Dimension Reduction and Nyström Method

Jiaxin Shi

Tsinghua University

April 10, 2018

Outline

Outline

Outline

Recap: Rayleigh Quotient

- $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{A} \in \mathbb{R}^{d \times d}$

$$R(\mathbf{A}, \mathbf{x}) = \frac{\mathbf{x}^\top \mathbf{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \tag{1}$$

- Eigenvalues of $\mathbf{A}$:
  - $\lambda_{min} = \min_{\mathbf{x}} R(\mathbf{A}, \mathbf{x})$, corresponding eigenvector $\mathbf{u}_{min}$
  - $\lambda_{max} = \max_{\mathbf{x}} R(\mathbf{A}, \mathbf{x})$, corresponding eigenvector $\mathbf{u}_{max}$
- Equivalent problem:

$$\max_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}\mathbf{x} \quad \text{s.t.} \quad \mathbf{x}^\top \mathbf{x} = 1 \quad \Rightarrow \quad \mathbf{x}^* = \mathbf{u}_{max} \tag{2}$$

- Matrix form, $\mathbf{U} \in \mathbb{R}^{d \times k}$:

$$\max_{\mathbf{U}} tr(\mathbf{U}^\top \mathbf{A}\mathbf{U}) \quad \text{s.t.} \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I} \quad \Rightarrow \quad \mathbf{U}^* = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \tag{3}$$

$[\mathbf{u}_1, \ldots, \mathbf{u}_k]$ are the eigenvectors corresponding to the largest k eigenvalues $\lambda_1 \geq \cdots \geq \lambda_k$.

Recap: Principle Component Analysis (PCA)

- Dataset: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]^\top$, sample mean: $\boldsymbol{\mu} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m$.
- PCA finds the directions $\mathbf{u}$ that maximize the variance of projected data:

$$\frac{1}{M}\sum_{m=1}^{M}\left[\mathbf{u}^\top(\mathbf{x}_m - \boldsymbol{\mu})\right]^2 = \mathbf{u}^\top\mathbf{C}\mathbf{u} \tag{4}$$

  $\mathbf{C}$ is the sample covariance

$$\mathbf{C} = \frac{1}{M}\sum_{m=1}^{M}(\mathbf{x}_m - \boldsymbol{\mu})(\mathbf{x}_m - \boldsymbol{\mu})^\top \tag{5}$$

- The problem

$$\max_{\mathbf{u}} \mathbf{u}^\top\mathbf{C}\mathbf{u} \quad \text{s.t.} \quad \mathbf{u}^\top\mathbf{u} = 1 \tag{6}$$

- Matrix form

$$\max_{\mathbf{U}} tr(\mathbf{U}^\top\mathbf{C}\mathbf{U}) \quad \text{s.t.} \quad \mathbf{U}^\top\mathbf{U} = \mathbf{I} \tag{7}$$

Recap: Principle Component Analysis (PCA)

- Centered dataset: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]^\top$
- The sample covariance matrix: $\mathbf{C} = \frac{1}{M}\mathbf{X}^\top\mathbf{X}$
- The PCA eigenvalue problem:

$$\mathbf{C}\mathbf{U} = \mathbf{U}\Lambda, \tag{8}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$.

Dual PCA

PCA:

$$\mathbf{C}\mathbf{v} = \mu\mathbf{v}, \tag{9}$$

$$\mathbf{X}^\top\mathbf{X}\mathbf{v} = M\mu\mathbf{v}. \tag{10}$$

Left multiplying $\mathbf{X}$:

$$\mathbf{X}\mathbf{X}^\top(\mathbf{X}\mathbf{v}) = M\mu(\mathbf{X}\mathbf{v}), \tag{11}$$

Let $\boldsymbol{\alpha} = \mathbf{X}\mathbf{v}$, from eq. (10):

$$\mathbf{v} = \frac{1}{M\mu}\mathbf{X}^\top\boldsymbol{\alpha}. \tag{12}$$

Plugging eq. (12) into eq. (11), we have

$$\mathbf{X}\mathbf{X}^\top\mathbf{X}\mathbf{X}^\top\boldsymbol{\alpha} = M\mu\mathbf{X}\mathbf{X}^\top\boldsymbol{\alpha} \tag{13}$$

turns out an eigenvalue problem for $\mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{M\times M}$, the Gram matrix.

Outline

Kernel PCA (KPCA)

- Consider doing PCA in the feature space induced by the mapping ϕ:

$$\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_M)]^\top, \tag{14}$$

$$\mathbf{C} = \frac{1}{M}\mathbf{\Phi}^\top\mathbf{\Phi}, \tag{15}$$

$$\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{v} = M\mu\mathbf{v}. \tag{16}$$

- The dual form:

$$\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{\Phi}^\top\boldsymbol{\alpha} = M\mu\mathbf{\Phi}\mathbf{\Phi}^\top\boldsymbol{\alpha}. \tag{17}$$

- Applying the kernel trick $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$:

$$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top, \tag{18}$$

we have

$$\mathbf{K}\mathbf{K}\boldsymbol{\alpha} = M\mu\mathbf{K}\boldsymbol{\alpha}. \tag{19}$$

Kernel PCA (KPCA)

Note

Any solution of $\mathbf{K}\boldsymbol{\alpha} = M\mu\boldsymbol{\alpha}$ is a solution to $\mathbf{K}\mathbf{K}\boldsymbol{\alpha} = M\mu\mathbf{K}\boldsymbol{\alpha}$. Though other solutions can be obtained by adding vectors in the null space of $\mathbf{K}$, it is sufficient to only consider the former since these added vectors do not change the projection results, as we shall see later.

