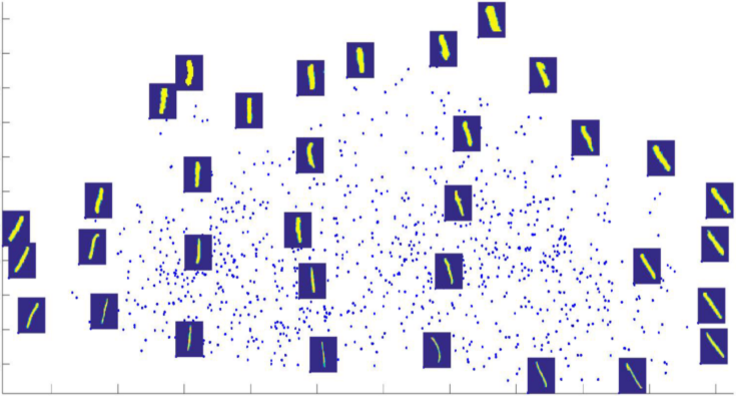
Python implementation: `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html`

[2] `https://web.archive.org/web/20161020154438/http://isomap.stanford.edu/`

Numerical issues

- ISOMAP is very slow on large data (use a subset).
- Needs a good estimate of the neighborhood size, r or ϵ.
- Assumes the given data set has no holes.
- ISOmap is prone to effects of short-circuiting.

Demonstration



More experiments

See

`https://www.cs.cmu.edu/~bapoczos/Classes/ML10715_2015Fall/slides/ManifoldLearning.pdf`

Summary

We have presented ISOmap as a nonlinear dimensionality reduction method.

Like PCA, it is a special instance of MDS:

- MDS + Euclidean distance: PCA
- MDS + geodesic distance: ISOmap