CSI 436/536
Introduction to Machine Learning

Dimension reduction: MDS & ISOMAP

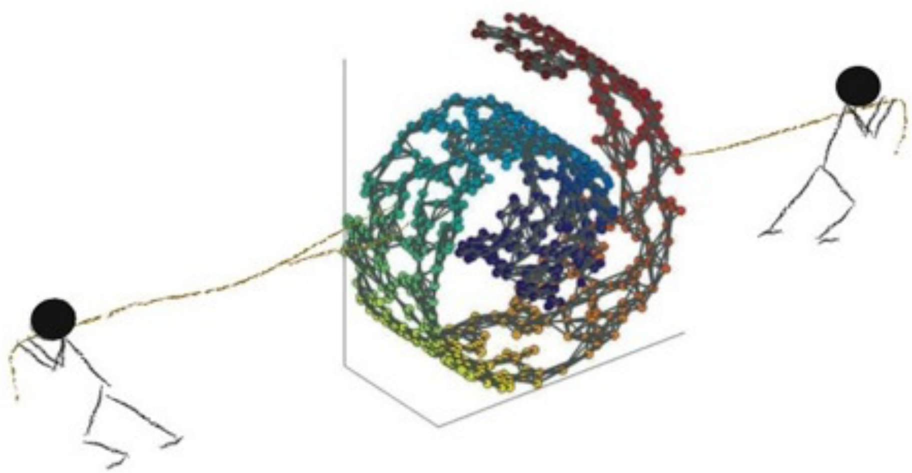
Professor Siwei Lyu
Computer Science
University at Albany, State University of New York

Dimension reduction

- For an input high dimensional data source $x \in \mathscr{R}^d$, find a low dimensional representation $\tilde{x} \in \mathscr{R}^m$ with $m \ll d$ that "best" approximate the original data
  - Determine a pair of transforms $\phi : \mathscr{R}^d \mapsto \mathscr{R}^m$ (***encoder***) and $\psi : \mathscr{R}^m \mapsto \mathscr{R}^d$ (***decoder***) such that $\tilde{x} = \phi(x)$, and $L(x - \psi(\tilde{x})) = L(x - \psi(\phi(x)))$ is minimized, where L is a loss function
  - Dimension reduction is an example of unsupervised learning problem (**self-supervised learning**)
- The dimensionality constraint is served as an ***information bottleneck***, filtering out less relevant information as discarded dimension

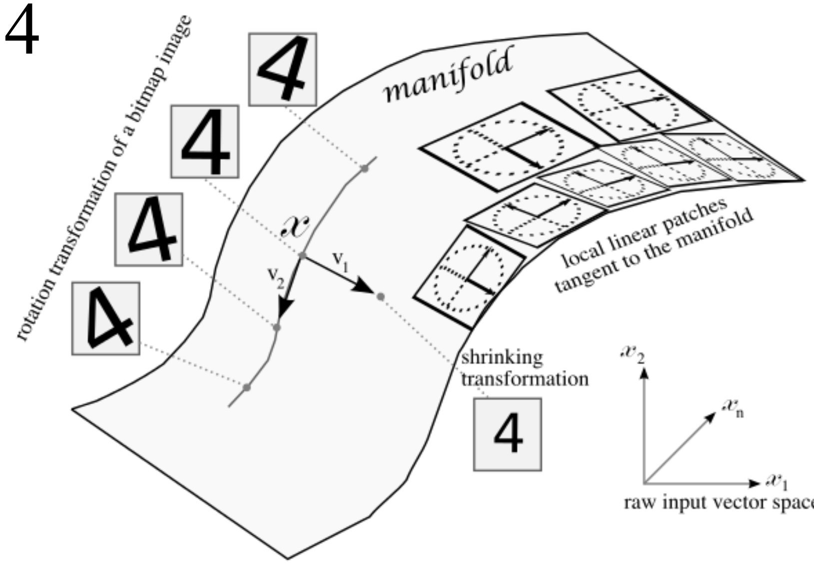
Nonlinear dimension reduction

- When we choose the encoder and decoder as nonlinear functions, it is nonlinear dimension reduction
- All data in the d-dimensional space is fully represented by points in an m-dimensional space ***non-linearly*** embedded in the d-dimensional space
- A low dimensional subspace nonlinearly embedded in the high dimensional space can be modeled as a ***manifold***, nonlinear dimension reduction aims to recover the m-dimensional subspace