- The eigenvalue problem:

$$\mathbf{K}\boldsymbol{\alpha} = M\mu\boldsymbol{\alpha} \tag{20}$$

- Denote the eigenvectors of $\mathbf{K}$ that correspond to the J largest eigenvalues $\lambda_1 \geq \lambda_2, \ldots, \geq \lambda_J$ by $\mathbf{u}_1, \ldots, \mathbf{u}_J$, then we have $\mu_j = \frac{\lambda_j}{M}$
- To determine the scale of $\boldsymbol{\alpha}$s, we note

$$\|\boldsymbol{\alpha}_j\|^2 = \|\boldsymbol{\Phi}\mathbf{v}_j\|^2 = M\mathbf{v}_j^\top \mathbf{C}\mathbf{v}_j = M\mu_j = \lambda_j \tag{21}$$

so $\boldsymbol{\alpha}_j = \sqrt{\lambda_j}\mathbf{u}_j$.

Kernel PCA (KPCA)

- Training data embeddings:

$$\xi(\mathbf{x}_i)^\top = \phi(\mathbf{x}_i)^\top [\mathbf{v}_1, \ldots, \mathbf{v}_J] = \left[\sqrt{\lambda_1} u_{1i}, \ldots, \sqrt{\lambda_J} u_{Ji}\right]$$

- Out-of-sample extension:

$$\xi(\mathbf{x})^\top = \phi(\mathbf{x})^\top [\mathbf{v}_1, \ldots, \mathbf{v}_J] = [\xi_1(\mathbf{x}), \ldots, \xi_J(\mathbf{x})]$$

$$\xi_j(\mathbf{x}) = \frac{1}{\sqrt{\lambda_j}} \mathbf{u}_j^\top \mathbf{k}_\mathbf{x} = \frac{1}{\sqrt{\lambda_j}} \sum_{m=1}^{M} u_{jm} k(\mathbf{x}, \mathbf{x}_m) \tag{22}$$

Note

$$\mathbf{v}_j = \frac{1}{M\mu_j} \mathbf{\Phi}^\top \boldsymbol{\alpha}_j = \frac{1}{\lambda_j} \mathbf{\Phi}^\top \boldsymbol{\alpha}_j \tag{23}$$

$$\phi(\mathbf{x})^\top \mathbf{v}_j = \frac{1}{\lambda_j} \mathbf{k}_\mathbf{x}^\top \boldsymbol{\alpha}_j, \quad \phi(\mathbf{x}_i)^\top \mathbf{v}_j = \frac{1}{\lambda_j} \mathbf{k}_{\mathbf{x}_i}^\top \boldsymbol{\alpha}_j = [\boldsymbol{\alpha}_j]_i = \sqrt{\lambda_j} [\mathbf{u}_j]_i \tag{24}$$

where $\mathbf{k}_\mathbf{x} = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_M)]^\top$.

Centering the Kernel Matrix

Now consider the case where features have non-zero means

$$\tilde{K}_{ij} = \left[\phi(\mathbf{x}_i) - \frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}_m)\right]^\top \left[\phi(\mathbf{x}_j) - \frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}_m)\right] \tag{25}$$

$$= K_{ij} - \frac{1}{M}\sum_{m=1}^{M}K_{im} - \frac{1}{M}\sum_{m=1}^{M}K_{mj} + \frac{1}{M^2}\sum_{k,m=1}^{M}K_{km} \tag{26}$$

$$= [\mathbf{PKP}]_{ij}. \tag{27}$$

where $\mathbf{P} = \mathbf{I} - \frac{1}{m}\mathbf{e}\mathbf{e}^\top$, $\mathbf{e} = [1, \ldots, 1]^\top$ is called the "centering" matrix.

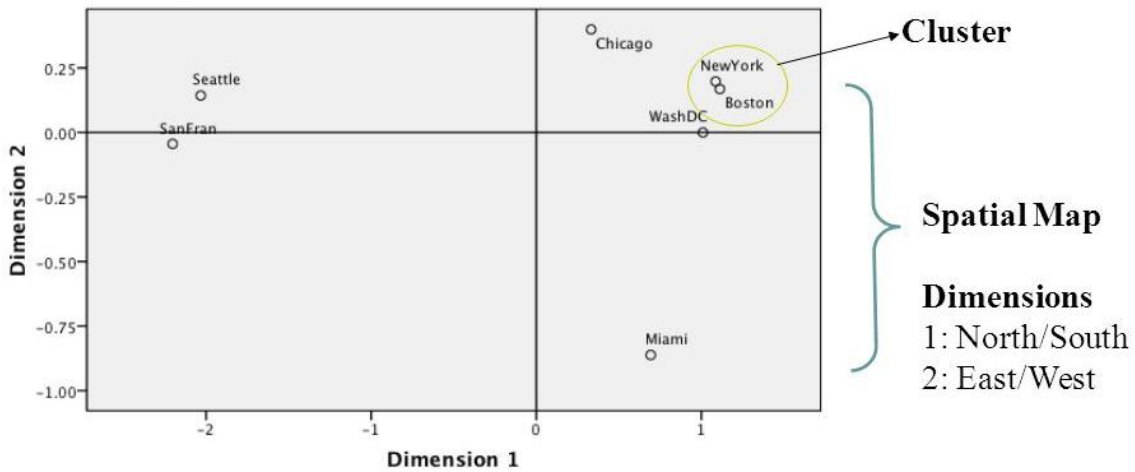
Outline

Multidimensional Scaling (MDS)

- MDS arose in behavior sciences.
- Given a measure of dissimilarity (distances) between each pair of data points, MDS searches a low-dimensional embedding for each point in the Euclidean space, where their dissimilarities become squared distances.

