Lecture 22: Clustering, LLoyd's algorithm (k-means), spectral clustering

Nisha Chandramoorthy

November 14, 2023

Last time: Johnson-Lindenstrauss lemma

- Let $X \in \mathbb{R}^{m \times d}$ be a matrix of m points in $\mathbb{R}^d$.

Last time: Johnson-Lindenstrauss lemma

- Let $X \in \mathbb{R}^{m \times d}$ be a matrix of m points in $\mathbb{R}^d$.
- Let $0 < \epsilon < 1/2$, $m > 4$. Then, there exists a linear map $A : \mathbb{R}^d \to \mathbb{R}^n$ with $n = O(\epsilon^{-2} \log m)$ such that for all $x_i, x_j \in X$, $i, j \in [m]$, $(1-\epsilon)\|x_i - x_j\|^2 \leqslant \|Ax_i - Ax_j\|^2 \leqslant (1+\epsilon)\|x_i - x_j\|^2$.

Last time: Johnson-Lindenstrauss lemma

- Let $X \in \mathbb{R}^{m \times d}$ be a matrix of m points in $\mathbb{R}^d$.
- Let $0 < \epsilon < 1/2$, $m > 4$. Then, there exists a linear map $A : \mathbb{R}^d \to \mathbb{R}^n$ with $n = O(\epsilon^{-2} \log m)$ such that for all $x_i, x_j \in X$, $i, j \in [m]$, $(1-\epsilon)\|x_i - x_j\|^2 \leqslant \|Ax_i - Ax_j\|^2 \leqslant (1+\epsilon)\|x_i - x_j\|^2$.
- Informal: any set of points in high-dimensional space can be mapped to a lower-dimensional space while approximately preserving the distances between the points.

Proof

- Distortion by Gaussian random matrices: for any $x \in \mathbb{R}^d$, when the entries A_{ij} are iid standard Gaussian,

$$\mathbb{P}(n(1-\epsilon)\|x\|^2 \leqslant \|Ax\|^2 \leqslant n(1+\epsilon)\|x\|^2) \geqslant 1 - 2\exp(-(\epsilon^2 - \epsilon^3)n/4).$$

Proof

- Distortion by Gaussian random matrices: for any $x \in \mathbb{R}^d$, when the entries A_{ij} are iid standard Gaussian,

$$\mathbb{P}(n(1-\epsilon)\|x\|^2 \leqslant \|Ax\|^2 \leqslant n(1+\epsilon)\|x\|^2) \geqslant 1 - 2\exp(-(\epsilon^2 - \epsilon^3)n/4).$$

- Then, deterministic statement of J-L lemma follows from union bound over all m^2 pairs of points.

To show: distortion by Gaussian random matrices

- Let A be a $n \times d$ matrix with iid standard Gaussian entries. Then, $E[(Ax)_j] = 0$ and $\mathrm{Var}((Ax)_j) = \|x\|^2$, for all $j \leqslant n$.

To show: distortion by Gaussian random matrices

- Let A be a $n \times d$ matrix with iid standard Gaussian entries. Then, $E[(Ax)_j] = 0$ and $\mathrm{Var}((Ax)_j) = \|x\|^2$, for all $j \leqslant n$.
- Thus, $1/\|x\|^2 \|Ax\|^2$ is a χ^2 random variable with n degrees of freedom.

To show: distortion by Gaussian random matrices

- Let A be a $n \times d$ matrix with iid standard Gaussian entries. Then, $E[(Ax)_j] = 0$ and $\mathrm{Var}((Ax)_j) = \|x\|^2$, for all $j \leqslant n$.
- Thus, $1/\|x\|^2\|Ax\|^2$ is a χ^2 random variable with n degrees of freedom.
- Chi-squared distribution:
  $\rho(x) = \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2}$, $x \geqslant 0$.

