Advanced Data Analysis: Spectral Clustering

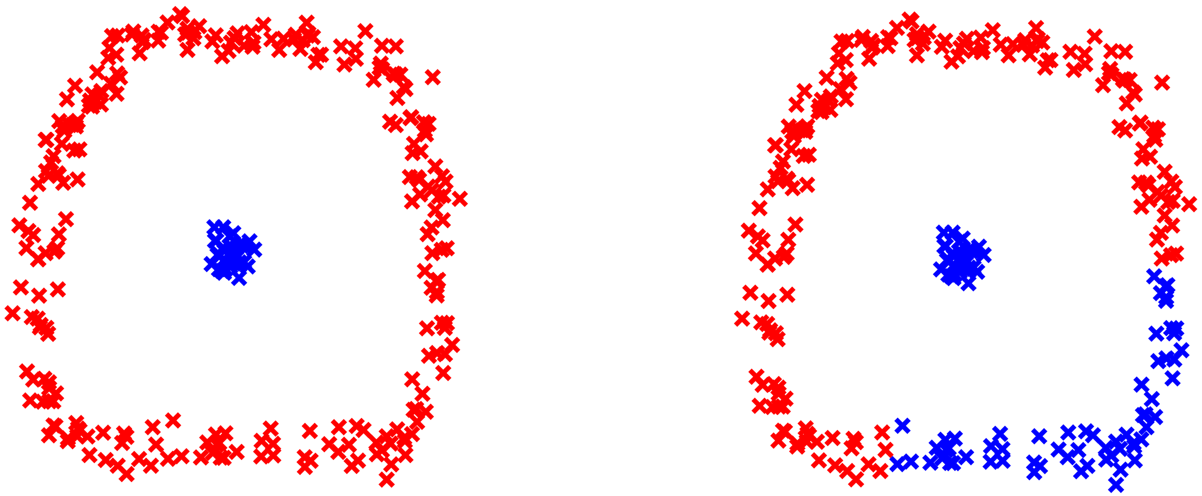
Masashi Sugiyama (Computer Science)

W8E-406, sugi@cs.titech.ac.jp

http://sugiyama-www.cs.titech.ac.jp/~sugi

Kernel K-Means

- Ordinary k-means clustering does not work well if the data crowd has non-convex shapes.
- Kernel k-means is more flexible.
- However, its solution depends crucially on initial cluster assignments since k-means is performed in a high-dimensional feature space.



Similarity-Based Clustering

- Similarity matrix $\boldsymbol{W}$: $W_{i,j}$ is large if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar.
- Assumptions on $\boldsymbol{W}$:
  - Symmetric: $W_{i,j} = W_{j,i}$
  - Non-negative: $W_{i,j} \geq 0$
  - Positive semi-definite: $\forall \boldsymbol{y}, \ \ \langle \boldsymbol{W}\boldsymbol{y}, \boldsymbol{y} \rangle \geq 0$

Examples of Similarity Matrix

$$W_{i,j} = W(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- Distance-based:

$$W(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\gamma^2) \qquad \gamma > 0$$

- Nearest-neighbor-based:

$W(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$ if $\boldsymbol{x}_i$ is a k'-nearest neighbor of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is a k'-nearest neighbor of $\boldsymbol{x}_i$. Otherwise $W(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$.

- Combination of two is also possible.

$$W(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\gamma^2) \\ 0 \end{cases}$$

Local Scaling Heuristic

- γ_i : scaling around the sample $\boldsymbol{x}_i$

$$\gamma_i = \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(k')}\|$$

$\boldsymbol{x}_i^{(k')}$: k'-th nearest neighbor sample of $\boldsymbol{x}_i$

- Local scaling based similarity matrix:

$$W(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / (\gamma_i \gamma_j))$$

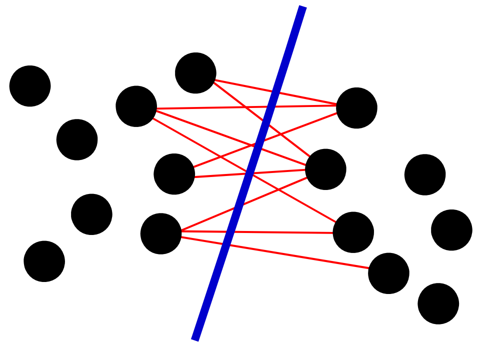
- A heuristic choice is $k' = 7$.

