CSE 291B 2/24

Spectral clustering (review)

How to map $\vec{x}_i \in \mathbb{R}^D$ to $y_i \in \{1, 2, \dots, k\}$?

Step 1. Construct weighted graph:

$$W_{ij} = \exp\left\{-\frac{1}{2\sigma^2}\|\vec{x}_i - \vec{x}_j\|^2\right\} \quad \text{affinity matrix}$$

$$D_{ii} = \sum_j W_{ij}$$

Step 2. Solve eigenvalue problem

$$\min \left[\frac{\sum_{ij} W_{ij}(y_i - y_j)^2}{\sum_i D_{ii} y_i^2}\right] \quad \text{subject to } \sum_i y_{ii} D_{ii} = 0$$

$$\Longrightarrow \quad (D-W)y = \lambda y$$

Step 3. For $k=2$, threshold elements of eigenvector $\vec{y}_i$ with smallest non-zero eigenvalue λ_1.

For $k>2$, recurse on $k=2$ solution.

- Application #1: graph partitioning by normalized cut

$$\text{cut}(A,B) = \sum_{i\in A}\sum_{j\in B} W_{ij}$$

$$\text{NCut}(A,B) = \frac{\text{cut}(A,B)}{|A|} + \frac{\text{cut}(A,B)}{|B|} \quad \text{where } |A| = \sum_{i\in A} D_{ii}$$

- Application #2: image segmentation

segmentation can be viewed as problem in pixel clustering

Pixel affinities can be computed from spatial proximity, color matching, texture matching, etc...

E.g. let $\phi_i = (x_i, y_i, r_i, g_i, b_i)$ store xy-location and rgb-value of ith pixel

let $W_{ij} = \exp\left[-\frac{1}{2\sigma}\|\phi_i - \phi_j\|^2\right]$

Carefully constructed affinity matrices lead to state-of-the-art results.

- Physical inuition

  Potential energy $\frac{1}{2}\sum_{ij} W_{ij}(y_i - y_j)^2$ "springs"

  Kinetic energy $\frac{1}{2}\sum_i D_{ii}\, y_i^2$ "masses"

  Spectral clustering computes normal modes of mechanical system.

- Probabilistic intuition

  $P = D^{-1}W$ is a stochastic matrix $\sum_j P_{ij} = 1$

  Let π denote stationary distribution : $\pi^T P = \pi^T$. (π is a vector)

  $\text{Ncut}[A,B] = \Pr[A \to B \mid A] + \Pr[B \to A \mid B]$ under P.

  Min $\text{Ncut}[A,B]$ looks for clusters that "trap" the random walk.

$\boxed{\text{Manifold learning}}$

How to map $\vec{x}_i \in \mathbb{R}^D$ to $\vec{y}_i \in \mathbb{R}^d$

Laplacian eigenmaps

- Step 1. Compute kNN of inputs $\vec{x}_i \in \mathbb{R}^D$
- Step 2. Construct sparse weighted graph

$$W_{ij} = \begin{cases} 0 & \text{if } \vec{x}_i \text{ and } \vec{x}_j \text{ \underline{aren't} kNN} \\ e^{-\|\vec{x}_i - \vec{x}_j\|^2/(2\sigma^2)} & \text{if } \vec{x}_i \text{ and } \vec{x}_j \text{ \underline{are} kNN} \end{cases}$$

- Step 3. Solve eigenvalue problem

  Minimize $\displaystyle\sum_{ij} \frac{W_{ij}\|\vec{y}_i - \vec{y}_j\|^2}{\sqrt{D_{ii} D_{jj}}}$ where $D_{ii} = \sum_j W_{ij}$

  subject to
$$\begin{cases} \sum_i \vec{y}_i = \vec{0} & \text{(centering)} \\ \frac{1}{N}\sum_i \vec{y}_i \vec{y}_i^T = I_d & \text{(orthogonality)} \end{cases}$$

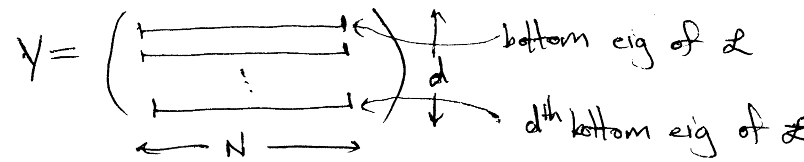
Why? Minimizing above favors locality-preserving embeddings.

