Lecture 20: Kernel PCA, tSNE, Laplacian eigenmaps

Nisha Chandramoorthy

November 2, 2023

Last time: PCA

- when E and D are linear $\rightarrow$ PCA.

Last time: PCA

- when E and D are linear $\rightarrow$ PCA.
- $E(x) = Wx$, $D(z) = W^\top z$.

Last time: PCA

- when E and D are linear $\to$ PCA.
- $E(x) = Wx$, $D(z) = W^\top z$.
- Let $C = \sum_{i=1}^{m} x_i x_i^\top = X^\top X$ be the data correlation matrix, neglecting the $1/m$ factor.

Last time: PCA

- when E and D are linear $\rightarrow$ PCA.
- $E(x) = Wx$, $D(z) = W^\top z$.
- Let $C = \sum_{i=1}^{m} x_i x_i^\top = X^\top X$ be the data correlation matrix, neglecting the $1/m$ factor.
- C is symmetric and positive semi-definite, $C = V \Lambda V^\top$.
- Theorem PCA: among linear hypothesis classes, $E^* = V^\top$, $D^* = V$, where V is the matrix of eigenvectors of $C = X^\top X$.

Linear algebra review: Rayleigh Quotient

- For a square matrix $A \in \mathbb{R}^{d \times d}$, the Rayleigh quotient is a scalar function,

$$r(x) = \frac{x^\top A x}{x^\top x}.$$

Linear algebra review: Rayleigh Quotient

- For a square matrix $A \in \mathbb{R}^{d \times d}$, the Rayleigh quotient is a scalar function,

$$r(x) = \frac{x^\top A x}{x^\top x}.$$

- Eigenvalues of A are the stationary points of $r(x)$.

Linear algebra review: Rayleigh Quotient

- For a square matrix $A \in \mathbb{R}^{d \times d}$, the Rayleigh quotient is a scalar function,

$$r(x) = \frac{x^\top A x}{x^\top x}.$$

- Eigenvalues of A are the stationary points of $r(x)$.
- $\nabla r(x) = \frac{2}{x^\top x}(Ax - r(x)x)$.

PCA by SVD

- When $m > d$, do eigenvalue decomposition of $X^\top X$ or SVD of X.

PCA by SVD

- When $m > d$, do eigenvalue decomposition of $X^\top X$ or SVD of X.
- When $m < d$, do eigenvalue decomposition of $XX^\top$. If $v_1, v_2, \cdots, v_n$ are the n largest eigenvectors, principal vectors are $\frac{1}{\|X^\top v_i\|} X^\top v_i$.

PCA by SVD

- When $m > d$, do eigenvalue decomposition of $X^\top X$ or SVD of X.
- When $m < d$, do eigenvalue decomposition of $XX^\top$. If $v_1, v_2, \cdots, v_n$ are the n largest eigenvectors, principal vectors are $\frac{1}{\|X^\top v_i\|} X^\top v_i$.
- Computational complexity: $O(\min(m^2 d, md^2))$.

PCA properties

- Exact when $X^\top X$ is rank n.

PCA properties

- Exact when $X^\top X$ is rank n.
- Maximizes variance. Let x be a random vector chosen uniformly from centered data $x_1, \cdots, x_m$. Then, for any $w \in \mathbb{R}^d$,
$$\mathrm{var}(x \cdot w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i)^2.$$

PCA properties

- Exact when $X^\top X$ is rank n.
- Maximizes variance. Let x be a random vector chosen uniformly from centered data $x_1, \cdots, x_m$. Then, for any $w \in \mathbb{R}^d$,
$$\mathrm{var}(x \cdot w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i)^2.$$
- First principal component maximizes $\mathrm{var}(x \cdot w)$ over all w with $\|w\| = 1$. (See Ex 23.4 in book.)

PCA properties

- Exact when $X^\top X$ is rank n.
- Maximizes variance. Let x be a random vector chosen uniformly from centered data $x_1, \cdots, x_m$. Then, for any $w \in \mathbb{R}^d$,
$$\mathrm{var}(x \cdot w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i)^2.$$
- First principal component maximizes $\mathrm{var}(x \cdot w)$ over all w with $\|w\| = 1$. (See Ex 23.4 in book.)
- Informally, PCA rotates the data so that the variance is maximized along the first axis, then the second, and so on.

