Advanced Data Analysis: Spectral Clustering

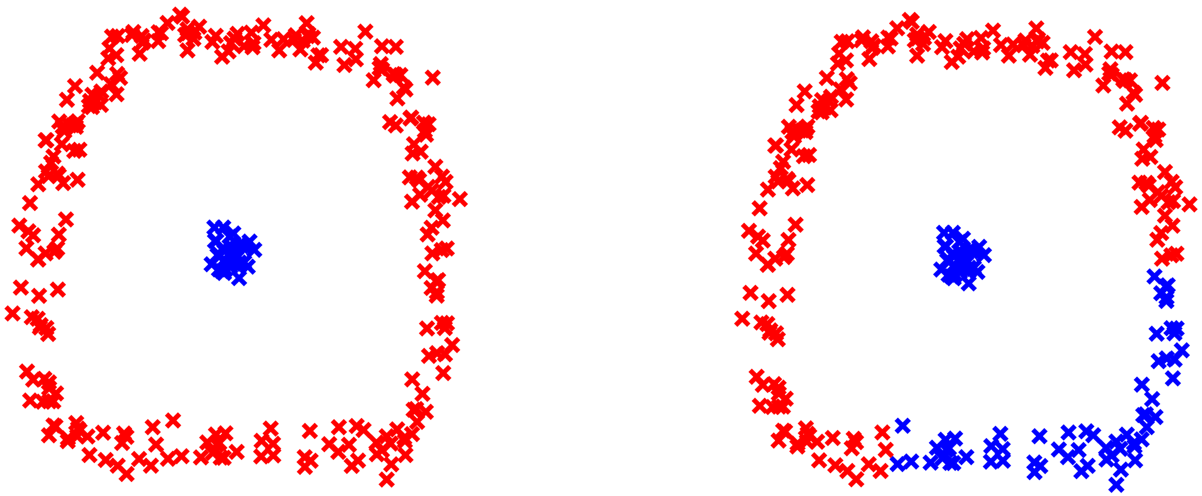
Masashi Sugiyama (Computer Science)

W8E-505, 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 crowds have non-convex shapes.
- Kernel k-means is more flexible.
- However, solution depends crucially on the initial cluster assignments since clustering is carried out in a high-dimensional feature space.



Similarity-Based Clustering

- Similarity matrix $\boldsymbol{W}$: $\boldsymbol{W}_{i,j}$ is large if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar.
- Assumptions on $\boldsymbol{W}$:
  - Symmetric: $\boldsymbol{W}_{i,j} = \boldsymbol{W}_{j,i}$
  - Positive entries: $\boldsymbol{W}_{i,j} \geq 0$
  - Invertible: $^{\exists}\boldsymbol{W}^{-1}$

Examples of Similarity Matrix

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

- 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:

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

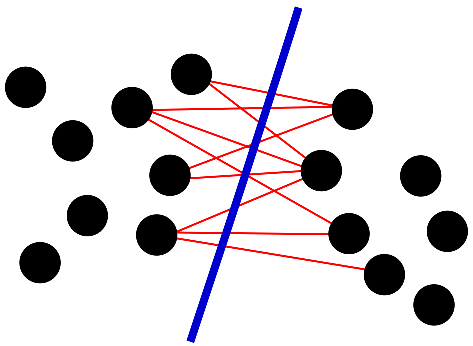
- A heuristic choice is $k = 7$.