Cut Criterion

- Idea: Minimize sum of similarities between samples inside and outside the cluster

$$\operatorname*{argmin}_{\{\mathcal{C}_i\}_{i=1}^k} \left[\sum_{i=1}^k \sum_{\boldsymbol{x}\in\mathcal{C}_i} \sum_{\boldsymbol{x}'\notin\mathcal{C}_i} W(\boldsymbol{x},\boldsymbol{x}') \right]$$

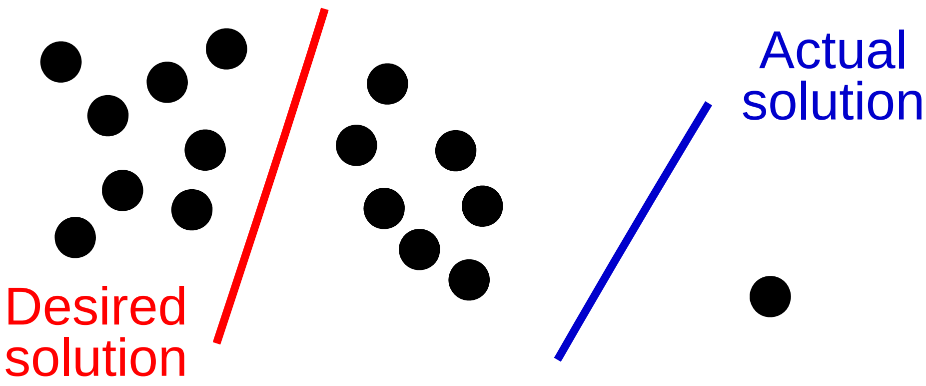
- From a graph-theoretic viewpoint, this corresponds to finding the minimum cut.



Cut Criterion (cont.)

$$\underset{\{\mathcal{C}_i\}_{i=1}^{k}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathcal{C}_i} \sum_{\boldsymbol{x}' \notin \mathcal{C}_i} W(\boldsymbol{x}, \boldsymbol{x}')$$

- Mincut method tends to give a cluster with a very small number of samples.



Normalized Cut Criterion

- Idea: Penalize small clusters

$$\underset{\{\mathcal{C}_i\}_{i=1}^{k}}{\mathrm{argmin}} \left[J_{Ncut} \right]$$

$$J_{Ncut} = \sum_{i=1}^{k} \left[\frac{\sum_{\boldsymbol{x} \in \mathcal{C}_i} \sum_{\boldsymbol{x}' \notin \mathcal{C}_i} W(\boldsymbol{x}, \boldsymbol{x}')}{\sum_{\boldsymbol{x}'' \in \mathcal{C}_i} \sum_{j=1}^{n} W(\boldsymbol{x}'', \boldsymbol{x}_j)} \right]$$

- Denominator is a normalization factor, which is the sum of similarities between samples inside the class and all samples.

Normalized Cut As Weighted Kernel K-Means (Homework)

- Weighted kernel k-means criterion with
  - Weight: $d(\boldsymbol{x}) = \sum_{i=1}^{n} W(\boldsymbol{x}, \boldsymbol{x}_i)$
  - Kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = W(\boldsymbol{x}_i, \boldsymbol{x}_j)/(d(\boldsymbol{x}_i)d(\boldsymbol{x}_j))$

  shares the same optimal solution as the normalized cut criterion:

$$\underset{\{\mathcal{C}_i\}_{i=1}^{k}}{\operatorname{argmin}} [J_{Ncut}] = \underset{\{\mathcal{C}_i\}_{i=1}^{k}}{\operatorname{argmin}} \left[\sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathcal{C}_i} d(\boldsymbol{x}) \|\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i\|^2 \right]$$

$$\boldsymbol{\mu}_i = \frac{1}{s_i} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} d(\boldsymbol{x}')\phi(\boldsymbol{x}') \qquad s_i = \sum_{\boldsymbol{x} \in \mathcal{C}_i} d(\boldsymbol{x})$$

Algorithm 1

- Clustering based on the normalized cut criterion can be obtained by weighted kernel k-means algorithm with

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{W(\boldsymbol{x}_i, \boldsymbol{x}_j)}{d(\boldsymbol{x}_i)d(\boldsymbol{x}_j)} \qquad d(\boldsymbol{x}) = \sum_{i=1}^{n} W(\boldsymbol{x}, \boldsymbol{x}_i)$$

1. Randomly initialize partition: $\{\mathcal{C}_i\}_{i=1}^{k}$
2. Update cluster assignments until convergence:

$$\boldsymbol{x}_j \rightarrow \mathcal{C}_t$$

$$t = \operatorname*{argmin}_i \left[-\frac{2}{s_i} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} d(\boldsymbol{x}')K(\boldsymbol{x}_j, \boldsymbol{x}') + \frac{1}{s_i^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} d(\boldsymbol{x}')d(\boldsymbol{x}'')K(\boldsymbol{x}', \boldsymbol{x}'') \right]$$

Normalized Cut As Weighted Kernel K-Means (cont.)