Distances Matrix: Symmetric

		1 BOST	2 NY	3 DC	4 MIAM	5 CHIC	6 SEAT	7 SF	8 LA	9 DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DENVER	1949	1771	1616	2037	996	1307	1235	1059	0



Classical MDS

$$\mathbf{P}[\mathbf{x}_i^\top \mathbf{x}_j]\mathbf{P} = [(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_j - \boldsymbol{\mu})], \tag{28}$$

$$\mathbf{P}[\|\mathbf{x}_i - \mathbf{x}_j\|^2]\mathbf{P} = -2[(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_j - \boldsymbol{\mu})], \tag{29}$$

where $\mathbf{P} = \mathbf{I} - \frac{1}{m}\mathbf{e}\mathbf{e}^\top$.

Theorem (Classical MDS)

Consider the class of symmetric matrices $\mathbf{A} \in \mathbb{R}^{M\times M}$ so that $A_{ii} = 0$, and $A_{ij} \geq 0$. Then

- $\mathbf{A}$ *is a distance matrix (with embedding space $\mathbb{R}^d$ for some d)* $\iff$ $\bar{\mathbf{A}} = -\mathbf{PAP}$ *is positive semidefinite (Gram matrix).*
- *Given that $\mathbf{A}$ is a distance matrix, the minimal embedding dimension d is the rank of $\bar{\mathbf{A}}$, and the embedding vectors are any set of Gram vectors of $\bar{\mathbf{A}}$, scaled by a factor of $\frac{1}{\sqrt{2}}$.*

Outline

Isomap

- Isomap generalizes MDS to nonlinear manifolds
- **Main idea**: Replacing the Euclidean distance by an empirical approximation to the geodesic distance on the manifold.
- Estimate the geodesic distance by finding the shortest path π (a sequence of points, $\pi_0 = a, \pi_l = b$) on the k-nearest-neighbor graph:

$$D(a, b) = \min_{\pi} \sum_{i=0}^{l-1} d(\pi_i, \pi_{i+1}). \tag{30}$$

  $\mathcal{D}(\cdot, \cdot)$ is the estimate of geodesic distance on the manifold of the dataset.
- Then MDS is performed on $A_{ij} = D^2(\mathbf{x}_i, \mathbf{x}_j)$.

Outline

Notations

- Graph $G = (V, E)$, $V = v_1, \ldots, v_n$
- Each edge carries a weight $w_{ij} \geq 0$, $w_{ij} = w_{ji}$. Weight matrix: $\mathbf{W} = (w_{ij})_{i,j=1,\ldots,n}$.
- The degree of a vertex $v_i \in V$:

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{31}$$

  Degree matrix: $\mathbf{D} = diag(d_1, \ldots, d_n)$.
- Given a subset of vertices $A \subset V$, denote its complement $\bar{A}$.
- Indicator vector $\mathbb{I}_A = [f_1, \ldots, f_n]^\top$, where $f_i = 1$ if $v_i \in A$ else $f_i = 0$.
- For $A, B \subset V$:

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{32}$$

$$cut(A, B) = \frac{1}{2}(W(A, B) + W(B, A)) \tag{33}$$

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

Graph Laplacian

The graph Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{34}$$

$\mathbf{L}$ satisfies the following properties:

1. $\forall \mathbf{x} \in \mathbb{R}^n$, we have $\mathbf{x}^\top \mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2$
2. $\mathbf{L}$ is symmetric and positive semidefinite.
3. $\mathbf{L}$ has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$. The smallest eigenvalue of $\mathbf{L}$ is 0, the corresponding eigenvector is the constant one vector $\mathbb{I}$.
4. The number of zero eigenvalues k of $\mathbf{L}$ equals the number of connected components $A_1, \ldots, A_k$ in the graph. The eigenspace of them is spanned by indicator vectors $\mathbb{I}_{A_1}, \ldots, \mathbb{I}_{A_k}$.