PCA properties

- Exact when $X^\top X$ is rank n.
- Maximizes variance. Let x be a random vector chosen uniformly from centered data $x_1, \cdots, x_m$. Then, for any $w \in \mathbb{R}^d$,
$$\mathrm{var}(x \cdot w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i)^2.$$
- First principal component maximizes $\mathrm{var}(x \cdot w)$ over all w with $\|w\| = 1$. (See Ex 23.4 in book.)
- Informally, PCA rotates the data so that the variance is maximized along the first axis, then the second, and so on.
- Separates dissimilar points

Kernel PCA

- Let V be the matrix of the top n eigenvectors of $K = XX^\top \in \mathbb{R}^{m \times m}$.

Kernel PCA

- Let V be the matrix of the top n eigenvectors of $K = XX^\top \in \mathbb{R}^{m \times m}$.
- Then, principal vectors are $d_i^* = \frac{1}{\|X^\top v_i\|} X^\top v_i$, $i = 1, 2, \cdots, n$.

Kernel PCA

- Let V be the matrix of the top n eigenvectors of $K = XX^\top \in \mathbb{R}^{m \times m}$.
- Then, principal vectors are $d_i^* = \frac{1}{\|X^\top v_i\|} X^\top v_i$, $i = 1, 2, \cdots, n$.
- For some PD kernel, if $K_{ij} = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = (XX^\top)_{ij}$, can compute K only using kernel evaluations.

Graphical reprensentation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.

Graphical reprensentation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.
- Let D be the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.

Graphical reprensentation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.
- Let D be the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- Graph laplacian: $L = D - W$.

Graphical reprensentation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.
- Let D be the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- Graph laplacian: $L = D - W$.
- Detects local structure / clusters in data.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.
- optimal embeddings: $y_i = E(x_i) = U[i, -n :]$ where U is the matrix of eigenvectors of L.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.
- optimal embeddings: $y_i = E(x_i) = U[i, -n:]$ where U is the matrix of eigenvectors of L.
- For any vector v, $v^\top L v = (1/2) \sum_{i,j=1}^{m} w_{ij}(v_i - v_j)^2$.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.
- optimal embeddings: $y_i = E(x_i) = U[i, -n:]$ where U is the matrix of eigenvectors of L.
- For any vector v, $v^\top L v = (1/2) \sum_{i,j=1}^{m} w_{ij}(v_i - v_j)^2$.
- L is positive semi-definite.

Bottom n eigenvectors

- Rayleigh quotient optimality

Bottom n eigenvectors

- Rayleigh quotient optimality
- Another interpretation: top n eigenvectors of $L^\dagger$. $L^\dagger_{ij}$ represents expected time for random walk $i \to j \to i$.

Bottom n eigenvectors

- Rayleigh quotient optimality
- Another interpretation: top n eigenvectors of $L^\dagger$. $L^\dagger_{ij}$ represents expected time for random walk $i \to j \to i$.
- Kernel PCA with $K = L^\dagger$ is equivalent to Laplacian eigenmaps.

Stochastic neighbor embedding [Hinton and Roweis 2002]

- Stochastic neighbor embedding(SNE): conditional probability that x_i would pick x_j as its neighbor, given by

$$p(x_j|x_i) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}.$$

Stochastic neighbor embedding [Hinton and Roweis 2002]

- Stochastic neighbor embedding(SNE): conditional probability that x_i would pick x_j as its neighbor, given by

$$p(x_j|x_i) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|x_i - x_k\|^2/2\sigma_i^2)}.$$

- For the embeddings $y_i = E(x_i)$,

$$q(y_j|y_i) = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k\neq i}\exp(-\|y_i - y_k\|^2)}.$$

Stochastic neighbor embedding [Hinton and Roweis 2002]

- Stochastic neighbor embedding(SNE): conditional probability that x_i would pick x_j as its neighbor, given by

$$p(x_j|x_i) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|x_i - x_k\|^2/2\sigma_i^2)}.$$

- For the embeddings $y_i = E(x_i)$,

$$q(y_j|y_i) = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k\neq i}\exp(-\|y_i - y_k\|^2)}.$$

- SNE minimizes $\sum_{i=1}^{m} D_{\mathrm{KL}}(p_i\|q_i)$, where p_i and q_i are the conditional probabilities of x_i and y_i respectively.