To show: distortion by Gaussian random matrices

- Let A be a $n \times d$ matrix with iid standard Gaussian entries. Then, $E[(Ax)_j] = 0$ and $\mathrm{Var}((Ax)_j) = \|x\|^2$, for all $j \leqslant n$.
- Thus, $1/\|x\|^2 \|Ax\|^2$ is a χ^2 random variable with n degrees of freedom.
- Chi-squared distribution: $\rho(x) = \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2}, x \geqslant 0$.
- Models sum of squares of n independent standard normal random variables.

Implications: random projections

- Random projections surprisingly preserve Euclidean distances between points.

Implications: random projections

- Random projections surprisingly preserve Euclidean distances between points.
- Can be used for dimensionality reduction.

Implications: random projections

- Random projections surprisingly preserve Euclidean distances between points.
- Can be used for dimensionality reduction.
- Can also be used for speeding up nearest neighbor search (e.g. within Laplacian eigenmaps).

Implications: random projections

- Random projections surprisingly preserve Euclidean distances between points.
- Can be used for dimensionality reduction.
- Can also be used for speeding up nearest neighbor search (e.g. within Laplacian eigenmaps).

Compressed sensing revisited

- Let $A \in \mathbb{R}^{n \times d}$ be a random matrix with iid standard Gaussian entries. This is an example of a matrix that satisfies the RIP (restricted isometry property).

Compressed sensing revisited

- Let $A \in \mathbb{R}^{n \times d}$ be a random matrix with iid standard Gaussian entries. This is an example of a matrix that satisfies the RIP (restricted isometry property).
- s-RIP: for all subsets $S \subset [d]$ with $|S| \leqslant s$, there exists an $\epsilon_s > 0$ such that

$$(1 - \epsilon)\|x\|^2 \leqslant \|Ax\|^2 \leqslant (1 + \epsilon)\|x\|^2. \tag{1}$$

Compressed sensing revisited

- Let $A \in \mathbb{R}^{n \times d}$ be a random matrix with iid standard Gaussian entries. This is an example of a matrix that satisfies the RIP (restricted isometry property).
- s-RIP: for all subsets $S \subset [d]$ with $|S| \leqslant s$, there exists an $\epsilon_s > 0$ such that

$$(1 - \epsilon)\|x\|^2 \leqslant \|Ax\|^2 \leqslant (1 + \epsilon)\|x\|^2. \tag{1}$$

- (Candes,Romberg, Tao 2005) If x is s-sparse, then,

$$x = \mathrm{argmin}_{z \in \mathbb{R}^d} \|z\|_1 \quad \text{s.t.} \quad Ax = Az. \tag{2}$$

Convolutional Neural Networks (source: cs231n.stanford.edu)

- Suitable for image recognition. Won the 2012 ImageNet competition and subsequent ones.
- Three types of layers: convolutional, FC, pooling
- Convolutional layer: accepts a volume of size $W_1 \times H_1 \times D_1$ and outputs a volume of size $W_2 \times H_2 \times D_2$ where $W_2 = (W_1 - F + 2P)/S + 1$ and $H_2 = (H_1 - F + 2P)/S + 1$ and $D_2 = K$.
- K is number of filters, F is filter size, S is stride, P is padding.
- Pooling layer: downsamples along width and height, and optionally along depth.
- FC layer: computes class scores, resulting in volume of size $1 \times 1 \times K$.

Clustering: unsupervised learning

- Given a set of points, $\{x_i\}_{i \in [m]}$, $x_i \in \mathbb{R}^d$, partition them into k clusters.

Clustering: unsupervised learning

- Given a set of points, $\{x_i\}_{i\in[m]}$, $x_i \in \mathbb{R}^d$, partition them into k clusters.
- Closely related to dimensionality reduction.

Clustering: unsupervised learning

- Given a set of points, $\{x_i\}_{i\in[m]}$, $x_i \in \mathbb{R}^d$, partition them into k clusters.
- Closely related to dimensionality reduction.
- Definition of clustering depends on the definition of distance between points.