- Normalized-cut clustering looks reasonable.
- But it is solved by weighted kernel k-means in the end.
- Thus, the drawback of kernel k-means (strong dependency on initial cluster assignments) still remains.

Dual Formulation

$$\underset{\{\mathcal{C}_i\}_{i=1}^k}{\mathrm{argmin}} \left[J_{Ncut} \right]$$

(Homework)

- Instead of optimizing $\{\mathcal{C}_i\}_{i=1}^k$, we optimize cluster indicator $\boldsymbol{A}$:

$$A_{i,j} = \begin{cases} 1 & \text{if } \boldsymbol{x}_j \in \mathcal{C}_i \\ 0 & \text{otherwise} \end{cases}$$

- An optimizer of J_{Ncut} is given by

$$\underset{\boldsymbol{A} \in \mathcal{B}^{k \times n}}{\mathrm{argmin}} \left[\mathrm{tr}(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^\top) \right]$$

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$$

$$\boldsymbol{D} = \mathrm{diag}(\textstyle\sum_{j=1}^n W_{i,j})$$

$$\text{subject to } \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^\top = \boldsymbol{I}_k$$

$\mathcal{B}^{k \times n}$: Set of all $k \times n$ matrices such that one of the elements in each column takes one and others are all zero

Relation to Laplacian Eigenmap

- Let us allow $\boldsymbol{A}$ to take any real values.
- Then the relaxed problem is given as

$$\min_{\boldsymbol{A} \in \mathbb{R}^{k \times n}} \left[\mathrm{tr}(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^\top) \right]$$

$$\text{subject to } \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^\top = \boldsymbol{I}_k$$

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \qquad \boldsymbol{D} = \mathrm{diag}\left(\sum_{j=1}^{n} W_{i,j}\right)$$

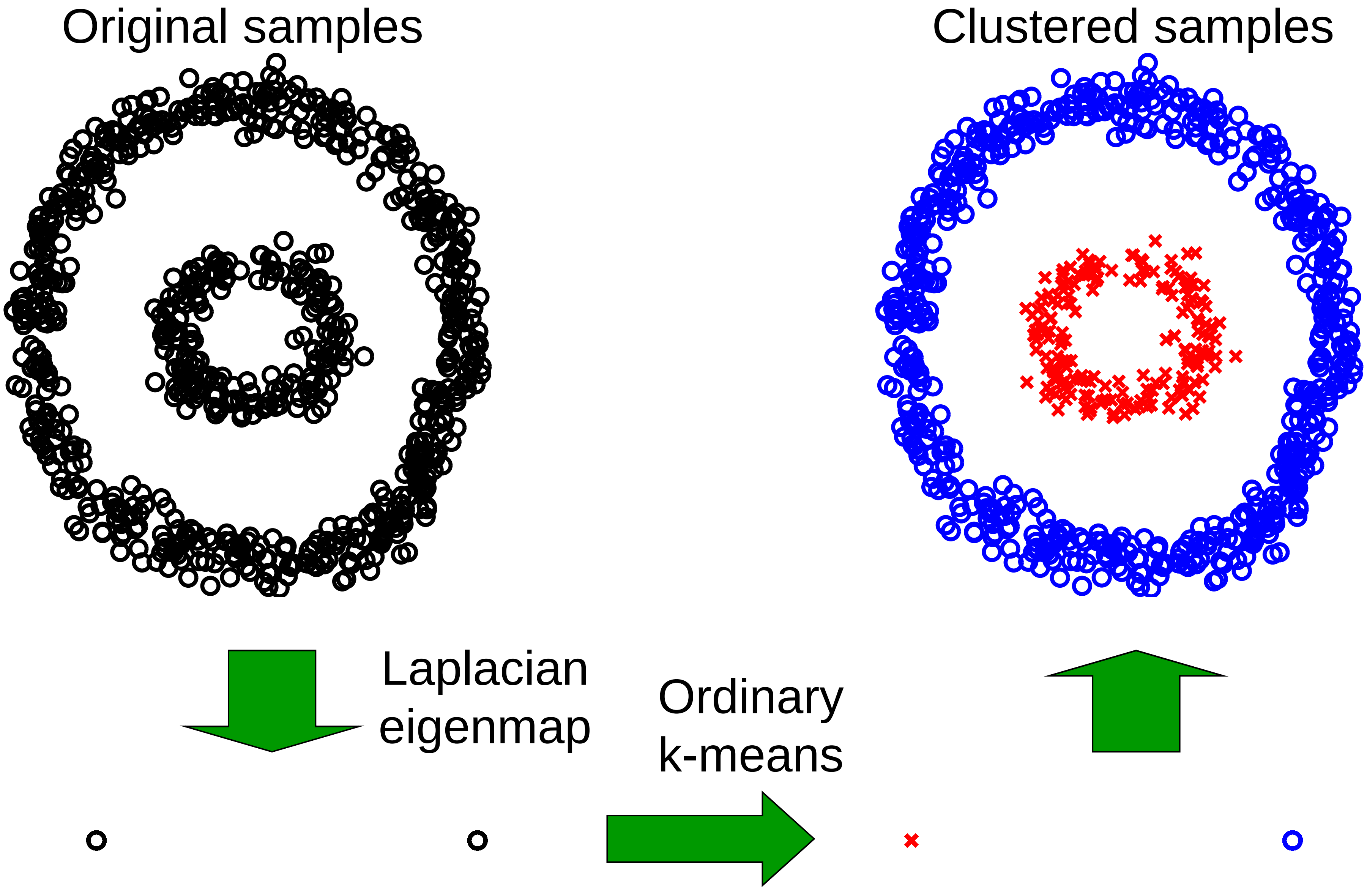
- This is equivalent to Laplacian eigenmap!
- Implication: Laplacian eigenmap embedding "softly" clusters the data samples!

