MATH466/MATH766
Math of machine learning
03/19 Lecture 16 Graph, Isomap and Laplacian matrix

References:

- Isomap: Tenenbaum, Joshua B., Silva, Vin de, and Langford, John C. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction
- Mikhail Belkin and Partha Niyogi, 2003, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation
- Ulrike von Luxburg, A tutorial on spectral clustering https://people.csail.mit.edu/dsontag/courses/ml14/notes/Luxburg07_tutorial_spectral_clustering.pdf

Todays contents:

- build graph from point cloud
- Isomap
- affinity matrix
- degree matrix
- Laplacian matrix
- Spectral clustering

Important concepts:

- distance on the graph
- affinity matrix, degree matrix and Laplacian matrix

Recommend reading:

- Ulrike von Luxburg, A tutorial on spectral clustering https://people.csail.mit.edu/dsontag/courses/ml14/notes/Luxburg07_tutorial_spectral_clustering.pdf

Similarity Graph

Undirected graph $G = (V, E)$

- vertex set $V = \{v_1, \cdots, v_n\}$
- each edge between two vertices v_i, v_j carries a nonnegative weight $w_{ij} \geq 0$

  $w_{ij} = 0$ means v_i and v_j are not connected by an edge

  $w_{ij} = w_{ji}$ since G is undirected

  adjacency matrix $W = (w_{ij}) \in \mathbb{R}^{n \times n}$ is symmetric
- degree of a vertex $v_i \in V$ is $d_i = \sum_{j=1}^{n} w_{ij}$

  degree matrix $D = \text{diag}(d_1, \cdots, d_n) \in \mathbb{R}^{n \times n}$

ways to construct similarity graph from data

denote the "distance" between two vertices as $d(v_i, v_j)$

e.g. $d(v_i, v_j) = \|v_i - v_j\|$

- ε-neighbourhood graph

$$w_{ij} = \begin{cases} 1, & d(v_i, v_j) \leq \varepsilon \\ 0, & d(v_i, v_j) \leq \varepsilon \end{cases}$$

$s(v_i, v_j) = \exp(-d^2(v_i, v_j)/2\sigma^2)$

- k-nearest neighbourhood graph

$$w_{ij} = \begin{cases} s(v_i, v_j) \\ 0 \end{cases}$$

if v_i is among the k-NN of v_j

or if v_j ——— ——— of v_i

o.w.

mutual k-NN graph

$$w_{ij} = \begin{cases} s(v_i, v_j) \\ 0 \end{cases}$$

if v_i is among the k-NN of v_j

and if v_j ——— ——— of v_i

o.w.

- fully connected graph

$$w_{ij} = s(v_i, v_j)$$

want higher weight on edges when d_{ij} is small (i, j are close)

Demo

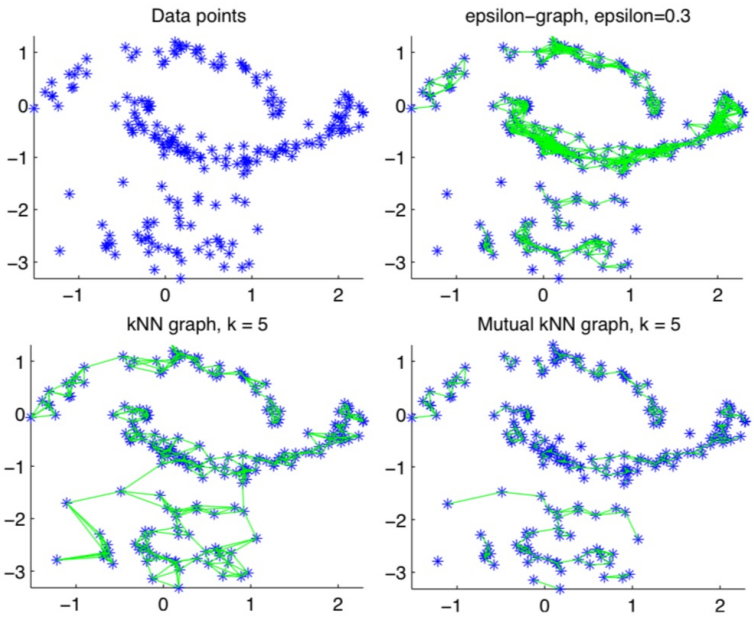

Figure 3: Different similarity graphs, see text for details.