Cut Criterion

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

$$\min_{\mathcal{C}_1,\mathcal{C}_2}\left[\sum_{\boldsymbol{x}\in\mathcal{C}_1}\sum_{\boldsymbol{x}'\in\mathcal{C}_2} W(\boldsymbol{x},\boldsymbol{x}') + \sum_{\boldsymbol{x}\in\mathcal{C}_2}\sum_{\boldsymbol{x}'\in\mathcal{C}_1} W(\boldsymbol{x},\boldsymbol{x}')\right]$$

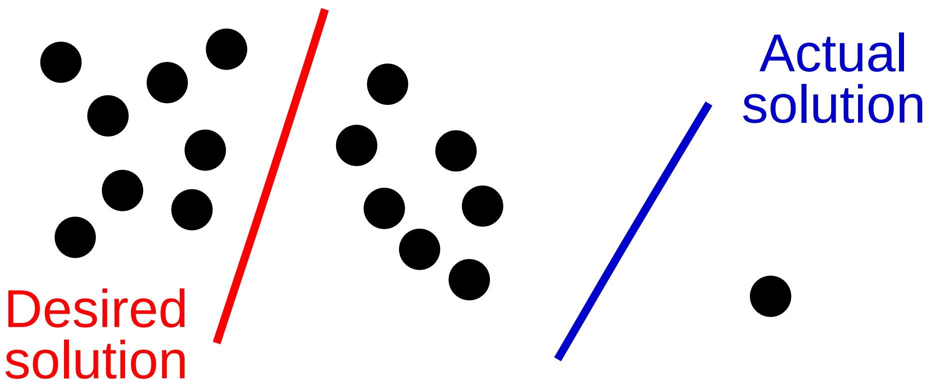
- From a graph-theoretic viewpoint, this corresponds to finding minimum cut.



Cut Criterion (cont.)

$$\min_{\mathcal{C}_1,\mathcal{C}_2}\left[\sum_{\boldsymbol{x}\in\mathcal{C}_1}\sum_{\boldsymbol{x}'\in\mathcal{C}_2}W(\boldsymbol{x},\boldsymbol{x}')+\sum_{\boldsymbol{x}\in\mathcal{C}_2}\sum_{\boldsymbol{x}'\in\mathcal{C}_1}W(\boldsymbol{x},\boldsymbol{x}')\right]$$

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



Normalized Cut Criterion

- Idea: Penalize small clusters
- In two-cluster cases:

$$\min_{\mathcal{C}_1,\mathcal{C}_2}\left[\frac{\dfrac{\sum_{\boldsymbol{x}\in\mathcal{C}_1}\sum_{\boldsymbol{x}'\in\mathcal{C}_2}W(\boldsymbol{x},\boldsymbol{x}')}{n}}{\sum_{\boldsymbol{x}''\in\mathcal{C}_1}\sum_{j=1}^{n}W(\boldsymbol{x}'',\boldsymbol{x}_j)}+\frac{\dfrac{\sum_{\boldsymbol{x}\in\mathcal{C}_2}\sum_{\boldsymbol{x}'\in\mathcal{C}_1}W(\boldsymbol{x},\boldsymbol{x}')}{n}}{\sum_{\boldsymbol{x}''\in\mathcal{C}_2}\sum_{j=1}^{n}W(\boldsymbol{x}'',\boldsymbol{x}_j)}\right]$$

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

Normalized Cut Criterion (cont.)

- In k -cluster cases, normalized cut is defined as

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

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

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:

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

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

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

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

Algorithm 1

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

$$d(\boldsymbol{x}) = \sum_{i=1}^{n} W(\boldsymbol{x}, \boldsymbol{x}_i) \qquad K(\boldsymbol{x}_i, \boldsymbol{x}_j) = [\boldsymbol{D}^{-1}\boldsymbol{W}\boldsymbol{D}^{-1}]_{i,j}$$

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 good.
- But it is solved by (weighted) kernel k-means in the end.
- Thus the drawback (strong dependency on initial cluster assignment) of kernel k-means still remains.