Algorithm 2 (Spectral Clustering)

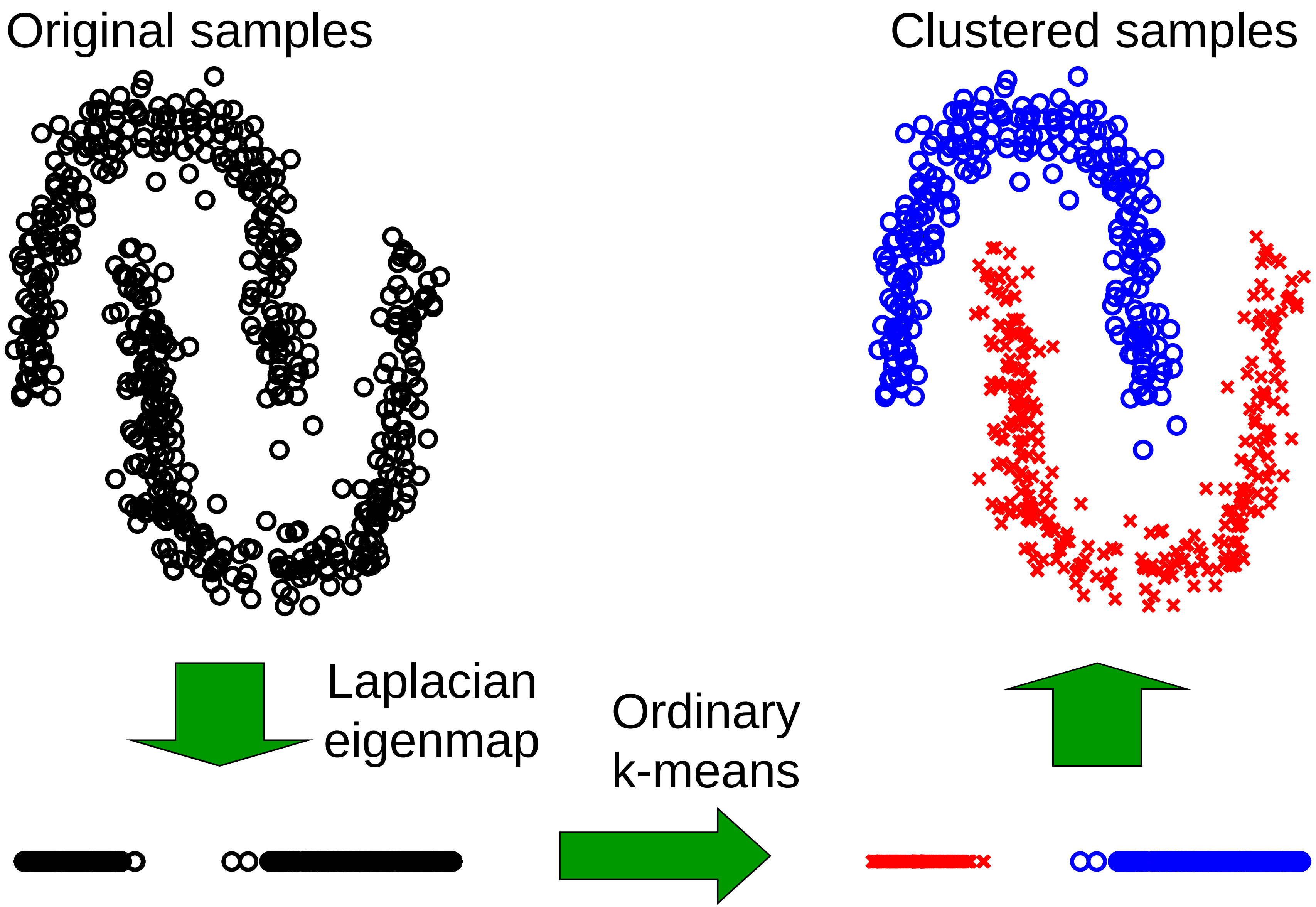
1. Embed $\{\boldsymbol{x}_i\}_{i=1}^{n}$ into $(k-1)$- dimensional space by Laplacian eigenmap embedding.
2. Cluster the embedded samples by (non-kernelized) k-means clustering algorithm.