Normalized Graph Laplacian

There are two kinds of normalized graph Laplacian matrices, defined as

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W} \tag{35}$$

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{36}$$

1. $\forall \mathbf{x} \in \mathbb{R}^n$, we have $\mathbf{x}^\top \mathbf{L}_{sym}\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(\frac{\mathbf{x}_i}{\sqrt{d_i}} - \frac{\mathbf{x}_j}{\sqrt{d_j}})^2$
2. λ is an eigenvalue of $\mathbf{L}_{rw}$ with eigenvector $\mathbf{u}$
   $\iff$ λ is an eigenvalue of $\mathbf{L}_{sym}$ with eigenvector $\mathbf{D}^{1/2}\mathbf{u}$.
   $\iff$ λ and $\mathbf{u}$ solve the generalized eigenvalue problem $\mathbf{L}\mathbf{u} = \lambda\mathbf{D}\mathbf{u}$.
3. $\mathbf{L}_{rw}, \mathbf{L}_{sym}$ has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$. 0 is an eigenvalue of $\mathbf{L}_{rw}$ with eigenvector $\mathbb{I}$ and an eigenvalue of $\mathbf{L}_{sym}$ with eigenvector $\mathbf{D}^{1/2}\mathbb{I}$.
4. The number of zero eigenvalues k of $\mathbf{L}_{sym}, \mathbf{L}_{rw}$ equals the number of connected components $A_1, \ldots, A_k$ in the graph. The eigenspace of them is spanned by indicator vectors $\mathbb{I}_{A_1}, \ldots, \mathbb{I}_{A_k}$ for $\mathbf{L}_{rw}$ and $\mathbf{D}^{1/2}\mathbb{I}_{A_1}, \ldots, \mathbf{D}^{1/2}\mathbb{I}_{A_k}$ for $\mathbf{L}_{sym}$.

Spectral Clustering

Normalized graph cut for clustering

$$Ncut(A_1, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \bar{A}_i)}{vol(A_i)} = \frac{1}{2}\sum_{i=1}^{k}\frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \tag{37}$$

Consider cluster indicator vectors $\mathbf{h}_j = [h_{j1}, \ldots, h_{jk}]^\top$, where h_{ji} denotes whether the jth node belongs to the ith cluster:

$$h_{ji} = \begin{cases} 1 & \text{if } i \in A_j, \\ 0 & otherwise. \end{cases} \tag{38}$$

we have

$$\mathbf{h}_j^\top \mathbf{L}\mathbf{h}_j = \frac{1}{2}\sum_{i,m=1}^{n} w_{im}(h_{ji} - h_{jm})^2 = \frac{1}{2}\left(\sum_{i\in A_j, m\in\bar{A}_j} w_{im} + \sum_{i\in\bar{A}_j, m\in A_j} w_{im}\right) \tag{39}$$

$$= cut(A_j, \bar{A}_j) \tag{40}$$

Spectral Clustering

Normalized graph cut for clustering

$$Ncut(A_1, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \bar{A}_i)}{vol(A_i)} = \frac{1}{2}\sum_{i=1}^{k}\frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \tag{41}$$

Consider cluster indicator vectors $\mathbf{h}_j = [h_{j1}, \ldots, h_{jk}]^\top$, where h_{ji} denotes whether the jth node belongs to the ith cluster:

$$h_{ji} = \begin{cases} 1/\sqrt{vol(A_j)} & \text{if } i \in A_j, \\ 0 & otherwise. \end{cases} \tag{42}$$

we have

$$\mathbf{h}_j^\top \mathbf{L}\mathbf{h}_j = \frac{1}{2}\sum_{i,m=1}^{n} w_{im}(h_{ji} - h_{jm})^2 = \frac{cut(A_j, \bar{A}_j)}{vol(A_i)}. \tag{43}$$

Let $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_k]$, then

$$Ncut(A_1, \ldots, A_k) = \frac{1}{2}tr(\mathbf{H}^\top \mathbf{L}\mathbf{H}) \tag{44}$$

Spectral Clustering

The problem

$$\min_{A_1,\ldots,A_k} tr(\mathbf{H}^\top \mathbf{L}\mathbf{H}) \quad \text{s.t.} \quad \mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_k], \mathbf{h}_j = \frac{1}{\sqrt{vol(A_j)}} \mathbb{I}_{A_j} \tag{45}$$

It is easy to see $\mathbf{H}^\top \mathbf{D}\mathbf{H} = \mathbf{I}$. We relax the problem to be continuous:

$$\min_{\mathbf{H} \in \mathbb{R}^{n \times k}} tr(\mathbf{H}^\top \mathbf{L}\mathbf{H}) \quad \text{s.t.} \quad \mathbf{H}^\top \mathbf{D}\mathbf{H} = \mathbf{I} \tag{46}$$

Let $\mathbf{T} = \mathbf{D}^{1/2}\mathbf{H}$, we have

$$\min_{\mathbf{T} \in \mathbb{R}^{n \times k}} tr(\mathbf{T}^\top \mathbf{L}_{sym}\mathbf{T}) \quad \text{s.t.} \quad \mathbf{T}^\top \mathbf{T} = \mathbf{I} \tag{47}$$

Remember the eigenvectors of $\mathbf{L}_{sym}$ times $\mathbf{D}^{-1/2}$ is the eigenvectors of $\mathbf{L}_{rw}$, so we can solve

$$\min_{\mathbf{H} \in \mathbb{R}^{n \times k}} tr(\mathbf{H}^\top \mathbf{L}_{rw}\mathbf{H}) \quad \text{s.t.} \quad \mathbf{H}^\top \mathbf{H} = \mathbf{I}, \quad \text{i.e.} \quad \mathbf{L}\mathbf{h} = \lambda \mathbf{D}\mathbf{h}. \tag{48}$$

Spectral Clustering

To recover the discrete partition from the relaxed continuous solution, use k-means on the rows of **H**.