Let $Y = [\vec{y}_1\ \vec{y}_2 \ldots \vec{y}_N] \in \mathbb{R}^{d \times N}$

$\mathcal{L} = I_N - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ normalized graph Laplacian

Then $\displaystyle\sum_{ij} \frac{W_{ij}\|\vec{y}_i - \vec{y}_j\|^2}{\sqrt{D_{ii}}\sqrt{D_{jj}}} = \mathrm{tr}[Y \mathcal{L} Y^T]$

Solution: output d eigenvectors of $\mathcal{L}$ with d smallest non-zero eigenvalues



- Relation to spectral clustering

  Same eigenvalue problem, but different output:
  - bottom $d+1$ eigenvectors are computed
  - no thresholding

- Relation to Isomap and MVU
  - bottom versus top eigenvectors
  - locality-preserving vs. local-distance-preserving
  - sparse vs. dense eigenvalue problem
  - eigenvalues of $\mathcal{L}$ do <u>not</u> reveal intrinsic dimensionality

- Spectral graph theory

  Let f_i denote function over nodes i of graph.

  Graph laplacians measure smoothness of functions on graphs:

$$\sum_{ij} \frac{(f_i - f_j)^2 W_{ij}}{\sqrt{D_{ii} D_{jj}}} = f^T \mathcal{L} f$$

  Eigenvectors of $\mathcal{L}$ provide ordered basis for functions on graph:

  smaller eigenvalues $\longleftrightarrow$ smoother eigenfunctions

- Embeddings

  Laplacian eigenmaps return d smoothest, non-constant orthogonal functions on graph as low-dimensional coordinates $\vec{y}_i \in \mathbb{R}^d$.

Smoothest functions on graph $\Longleftrightarrow$ best locality-preserving coordinates

What else might we try to preserve?

Locally linear embedding (LLE)

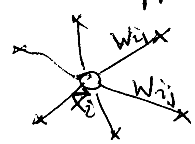
- Key idea

  Map $\vec{x}_i \in \mathbb{R}^D$ to $\vec{y}_i \in \mathbb{R}^d$ by preserving local (but not global) linear relationships.

- Reconstruction weights

  Consider weights W_{ij} that approximately reconstruct $\vec{x}_i$ from its kNN:

  $$\vec{x}_i \approx \sum_{j \in kNN} W_{ij} \vec{x}_j$$



Accurate linear reconstructions will exist if $\vec{x}_i \in \mathbb{R}^D$ lie on (or near) a d-dimensional manifold with $d < k$.

- Least squares fits

  $$\min \|\vec{x}_i - \sum_j W_{ij}\vec{x}_j\|^2 \quad \text{such that} \begin{cases} W_{ij}=0 \text{ if } \vec{x}_j \text{ is not a kNN of } \vec{x}_i \\ \sum_j W_{ij} = 1 \end{cases}$$

  Why impose sum-to-one constraint?

  Ex: Suppose $\vec{x}_i = 0$.

  Then $\|\vec{x} - \sum W_{ij}\vec{x}_j\|^2 = 0$ for $W_{ij}=0$, regardless of $\vec{x}_j$.

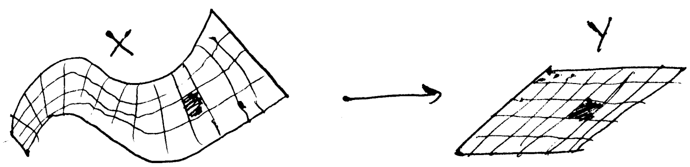
  Sum-to-one constraint makes optimal weights (e.g., $(\frac{1}{2}, \frac{1}{2})$) invariant to location of origin.

- Invariants

  Optimal weights W_{ij} encode information about local geometry.

  $\underset{\sum_j W_{ij}=1}{\operatorname{argmin}} \|\vec{x}_i - \sum_j W_{ij}\vec{x}_j\|^2$ is invariant to translation, rotation, scaling of local neighborhood

- Mapping from $\vec{x}_i \in \mathbb{R}^D$ to $\vec{y}_i \in \mathbb{R}^d$



Global mapping is nonlinear.

Suppose local mapping looks like translation, rotation plus scaling.

Then weights W_{ij} that reconstruct $\vec{x}_i$ from kNN $\vec{x}_j$ should also reconstruct $\vec{y}_i$ from corresponding $\vec{y}_j$.

How can we use this idea to compute a low-dimensional representation y of high dimensional x with same local linear structure encoded by optimal weights W?

<u>Next lecture.</u>