- Kernel k-means had a drawback that the clustering results crucially depend on initial cluster assignments.
- Since Laplacian eigenmap has a soft clustering property, the above algorithm is less dependent on initialization.

Examples


Original samples
Clustered samples
Laplacian
eigenmap
Ordinary
k-means

Examples (cont.)



Summary of Clustering Methods

- Three different clustering families result in the same criterion!!

- Weighted kernel k-means

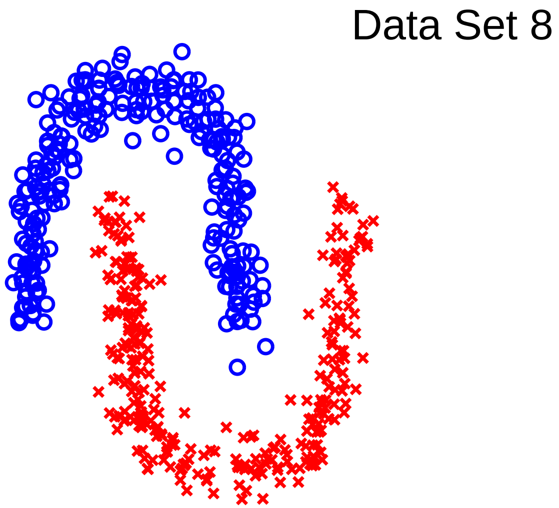
- Normalized cut

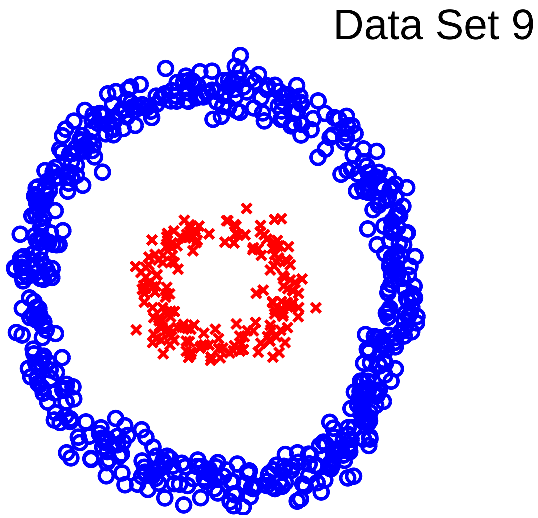
- “Hard” Laplacian eigenmap

Homework

1. Implement Algorithm 2 (spectral clustering) and reproduce the 2-dimensional examples shown in the class.

http://sugiyama-www.cs.titech.ac.jp/~sugi/data/DataAnalysis





Test the algorithm with your own (artificial or real) data and analyze their characteristics.