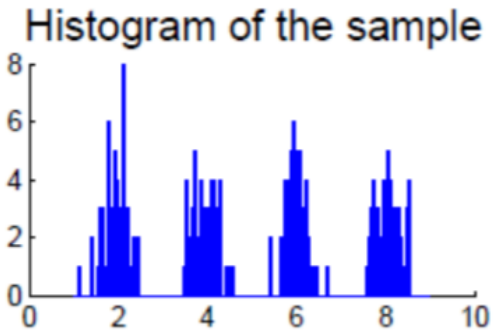
Normalized spectral clustering according to Shi and Malik (2000)

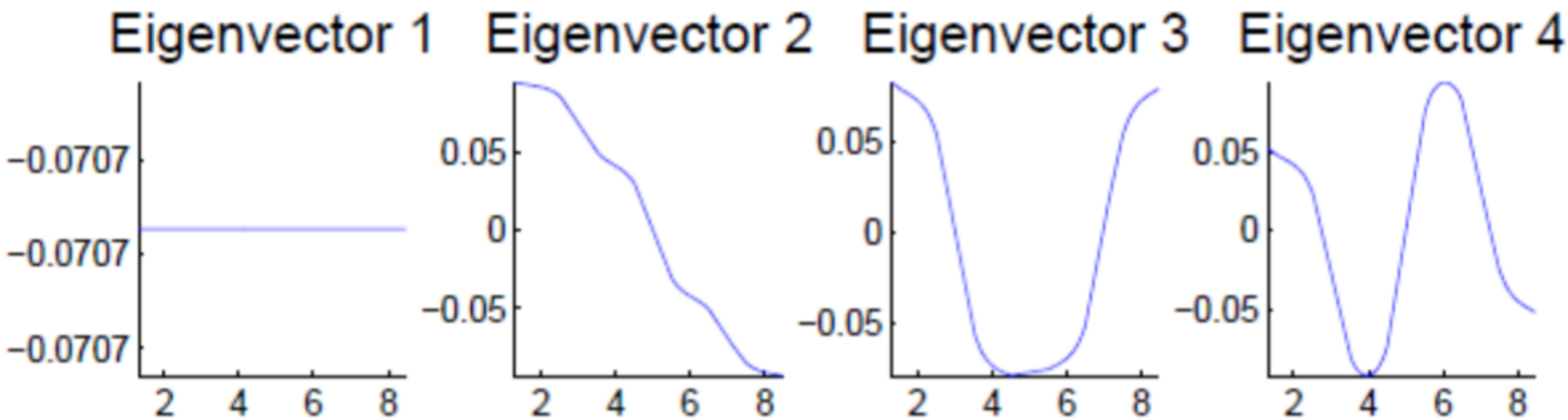
Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k generalized eigenvectors $u_1, \ldots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \, y_j \in C_i\}$.

Spectral Clustering





- 1st Eigenvector is the all ones vector ***1*** (if graph is connected)
- 2nd Eigenvector thresholded at 0 separates first two clusters from last two
- k-means clustering of the 4 eigenvectors identifies all clusters

Outline

Laplacian Eigenmap

- Same algorithm as spectral clustering, but for dimension reduction.
- Algorithm
  1. Constructing the (weighted) adjacency graph (e.g., k-nearest-neighbors, Gaussian kernel).
  2. Solving the generalized eigenvalue problem: $\mathbf{Lh} = \lambda \mathbf{Dh}$.
  3. Let the solutions be $[\mathbf{h}_0, \ldots, \mathbf{h}_n]$, then an m-dimensional embedding is given by $\mathbf{x}_i \to [h_{1i}, \ldots, h_{mi}]^\top$.
- Justification
  - Optimal embedding in terms of local distance preserving:

$$\min_{\mathbf{h}} \sum_{ij} w_{ij}(h_i - h_j)^2 \tag{49}$$

  - Approximately finding eigenfunctions f of Laplace Beltrami Operator $\mathcal{L}(\cdot) = -\mathrm{div}\nabla(\cdot)$ on the data manifold, given the mapping function f.

$$\operatorname*{argmin}_{\|f\|_{\mathcal{L}^2(\mathcal{M})}=1} \int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2 = \operatorname*{argmin}_{\|f\|_{\mathcal{L}^2(\mathcal{M})}=1} \int_{\mathcal{M}} \mathcal{L}(f) f. \tag{50}$$

Outline

Outline

Nyström Method

The Nyström method originates as a method for approximating the solution of Fredholm integral equations of the second kind. Specifically, to find the eigenfunctions $\{\psi_j\}_{j\geq 1}, \psi_j \in L^2(\mathcal{X}, q)$[1] of the covariance kernel $k(\mathbf{x}, \mathbf{y})$ w.r.t. the probability measure q:

$$\int k(\mathbf{x}, \mathbf{y})\psi(\mathbf{y})q(\mathbf{y})d\mathbf{y} = \mu\psi(\mathbf{x}). \tag{51}$$

And there is a constraint that the eigenfunctions $\{\psi_j\}_{j\geq 1}$ are orthonormal under q:

$$\int \psi_i(\mathbf{x})\psi_j(\mathbf{x})q(\mathbf{x})d\mathbf{x} = \delta_{ij}, \tag{52}$$

where $\delta_{ij} = \mathbb{1}(i = j)$.

[1] $L^2(\mathcal{X}, q)$ denotes the space of all square-integrable functions w.r.t. q.