Examples of nonlinear manifolds

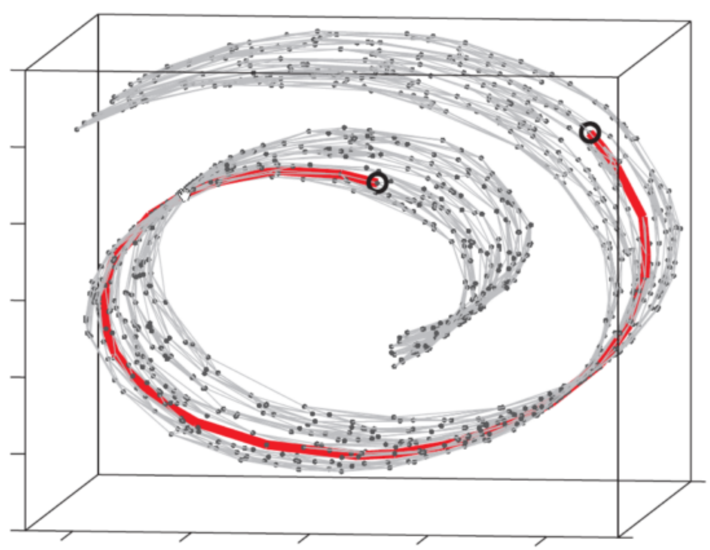
- Consider all images of number 4
  - Each image is treated as a point in a high-dimensional space as vectorized pixel values



  - All images of number 4 with different rotation angles are related by a smooth path, corresponding to different angles
- If we recover this intrinsic low dimensional manifold, it helps to understand the structure in this dataset
  - Synthesis: generate data of given configuration
  - denoising/projection: find closest examples on the manifold close to an input

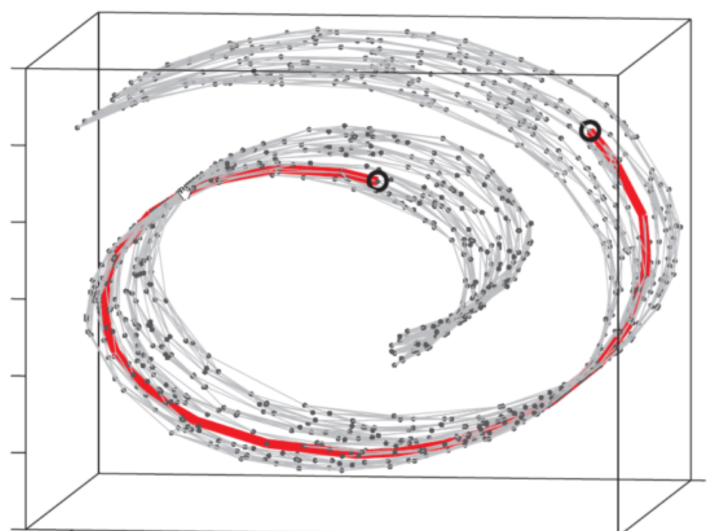
Manifold

- A mathematical (differential geometric) entity that is locally described with linear space (tangent space)
  - Manifold is smooth (differentiable)
  - At the adjacency of any point on the manifold, it can be closely approximated by a linear space (tangent space)
  - Globally it has a nonlinear structure