Clustering: unsupervised learning

- Given a set of points, $\{x_i\}_{i \in [m]}$, $x_i \in \mathbb{R}^d$, partition them into k clusters.
- Closely related to dimensionality reduction.
- Definition of clustering depends on the definition of distance between points.
- Center-based clustering: k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.

Lloyd's algorithm

- Randomly choose k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.

Lloyd's algorithm

- Randomly choose k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.
- Given centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, assign each point x_i to the closest center. That is,

$$C_j = \{x_i : j \in \mathrm{argmin}_l \|x_i - \mu_l\|\}.$$

Lloyd's algorithm

- Randomly choose k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.
- Given centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, assign each point x_i to the closest center. That is,

$$C_j = \{x_i : j \in \mathrm{argmin}_l \|x_i - \mu_l\|\}.$$

- Given clusters $C_1, \ldots, C_k$, update centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ as

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i.$$

k-means algorithm (Lloyd's algorithm)

- Lloyd's algorithm is an approximate method to solve the ERM problem:

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu(C_j)\|^2.$$

k-means algorithm (Lloyd's algorithm)

- Lloyd's algorithm is an approximate method to solve the ERM problem:

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu(C_j)\|^2.$$

- here, $\mu(C_j) = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i = \operatorname{argmin}_{\mu \in \mathbb{R}^d} \sum_{x_i \in C_j} \|x_i - \mu\|^2$ is the mean of the points in cluster C_j.

k-means algorithm (Lloyd's algorithm)

- Lloyd's algorithm is an approximate method to solve the ERM problem:

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu(C_j)\|^2.$$

- here, $\mu(C_j) = \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i = \operatorname{argmin}_{\mu \in \mathbb{R}^d} \sum_{x_i \in C_j} \|x_i - \mu\|^2$ is the mean of the points in cluster C_j.
- Lloyd's algorithm is a heuristic. It is not guaranteed to converge to the global optimum or even a local minimum.

Lloyd's algorithm properties

- Lloyd's algorithm decreases the ERM objective at each iteration.

Lloyd's algorithm properties

- Lloyd's algorithm decreases the ERM objective at each iteration.
- Proof: Let $C_1^{(t)}, \ldots, C_k^{(t)}$ be the clusters at iteration t.

Lloyd's algorithm properties

- Lloyd's algorithm decreases the ERM objective at each iteration.
- Proof: Let $C_1^{(t)}, \ldots, C_k^{(t)}$ be the clusters at iteration t.
- $C_j^{(t)} = \{x_i : j \in \mathrm{argmin}_l \|x_i - \mu_l^{(t-1)}\|\}$.

Lloyd's algorithm properties

- Lloyd's algorithm decreases the ERM objective at each iteration.
- Proof: Let $C_1^{(t)}, \ldots, C_k^{(t)}$ be the clusters at iteration t.
- $C_j^{(t)} = \{x_i : j \in \mathrm{argmin}_l \|x_i - \mu_l^{(t-1)}\|\}$.
- Since $\mu_j^{(t)} = \frac{1}{|C_j^{(t)}|} \sum_{x_i \in C_j^{(t)}} x_i = \mathrm{argmin}_{\mu \in \mathbb{R}^d} \sum_{x_i \in C_j^{(t)}} \|x_i - \mu\|^2$,

$$\sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t)}\|^2 \leqslant \sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t-1)}\|^2, \quad \forall j \in [k].$$

Lloyd's algorithm properties

- Proof (contd.): by definition of $C_j^{(t)}$,

$$\sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t)}\|^2 \leqslant \sum_{x_i \in C_j^{(t-1)}} \|x_i - \mu_j^{(t-1)}\|^2, \quad \forall j \in [k].$$

Lloyd's algorithm properties

- Proof (contd.): by definition of $C_j^{(t)}$,

$$\sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t)}\|^2 \leqslant \sum_{x_i \in C_j^{(t-1)}} \|x_i - \mu_j^{(t-1)}\|^2, \quad \forall j \in [k].$$