Stochastic neighbor embedding [Hinton and Roweis 2002]

- Stochastic neighbor embedding(SNE): conditional probability that x_i would pick x_j as its neighbor, given by

$$p(x_j|x_i) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}.$$

- For the embeddings $y_i = E(x_i)$,

$$q(y_j|y_i) = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}.$$

- SNE minimizes $\sum_{i=1}^{m} D_{\mathrm{KL}}(p_i\|q_i)$, where p_i and q_i are the conditional probabilities of x_i and y_i respectively.
- Penalizes large distances between x_i and x_j but also preserves local structure.

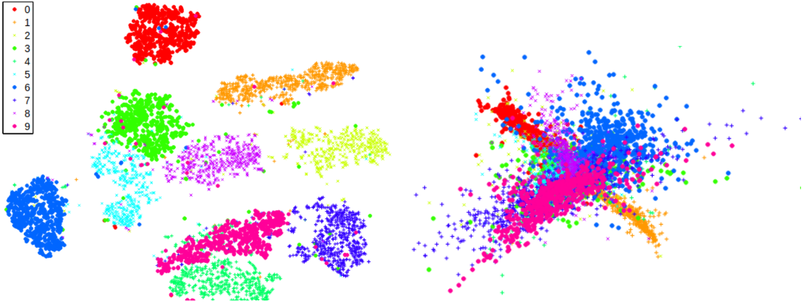
tSNE [Van der Maaten and Hinton 2008]

- tSNE cost function is $D_{\mathrm{KL}}(p\|q) = \sum_{i=1}^{m} \sum_{j\neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$, where $p_i j$ and q_{ij} are the joint probabilities of (x_i, x_j) and (y_i, y_j) respectively.
- Changes joint distribution to a heavy-tailed distribution, $q(y_j, y_i) = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_{k\neq i}(1+\|y_i - y_k\|^2)^{-1}}$.

tSNE [Van der Maaten and Hinton 2008]

- tSNE cost function is $D_{\mathrm{KL}}(p \| q) = \sum_{i=1}^{m} \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$, where $p_i j$ and q_{ij} are the joint probabilities of (x_i, x_j) and (y_i, y_j) respectively.
- Changes joint distribution to a heavy-tailed distribution, $q(y_j, y_i) = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i}(1+\|y_i - y_k\|^2)^{-1}}$.
- approaches inverse square law on embedded space.

tSNE visualization



From Van der Maaten and Hinton 2008. tSNE (left) and LLE (right) on MNIST dataset.

Random projections:

$\underline{x} \rightarrow \boxed{W}\underline{x}$ W: random matrix $\in \mathbb{R}^{d \times n}$

Johnson-Lindenstrauss lemma:

$\exists$ W with w_{ij} being an independent Normal.

s.t. for all $i, j \in [m]$,

$$(1-\varepsilon)\|x_i - x_j\|^2 \leq \|Wx_i - Wx_j\|^2 \leq (1+\varepsilon)\|x_i - x_j\|^2$$

for any $\varepsilon \in (0, \frac{1}{2})$ and $m > 4$

and $n = \frac{20 \log m}{\varepsilon^2}$

Informally:

Use random projection : $x \Rightarrow Wx$

W_{ij} are iid

$$\|Wx_i - Wx_j\| / \|x_i - x_j\| \sim O(1+\varepsilon)$$

$$n = O\left(\frac{\log m}{\varepsilon^2}\right)$$

$x \in \mathbb{R}^d \qquad E(x) \in \mathbb{R}^n$

$n << d$.