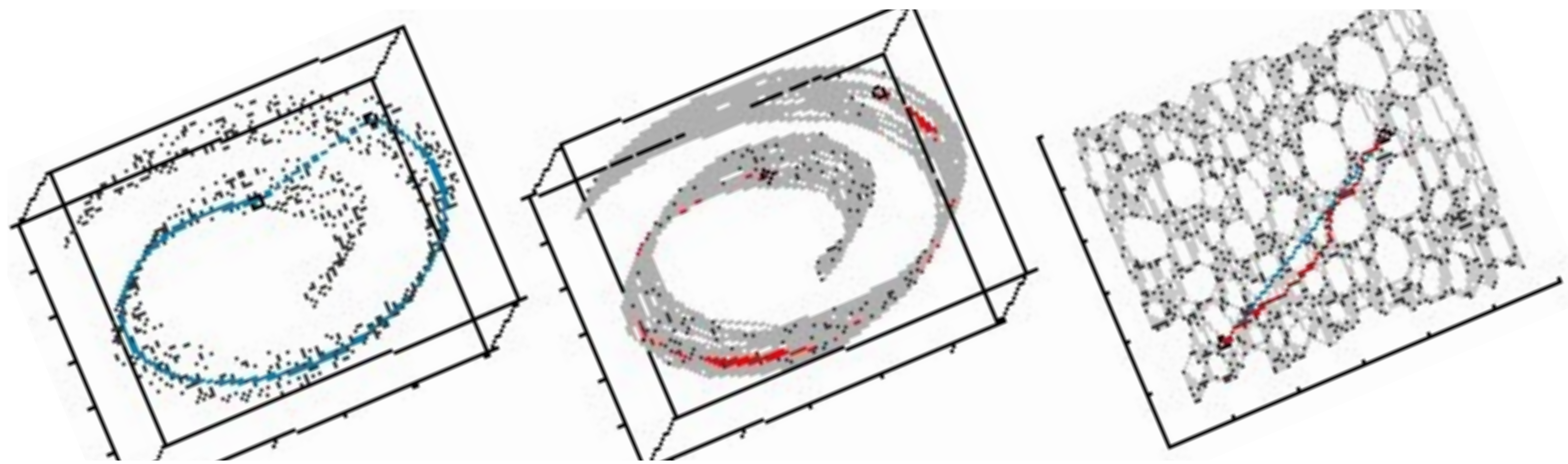
Manifold

- The curve corresponding to the shortest distance between any two points on a manifold is known as the **geodesic** curve
  - In a linear space, the geodesic is a straight line
  - In curved manifold, the geodesic is usually nonlinear and different from a straight-line in the ambient space
    - ex. The great arc on the surface of the earth
- If we can recover the correct geodesic distance between any pair of points, we can recover the nonlinear manifold
  - The algorithm is known as ISOMAP [Tenebaum et.al., 2005]



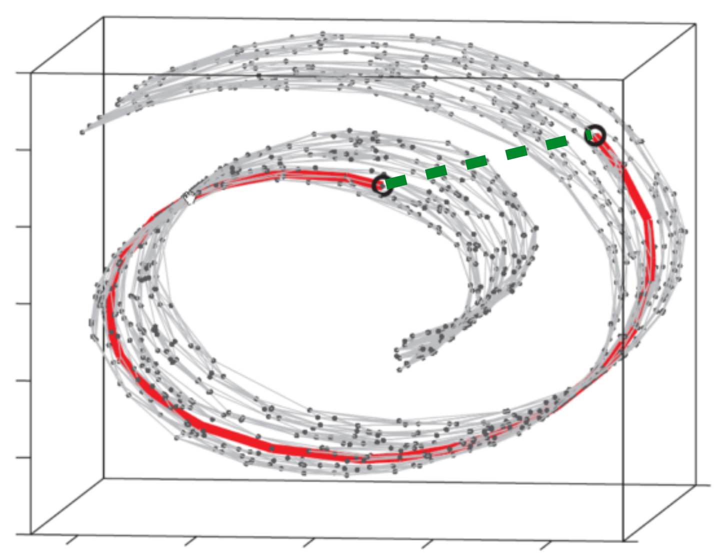
ISOMAP

- ISOMAP assume a set of high dimensional data points are determined by a low dimensional nonlinear manifold
- The basic idea of ISOMAP is to estimate the geodesic distance from a finite dataset
- Then from all pair geodesic distance we can obtain the Gram matrix, and further recover the low dimensional data representation