Nyström Method

$$\int k(\mathbf{x}, \mathbf{y})\psi(\mathbf{y})q(\mathbf{y})d\mathbf{y} = \mu\psi(\mathbf{x}). \tag{53}$$

$\{\mathbf{x}^1, \ldots, \mathbf{x}^M\} \sim q$ and applying the equation to these samples, we obtain

$$\frac{1}{M}\mathbf{K}\boldsymbol{\psi} \approx \mu\boldsymbol{\psi}, \tag{54}$$

where $\boldsymbol{\psi} = \left[\psi(\mathbf{x}^1), \ldots, \psi(\mathbf{x}^M)\right]^\top$. We compute the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_J$ with the J largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_J$ for $\mathbf{K}$. Comparing against $\mathbf{K}\mathbf{u}_j = \lambda_j\mathbf{u}_j$:

$$\psi_j(\mathbf{x}^i) \approx \sqrt{M}u_{jm}, \quad m = 1, \ldots, M, \tag{55}$$

$$\mu_j \approx \frac{\lambda_j}{M}. \tag{56}$$

Note $\delta_{ij} = \int \psi_i(\mathbf{x})\psi_j(\mathbf{x})q(\mathbf{x})d\mathbf{x} \approx \frac{1}{M}\sum_{m=1}^{M}\psi_i(\mathbf{x}^m)\psi_j(\mathbf{x}^m)$. For a fixed kernel k, $\frac{\lambda_j}{M}$ will converge to μ_j in the limit as $M \to \infty$.

Nyström Method

$$\int k(\mathbf{x},\mathbf{y})\psi_j(\mathbf{y})q(\mathbf{y})d\mathbf{y} = \mu_j\psi_j(\mathbf{x}). \tag{57}$$

$$\psi_j(\mathbf{x}) = \frac{1}{\mu_j}\int k(\mathbf{x},\mathbf{y})\psi_j(\mathbf{y})q(\mathbf{y})d\mathbf{y} \approx \frac{1}{\mu_j M}\sum_{m=1}^{M}\psi_j(\mathbf{x}^m)k(\mathbf{x},\mathbf{x}^m). \tag{58}$$

Remember

$$\psi_j(\mathbf{x}^i) \approx \sqrt{M}u_{jm}, \quad m = 1,\ldots,M, \tag{59}$$

$$\mu_j \approx \frac{\lambda_j}{M}. \tag{60}$$

The Nyström formula for approximating the value of the jth eigenfunction at any point $\mathbf{x}$:

$$\psi_j(\mathbf{x}) \approx \hat{\psi}_j(\mathbf{x}) = \frac{\sqrt{M}}{\lambda_j}\sum_{m=1}^{M}u_{jm}k(\mathbf{x},\mathbf{x}^m). \tag{61}$$