$\rightarrow \quad X \in \mathbb{R}^{m \times d}$

$X[i,:] = x_i^T$

$\rightarrow \quad X^T X = V \Sigma V^T$ (SVD = eigenvalue decomposition)

if $\text{rank}(X^T X) = n$,

$$X^T X = \sum_{i=1}^{n} \sigma_i v_i v_i^T$$

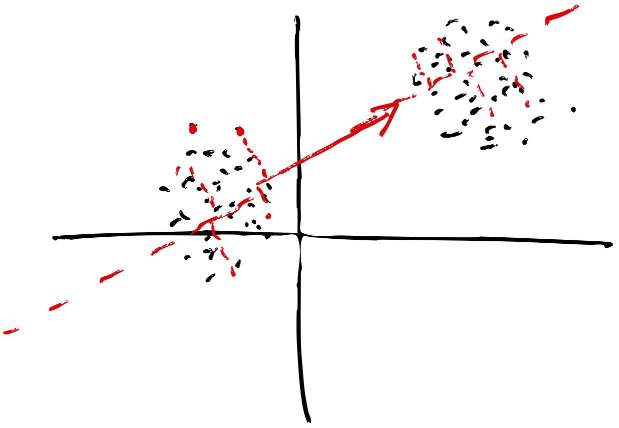
(reduced SVD)

$\rightarrow$ Centered data: $\mathbb{E}x = 0$

$$\rightarrow \quad w^* = \underset{\substack{w \\ \|w\|=1}}{\operatorname{argmax}} \ \operatorname{var}(x \cdot w)$$

$$= \underset{\|w\|=1}{\operatorname{argmax}} \sum_{i=1}^{m} (x_i \cdot w)^2$$

$w^* = v_1$ ($v_1, v_2, .. v_n$ are the top n singular vectors of $X^T X$).



$X^T X$

→ $E(x) = [v_1^T x, v_2^T x, \ldots, v_n^T x] \in \mathbb{R}^n$ (PCA)

→ $\underset{m \times m}{XX^T}[i,j] = \langle \Phi(x_i), \Phi(x_j) \rangle$
$= x_i \cdot x_j$
$= k(x_i, x_j)$

$\text{eig}(XX^T) = \text{eig}(K)$ → Gram matrix

→ $\dfrac{X^T v_i}{\|X^T v_i\|}$ principal vectors

Nonlinear

→ Dim reduction : LLE, Isomaps, } book Mohri
→ Laplacian eigenmaps
⋮
→ tSNE

Graph

Nodes : $x_1, x_2, \ldots, x_m$

Edges : $x_i - x_j$ if they are "neighbors"

Idea : preserve local structure in embedding

i.e, if $\underline{x_i}$ is a neighbor $\underline{x_j}$
then, $\underset{\mathbb{R}^n}{\underset{\cap}{y_i}} = E(x_i)$ neighbor of $\underline{y_i}$

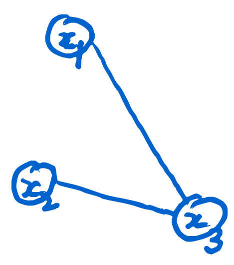
$$\rightarrow L \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \left(\begin{bmatrix} \sum_{j=1}^{m} w_{1j} & & \\ & \ddots & \\ & & \sum_{j=1}^{m} w_{mj} \end{bmatrix} - \begin{bmatrix} -w_{11} & \cdots & \\ & & \\ \cdots & & -w_{mm} \end{bmatrix} \right) \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$= 0$$

eigenvector $\mathbb{1} \in \mathbb{R}^m$ corresponding to 0 eigenvalue

$\rightarrow$ Smallest non-zero eigenvalue of L.

$0 = \lambda_1 \leq \lambda_2 \leq \cdots \lambda_m$



$$W = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

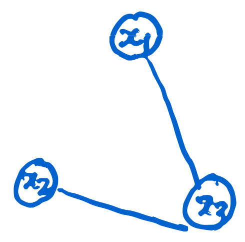
$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$L = D - W$

$$L\mathbb{1} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$\rightarrow \lambda_2$: Fiedler eigenvalue



Corresponding eigenvector represents clusters in data

SNE

$$D_{KL}(\underline{\underline{P}} \,\|\, \underline{\underline{Q}})$$

$$= \sum_{i,j=1}^{m} P(x_i | x_j) \log \frac{P(x_i | x_j)}{Q(y_i | y_j)}$$