Estimating geodesic distances

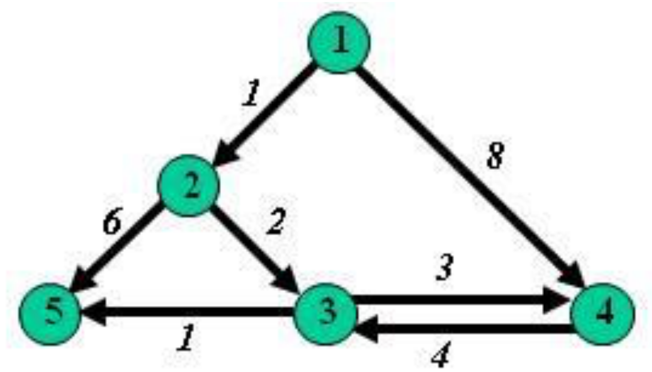
- Construct a graph using the top k-nearest neighbors of every data point in the set [k is a hyper-parameter]
  - The weight of each edge is the Euclidean distance between the two points
- Instead of using their direct Euclidean distance, we measure the distance between any two points using the shortest path between them
  - This gives an approximation to the geodesic distance of the two points on the surface of the manifold



Floyd algorithm

- The Floyd algorithm finds the shortest paths between any pair of nodes in a weighted undirected graph with a running time of $O(n^3)$, for n being the total number of nodes in a graph
  - a dynamic programming algorithm

```
Initialize
for k=1 to n
   for i=1 to n
      for j=1 to n
         if Dist[i,j] > Dist[i,k] + Dist[k,j]
            then Dist[i,j] ← Dist[i,k] + Dist[k,j]
```



- The result is an n-by-n matrix containing pairwise distances for the nodes on the graph
  - This matrix is known as the ***distance matrix***

MDS

- The geodesic distance between two points on the manifold corresponds to the Euclidean distance between the two points on the "flattened" manifold
- We can recover the coordinates of the points on the manifold using such pairwise distances if we assume data on the flattened manifold is centered
  - $X1 = 0$, so $G1 = X^TX1 = 0$
- we use the squared distance matrix to obtain low dimensional representation, this process is known as the ***multi-dimensional scaling*** (MDS) algorithm

From distance matrix to Gram matrix

- Distance matrix: D_{ij} = squared Euclidean distance between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$
- Gram matrix: $G = X^T X$, or $G_{ij} = \mathbf{x}_i^T \mathbf{x}_j$, inner products between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$
- Relation between distance matrix and Gram matrix

$$D = diag(G)1^T + 1diag(G)^T - 2G$$

- Then we can obtain

$$G = -\frac{1}{2}\left(I - \frac{1}{n}11^T\right) D \left(I - \frac{1}{n}11^T\right)$$

- this procedure is called ***double centering***, i.e., it centers a matrix across both rows and columns

Derivations

- First, $D_{ij} = (x_i - x_j)^T(x_i - x_j) = x_i^T x_i - 2x_i^T x_j + x_j^T x_j$, or $D_{ij} = G_{ii} - 2G_{ij} + G_{jj}$, put in the form of matrices, we get $D = diag(G)1^T + 1diag(G)^T - 2G$ ————(*)
- Multiply both sides by vector 1 and assume G1 = 0 (centered data), we have $D1 = diag(G)1^T1 + 1diag(G)^T1 = ndiag(G) + diag(G)^T11$
- Multiply by vector 1 on the left $1^TD1 = 2n1^Tdiag(G)$
- Put this back $D1 = diag(G)1^T1 + 1diag(G)^T1 = ndiag(G) + \dfrac{1}{2n}1^TD11$
- Now we have $diag(G) = \dfrac{1}{n}D1 - \dfrac{1}{2n^2}1^TD11$ and putting this back to (*) and with some algebraic manipulation shows the result

Obtaining low dimensional representation

- With the Gram matrix, we aim to further recover the low dimensional representation
- $G = X^T X$ is a symmetric and PSD matrix, so according to the spectral theorem, it can be decomposed as $G = U \Gamma U^T$, where U is an orthonormal matrix, Γ is a diagonal matrix containing nonnegative eigenvalues of G
- We can then recover data representation X by decomposing G as $G = U \Gamma^{½} \Gamma^{½} U^T$, so setting $X = \Gamma½ U^T$, we get data low dimensional representation
  - It is not unique, there are many similar decompositions
- We obtain a low dimensional representation of the data
- New data points can be projected on the manifold by interpolation

ISOMAP summary

- advantage: theoretical guarantee of performance
- drawback: sensitivity to hyper-parameter choices (degree of neighbors)