Nyström Method

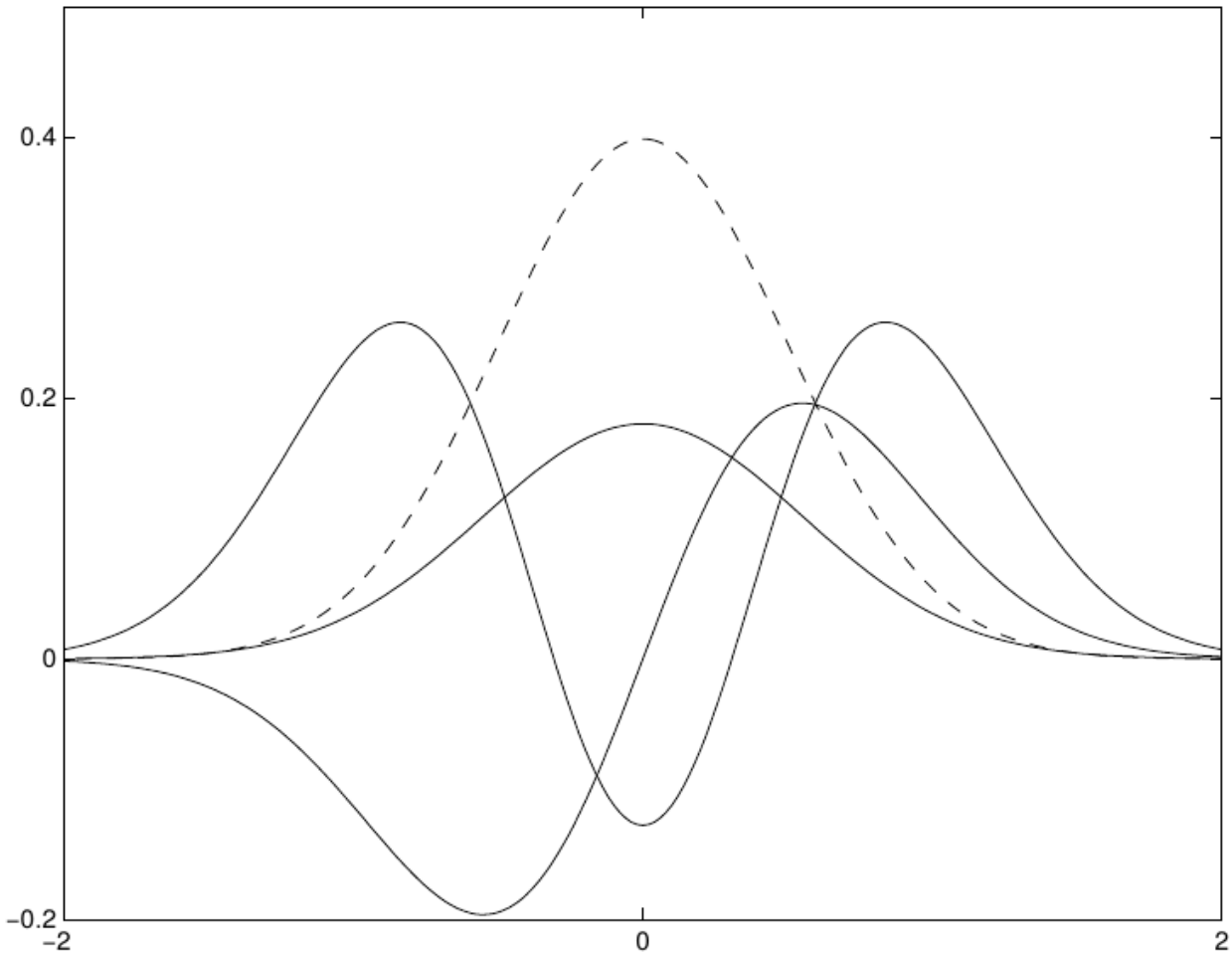


Figure: Eigenfunctions of an Gaussian kernel.

Nyström Method

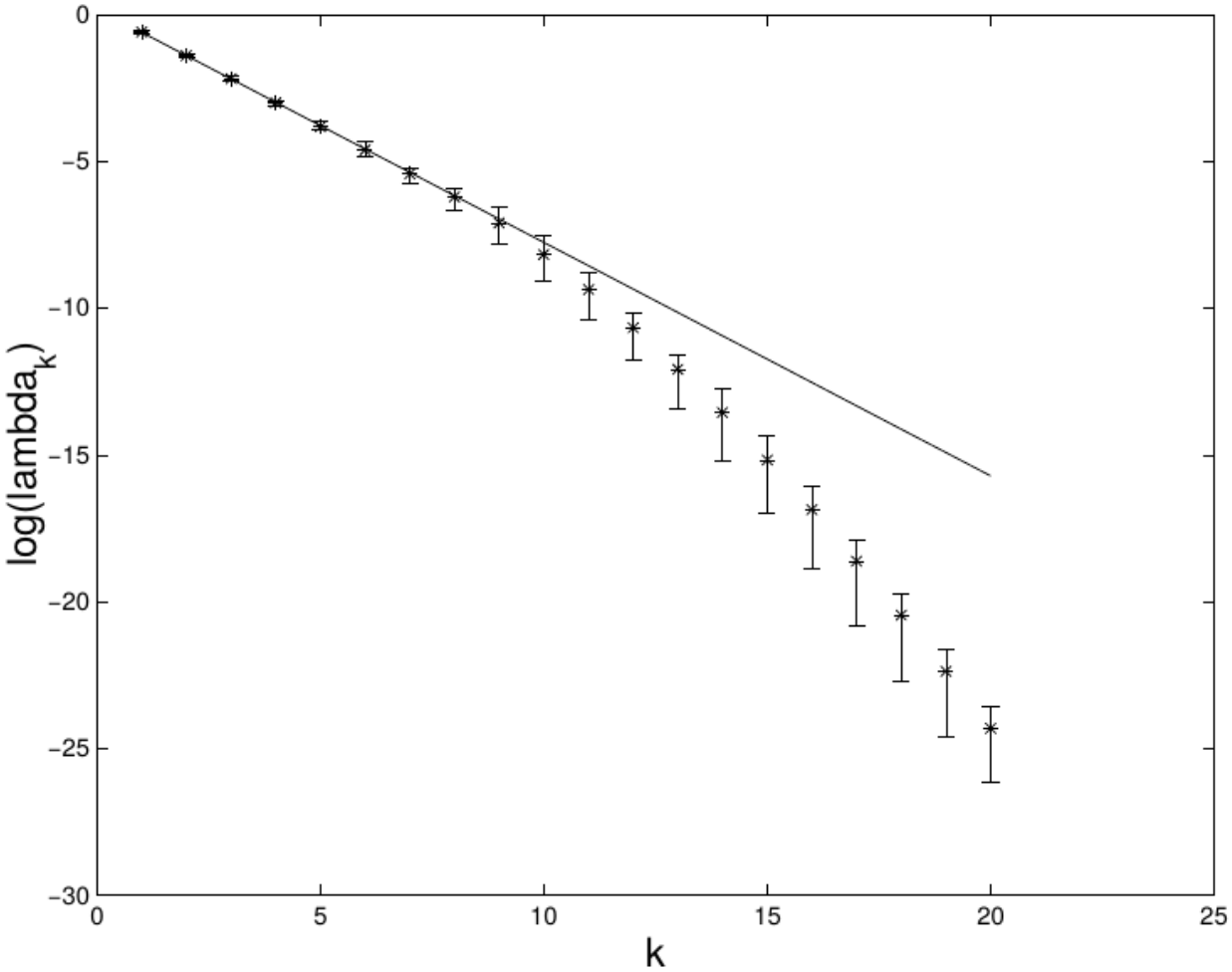


Figure: Eigenvalues of an Gaussian kernel $\mu_j \approx \frac{\lambda_j}{M}$.

Nyström Method Speeds Up Kernel Machines

$$\mathbf{K}_{mm} = \begin{bmatrix} \mathbf{A}_{rr} & \mathbf{B}_{rn} \\ \mathbf{B}_{nr}^{\top} & \mathbf{C}_{nn} \end{bmatrix}$$

where $m = r + n$. Do eigendecomposition for $\mathbf{A}_{rr}$ and then use Nyström formula to extend the eigenvectors to $\mathbf{K}_{mm}$.