Isomap

Procedure : ① build a weighted graph using nearest neighbours

② Compute pairwise shortest distance on the graph
(denote the distance matrix as S)

③ $G = -\frac{1}{2} J S J$ (analogue of gram matrix)

④ compute the leading eig-val, eig-vecs of G

Essentially, ISOMAP is applying classical MDS
to the graph distance matrix S

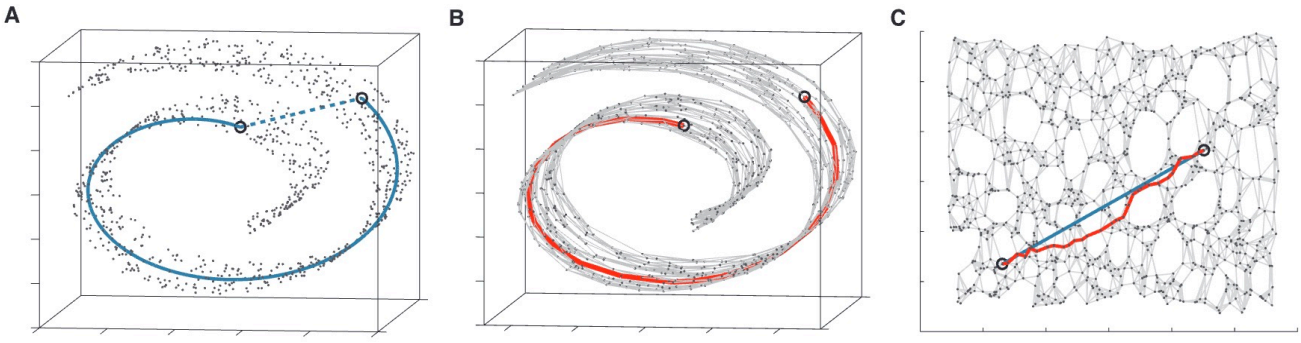

Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

Spectral Clustering

1. Connected components

$A \subseteq V$, $\bar{A} = V \setminus A$, $1_A = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$ w/. $f_i = \begin{cases} 1, & v_i \in A \\ 0, & v_i \notin A \end{cases}$

A is a **connected** if any two vertices in A can be joined by a path such that all intermediate points also lie in A

~~$\Rightarrow$~~ $\forall\, v_i, v_j \in A,\ w_{ij} > 0$

e.g. 

A is a connected component if it is connected and there are no connections b/t vertices in A and $\bar{A}$

The non-empty sets $A_1, \dots, A_k$ form a partition of G if $A_i \cap A_j = \phi$ and $A_1 \cup \dots \cup A_k = V$.

2. Graph Laplacian (Demo)

- unnormalized graph Laplacian $L = D - W$

properties: 1. for every vector $f \in \mathbb{R}^n$,

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

2. L is symmetric and p.s.d.

3. The smallest eigenvalue of L is 0, the corresponding eigenvector is $1_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$

Denote the eigenvalues of L as $0 = \lambda_1 \le \lambda_2 \le \dots \le \lambda_n$

★ # of connected components and the spetrum of L

Prop. Let G be an undirected graph w/. non-negative weights.

Then the multiplicity k of the eigenvalue 0 of L

= the # of connected components $A_1, \dots, A_k$ in the graph

The eigenspace of eigenvalue 0 is spanned by $1_{A_1}, \dots, 1_{A_k}$

Pf. Assume $Lf = 0$

then $0 = f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$

$\Rightarrow$ if $w_{ij} > 0$ then $f_i = f_j$

$\Rightarrow$ if v_i, v_j are connected by a path, then $f_i = f_j$

- normalized graph Laplacian

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W \quad \text{(HW)}$$

↑ real-diagonalizable b/c similar to a real-diagonalizable matrix

3. Spectral Clustering

Given $\{x_i\}_{i=1}^n$, compute $s(x_i, x_j)$, form L, or L_{sym}, L_{rw}.

(1) ideal case. clusters <-> connected components. Take $k=2$ as e.g.

compute eigenvectors corresponding to 0

	$u_1 \in \mathbb{R}^n$	$u_2 \in \mathbb{R}^n$	
y_1	1	0	$y_i \in \mathbb{R}^2$
y_2	1	0	gathering at $(1,0)$ and $(1,1)$
y_3	1	1	
y_4	1	1	clusters can be easily obtained
$\vdots$	$\vdots$	$\vdots$	through y_i

- in general, compute first k eigenvectors $u_1, \cdots, u_k$ of L

$0 = \lambda_1 \leq \cdots \leq \lambda_k$

$$U = [u_1, \cdots, u_k] \in \mathbb{R}^{n \times k}$$

$$= \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix}$$

run k-means to cluster $\{y_i\}$ to k clusters.

rmk if the multiplicity of 0 is 1, then we can omit u_1

(2) Noisy case (NOT Required)