- Summing over $j \in [k]$,

$$\sum_{j=1}^{k} \sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t)}\|^2 \leqslant \sum_{j=1}^{k} \sum_{x_i \in C_j^{(t-1)}} \|x_i - \mu_j^{(t-1)}\|^2.$$

Lloyd's algorithm properties

- Proof (contd.): by definition of $C_j^{(t)}$,

$$\sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t)}\|^2 \leqslant \sum_{x_i \in C_j^{(t-1)}} \|x_i - \mu_j^{(t-1)}\|^2, \quad \forall j \in [k].$$

- Summing over $j \in [k]$,

$$\sum_{j=1}^{k} \sum_{x_i \in C_j^{(t)}} \|x_i - \mu_j^{(t)}\|^2 \leqslant \sum_{j=1}^{k} \sum_{x_i \in C_j^{(t-1)}} \|x_i - \mu_j^{(t-1)}\|^2.$$

- Thus, the ERM objective decreases at each iteration.

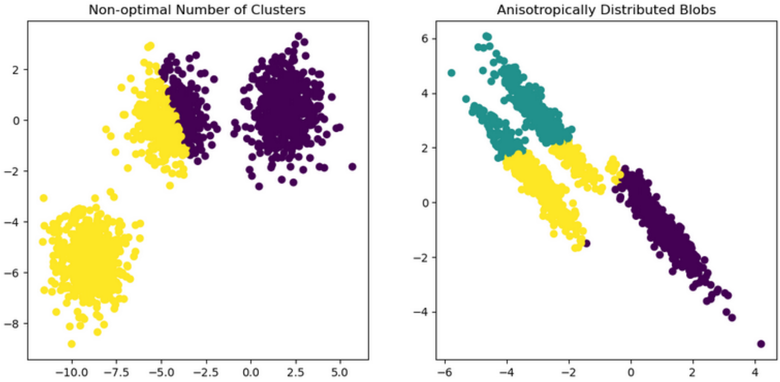
Lloyd's algorithm properties

- k-means algorithm is sensitive to initialization of the centers.

Lloyd's algorithm properties

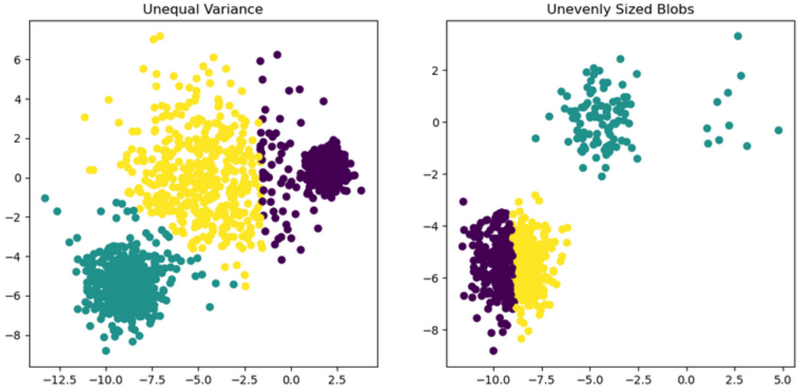
- k-means algorithm is sensitive to initialization of the centers.
- Complexity: $O(mdk)$ per iteration, where m is the number of points, d is the dimension, and k is the number of clusters.

k-means failure modes



Source: sklearn's toy examples

k-means failure modes contd



Source: sklearn's toy examples

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.
- W is symmetric and non-negative.

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.
- W is symmetric and non-negative.
- W is a weighted adjacency matrix of a graph.

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.
- W is symmetric and non-negative.
- W is a weighted adjacency matrix of a graph.
- ERM problem: $\min_{C_1,\dots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}$. Graph min-cut problem.

RatioCut problem: spectral clustering solution

- RatioCut problem: $\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}$.

RatioCut problem: spectral clustering solution

- RatioCut problem: $\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}$.
- Normalization by $|C_j|$ penalizes small clusters.