Homework (cont.)

2. Prove that weighted kernel k-means criterion with
   - Weight: $d(\boldsymbol{x}) = \sum_{i=1}^{n} W(\boldsymbol{x}, \boldsymbol{x}_i)$
   - Kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = W(\boldsymbol{x}_i, \boldsymbol{x}_j)/(d(\boldsymbol{x}_i)d(\boldsymbol{x}_j))$

   shares the same optimal solution as the normalized cut criterion:

$$\underset{\{\mathcal{C}_i\}_{i=1}^{k}}{\operatorname{argmin}} \left[J_{Ncut}\right] = \underset{\{\mathcal{C}_i\}_{i=1}^{k}}{\operatorname{argmin}} \left[J_{WS}\right]$$

Homework (cont.)

2. Hint:

Express all elements in J_{WS} in terms of the affinity $W(\boldsymbol{x}, \boldsymbol{x}')$, e.g.,

$$s_i = \sum_{\boldsymbol{x}'' \in \mathcal{C}_i} \sum_{j=1}^{n} W(\boldsymbol{x}'', \boldsymbol{x}_j)$$

$$J_{WS} = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathcal{C}_i} d(\boldsymbol{x}) \|\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i\|^2$$

$$\boldsymbol{\mu}_i = \frac{1}{s_i} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} d(\boldsymbol{x}')\phi(\boldsymbol{x}')$$

$$s_i = \sum_{\boldsymbol{x} \in \mathcal{C}_i} d(\boldsymbol{x})$$

$$J_{Ncut} = \sum_{i=1}^{k} \left[\frac{\sum_{\boldsymbol{x} \in \mathcal{C}_i} \sum_{\boldsymbol{x}' \notin \mathcal{C}_i} W(\boldsymbol{x}, \boldsymbol{x}')}{\sum_{\boldsymbol{x}'' \in \mathcal{C}_i} \sum_{j=1}^{n} W(\boldsymbol{x}'', \boldsymbol{x}_j)} \right]$$

Homework (cont.)

3. Prove that an optimizer of J_{Ncut} is given by

$$\underset{\boldsymbol{A}\in\mathcal{B}^{k\times n}}{\operatorname{argmin}}\left[\operatorname{tr}(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^\top)\right]$$

$$\text{subject to } \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^\top = \boldsymbol{I}_k$$

$\mathcal{B}^{k\times n}$: Set of all $k \times n$ matrices such that one of the elements in each column takes one and others are all zero

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$$

$$\boldsymbol{D} = \operatorname{diag}\left(\textstyle\sum_{j=1}^{n} \boldsymbol{W}_{i,j}\right)$$

$$\boldsymbol{A}_{i,j} = \begin{cases} 1 & \text{if } \boldsymbol{x}_j \in \mathcal{C}_i \\ 0 & \text{o.w.} \end{cases}$$

Homework (cont.)

3. Hint:

Let $\boldsymbol{A} = (\boldsymbol{a}_1 | \boldsymbol{a}_2 | \cdots | \boldsymbol{a}_k)^\top$ and express all elements in J_{Ncut} in terms of $\{\boldsymbol{a}_i\}_{i=1}^k$, e.g.,

$$\sum_{\boldsymbol{x}'' \in \mathcal{C}_i} \sum_{j=1}^n W(\boldsymbol{x}'', \boldsymbol{x}_j) = \langle \boldsymbol{W}\boldsymbol{a}_i, \boldsymbol{1}_n \rangle = \langle \boldsymbol{D}\boldsymbol{a}_i, \boldsymbol{a}_i \rangle$$

$$J_{Ncut} = \sum_{i=1}^k \left[\frac{\sum_{\boldsymbol{x} \in \mathcal{C}_i} \sum_{\boldsymbol{x}' \notin \mathcal{C}_i} W(\boldsymbol{x}, \boldsymbol{x}')}{\sum_{\boldsymbol{x}'' \in \mathcal{C}_i} \sum_{j=1}^n W(\boldsymbol{x}'', \boldsymbol{x}_j)} \right]$$

Schedule

- June 11th: Projection Pursuit (1)
- June 18th: Preparation for Mini-Conference
- June 25th: Projection Pursuit (2)
  - Application Deadline to Mini-Conference
- July 2nd: Independent Component Analysis
- July 9th: Preparation for Mini-Conference
- July 16th: Mini-Conference Day 1
- July 23rd: Mini-Conference Day 2