Note

When the rank of $\mathbf{K}_{mm}$ is r, the extension is exact, otherwise is approximate.

Outline

MDS, Isomap as KPCA

Classical MDS

- Distance matrix:

$$A_{ij} = dist(\mathbf{x}_i, \mathbf{x}_j) \tag{62}$$

- Gram matrix:

$$\tilde{\mathbf{A}} = -\frac{1}{2}\mathbf{PAP} \tag{63}$$

As KPCA

- Gram matrix:

$$\mathbf{K} = -\frac{1}{2}\mathbf{PAP} \tag{64}$$

- Data-dependent kernel:

$$k_n(a,b) = -\frac{1}{2}(d^2(a,b) - \hat{E}_x[d^2(x,b)] - \hat{E}_{x'}[d^2(a,x')] + \hat{E}_{x,x'}[d^2(x,x')]) \tag{65}$$

MDS, Isomap as KPCA

Data-dependent kernel:

$$k_n(a,b) = -\frac{1}{2}(d^2(a,b) - \hat{E}_x[d^2(x,b)] - \hat{E}_{x'}[d^2(a,x')] + \hat{E}_{x,x'}[d^2(x,x')]) \tag{66}$$

Problem Is $k_n(a,b)$ a valid positive definite (PD) kernel?
True for MDS and almost true for Isomap!

- In the continuum limit for a smooth manifold, the geodesic distance between points on the manifold will be propotional to Euclidean distance in the low-dimensional parameter space of the manifold.
- $k(\mathbf{x},\mathbf{x}') = -\|\mathbf{x}-\mathbf{x}'\|^\beta$ is conditionally positive definite (CPD) for $0 < \beta \leq 2$.
- The above equation transforms a CPD kernel into a PD kernel!

MDS, Isomap as KPCA

Definition 2.20 (Conditionally Positive Definite Matrix) A symmetric $m \times m$ matrix K $(m \geq 2)$ taking values in $\mathbb{K}$ and satisfying

$$\sum_{i,j=1}^{m} c_i \bar{c}_j K_{ij} \geq 0 \text{ for all } c_i \in \mathbb{K}, \text{ with } \sum_{i=1}^{m} c_i = 0$$

is called conditionally positive definite (cpd).

Definition 2.21 (Conditionally Positive Definite Kernel) Let $\mathcal{X}$ be a nonempty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{K}$ which for all $m \geq 2, x_1, \ldots, x_m \in \mathcal{X}$ gives rise to a conditionally positive definite Gram matrix is called a conditionally positive definite (cpd) kernel.

Proposition 2.22 (Constructing PD Kernels from CPD Kernels [42]) Let $x_0 \in \mathcal{X}$, and let k be a symmetric kernel on $\mathcal{X} \times \mathcal{X}$. Then

$$\tilde{k}(x, x') := \frac{1}{2}\big(k(x, x') - k(x, x_0) - k(x_0, x') + k(x_0, x_0)\big)$$

is positive definite if and only if k is conditionally positive definite.

Spectral Clustering, Laplacian Eigenmap as KPCA

Gram matrix:

$$\mathbf{L} = \mathbf{PLP} \tag{67}$$

$$\mathbf{K} = \mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \tag{68}$$

Data-dependent kernel (Bengio, 2003), $\tilde{k}_n$ is the corresponding kernel derived from the graph laplacian (p.d.):

$$k_n(a,b) = \frac{1}{n} \cdot \frac{\tilde{k}_n(a,b)}{\sqrt{\hat{E}_x[\tilde{k}_n(a,x)]\hat{E}_{x'}[\tilde{k}_n(x',b)]}} \tag{69}$$

Spectral Clustering, Laplacian Eigenmap as KPCA

Gram matrix:

$$\mathbf{L} = \mathbf{PLP} \tag{70}$$

$$\mathbf{K} = \mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \tag{71}$$

Data-dependent kernel (Bengio, 2003), $\tilde{k}_n$ is the corresponding kernel derived from the graph laplacian (p.d.):

$$k_n(a,b) = \frac{1}{n} \cdot \frac{\tilde{k}_n(a,b)}{\sqrt{\hat{E}_x[\tilde{k}_n(a,x)]\hat{E}_{x'}[\tilde{k}_n(x',b)]}} \tag{72}$$

Problem (Open) Is $k_n(a,b)$ a valid positive definite kernel?

- (Bengio, 2003) only considers the normalization step from $\mathbf{L}$ to $\mathbf{L}_{sym}$, and leaves open if this step keeps a p.d. kernel.
- (Ham, 2003) only shows the pseudo-inverse of the unnormalized laplacian $\mathbf{L}$ corresponds to a diffusion kernel on the data manifold. However, it skips the normalizing step.

Discussion

- The eigenfunctions of a kernel operator is an orthonormal basis of the space of square-integrable functions.
- Nyström approximation to eigenfunctions and eigenvectors has well-studied error bounds, which is good for theoretical analysis.
- In high-dimensional space, all methods suffer with limited samples.
- In KPCA, commonly used RBF kernels perform poorly in high-dimensions.
- MDS, Isomap, Spectral clustering, Laplacian eigenmap derive a data-dependent kernel by estimating the manifold, which is also restricted by curse-of-dimensionality.
- Is there a good criterion to learn a better kernel from a dataset that fits the data manifold and reduce curse-of-dimensionality?