① Perturbation Theory p.o.v. $\tilde{L} = L + H$, $L, H, \tilde{L} \in \mathbb{R}^{n \times n}$ sym

Weyl's Thm (perturbation of eigen-values)

$$|\lambda_k(\tilde{L}) - \lambda_k(L)| \le \max(|\lambda_1(H)|, |\lambda_n(H)|)$$

Davis-Kahan Thm (perturbation of eigen-spaces)

- distance b/t two subspaces of $\mathbb{R}^d$ $A_r = \text{span}\{a_1, \cdots, a_r\}$ o.n. basis
  (r < s WLOG, r, s ≤ d) $B_s = \text{span}\{b_1, \cdots, b_s\}$

SVD of $A_r^T B_s$: $U \cos\textcircled{H} V^T$

$$\cos\textcircled{H} = \begin{bmatrix} \cos\theta_1 & & & \\ & \cos\theta_2 & & 0 \\ & & \ddots & \\ & & & \cos\theta_r \end{bmatrix}$$

$$d(A_r, B_s) = \max_j |\sin\theta_j|$$

- EVD: L: $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$, $\psi_1, \psi_2, \cdots, \psi_n$

  $\tilde{L}$: $\tilde{\lambda}_1 \ge \tilde{\lambda}_2 \ge \cdots \ge \tilde{\lambda}_n$, $\tilde{\psi}_1, \tilde{\psi}_2, \cdots, \tilde{\psi}_n$

  $V_r = \text{span}\{\psi_1, \cdots, \psi_r\}$

  $\tilde{V}_r = \text{span}\{\tilde{\psi}_1, \cdots, \tilde{\psi}_r\}$

(i) $\delta := |\lambda_r - \hat{\lambda}_{r+1}|$

$$d(V_r, \tilde{V}_r) \le \frac{\|H\|_F}{\delta}$$

(ii) $\delta := \min(|\lambda_r - \hat{\lambda}_{r+1}|, |\lambda_r - \tilde{\lambda}_{r-1}|)$

$$|\sin(\psi_r, \tilde{\psi}_r)| \le \frac{\|H\|_F}{\delta}$$

② Graph Cut p.o.v

$W(A, B) := \sum_{i \in A, j \in B} w_{ij}$ $\quad |A| :=$ # of vertices in A

$\text{vol}(A) := \sum_{i \in A} d_i$

$\text{cut}(A_1, A_2, \cdots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i)$ simple to solve but not ideal for clustering

$\text{RatioCut}(A_1, A_2, \cdots, A_k) := \sum_{i=1}^{k} \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$ Try to balance clusters

$\text{Ncut}(A_1, A_2, \cdots, A_k) := \sum_{i=1}^{k} \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}$ but not easy to solve

Reformulation of the problem ($k=2$)

partition $A, \bar{A}$ $\Leftrightarrow$ $f \in \mathbb{R}^n$ of form $f_i = \begin{cases} \sqrt{|\bar{A}|/|A|}, & v_i \in A \\ -\sqrt{|A|/|\bar{A}|}, & v_i \in \bar{A} \end{cases}$

(i) $f^T 1_n = \sum_{i \in A} \sqrt{|\bar{A}|/|A|} - \sum_{i \in \bar{A}} \sqrt{|A|/|\bar{A}|} = 0$

(ii) $\|f\|_2^2 = \sum_{i \in A} |\bar{A}|/|A| + \sum_{i \in \bar{A}} |A|/|\bar{A}| = |A| + |\bar{A}| = n$

(iii) $f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$

$= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} (f_i - f_j)^2 + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} (f_i - f_j)^2$

$= \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 \text{cut}(A, \bar{A})$

$= \left(\frac{|\bar{A}|}{|A|} + 1 + \frac{|A|}{|\bar{A}|} + 1 \right) \text{cut}(A, \bar{A})$

$= n \left(\frac{\text{cut}(A, \bar{A})}{|A|} + \frac{\text{cut}(A, \bar{A})}{|\bar{A}|} \right)$

$= n \, \text{Ratio Cut}(A, \bar{A})$

$\min_f \; f^T L f \quad \text{s.t.} \quad f^T 1 = 0, \; \|f\| = \sqrt{n}, \; f \text{ derived from some } A$

NP hard

↓ relaxation

$\min_f \; f^T L f \quad \text{s.t.} \quad f^T 1 = 0, \; \|f\| = \sqrt{n}$

$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \bar{A} & \text{if } f_i < 0 \end{cases}$ or $\begin{cases} v_i \in A & \text{if } f_i \in C \\ v_i \in \bar{A} & \text{if } f_i \in \bar{C} \end{cases}$