RatioCut problem: spectral clustering solution

- RatioCut problem: $\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}$.
- Normalization by $|C_j|$ penalizes small clusters.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- $H \in \mathbb{R}^{m \times k}$ is the indicator matrix of the clusters. $H_{ij} = 1/\sqrt{|C_j|}$ if $x_i \in C_j$ and 0 otherwise.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- $H \in \mathbb{R}^{m \times k}$ is the indicator matrix of the clusters. $H_{ij} = 1/\sqrt{|C_j|}$ if $x_i \in C_j$ and 0 otherwise.
- h_i (ith column of H) is nonzero at row j if x_j is in cluster i.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- $H \in \mathbb{R}^{m \times k}$ is the indicator matrix of the clusters. $H_{ij} = 1/\sqrt{|C_j|}$ if $x_i \in C_j$ and 0 otherwise.
- h_i (ith column of H) is nonzero at row j if x_j is in cluster i.
- H has orthonormal columns.

Recall: graphical representation 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.

Recall: graphical representation 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 $D_{ii} = \sum_{j=1}^{m} w_{ij}$.

Recall: graphical representation 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: $L = D - W$.

Recall: graphical representation 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.

Lemma proof: RatioCut objective and graph laplacian connection

- RatioCut objective$(C_1, \cdots, C_k)$

$$:= \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}.$$

Lemma proof: RatioCut objective and graph laplacian connection

- RatioCut objective$(C_1, \cdots, C_k)$

$$:= \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}.$$

- Need to show equal to $\mathrm{Tr}(H^\top LH)$.

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.

Final project deliverables

→ Proposal 10% (2nd Nov)

→ Presentation 25%
= 1 per team 5-6 minutes

3-4 minutes

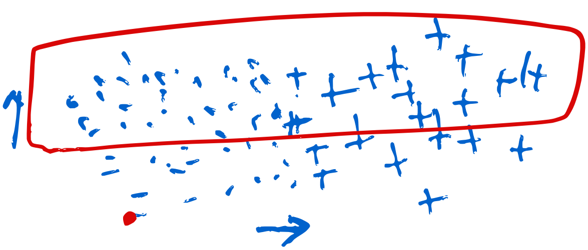
3 sessions
1 overflow session
Zoom 4-8th

7-9 pm

Finalize presentation schedules by 14th Nov

Rules for attendance (% of points for asking questions)

→ Reports (65%) : 7th Dec

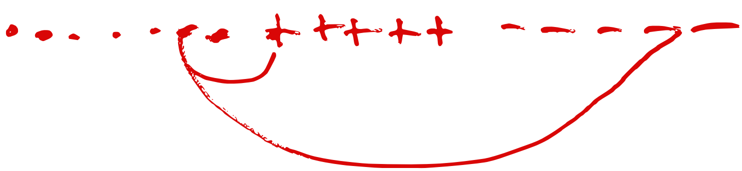


$k = 2$

$\bullet \rightarrow 1$

$+ \rightarrow 2$

Distance b/w two points



$$\rightarrow \quad w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

$$L = D - W$$

$$v^T L v = \frac{1}{2} \sum_{i,j \in [m]} w_{ij} (v_i - v_j)^2$$

$L \in \mathbb{R}^{m \times m}$ $\quad v \in \mathbb{R}^m$

Laplacian eigenmaps objective

Rayleigh quotients

$$\min_{\|v\|=1} v^T L v = \lambda_{min}(L)$$

$$\min_{\substack{V \in \mathbb{R}^{m \times k} \\ \|v_i\| = 1}} V^T L V = \min_{\substack{v_1, v_2, \ldots, v_k \\ v_i \cdot v_j = 0}} \sum_{i=1}^{k} v_i^T L v_i$$

v_i : i^{th} column of V

$$\operatorname{argmin} \sum_{i=1}^{k} v_i^T L v_i = \{\text{bottom } k \text{ eigenvectors of } L\}$$