Dual Formulation

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

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

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

- An optimizer of J_{Ncut} is given by

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

(Homework)

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

$\mathcal{B}^{k \times n}$: 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 relaxed problem is given as

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

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

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

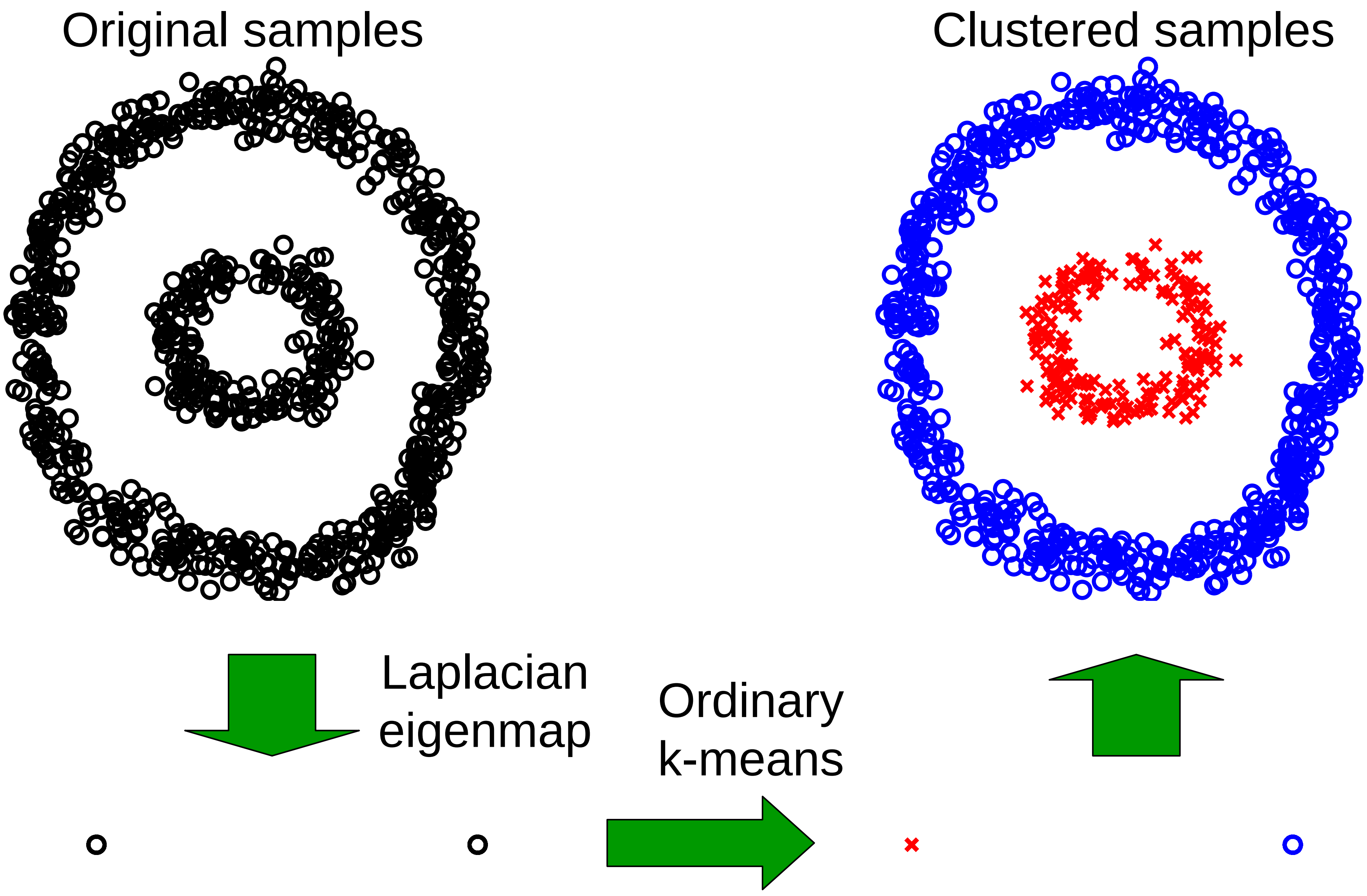
- This is equivalent to Laplacian eigenmap!
- Implication: Laplacian eigenmap embedding "softly" cluster the data samples!

Algorithm 2 (Spectral Clustering)

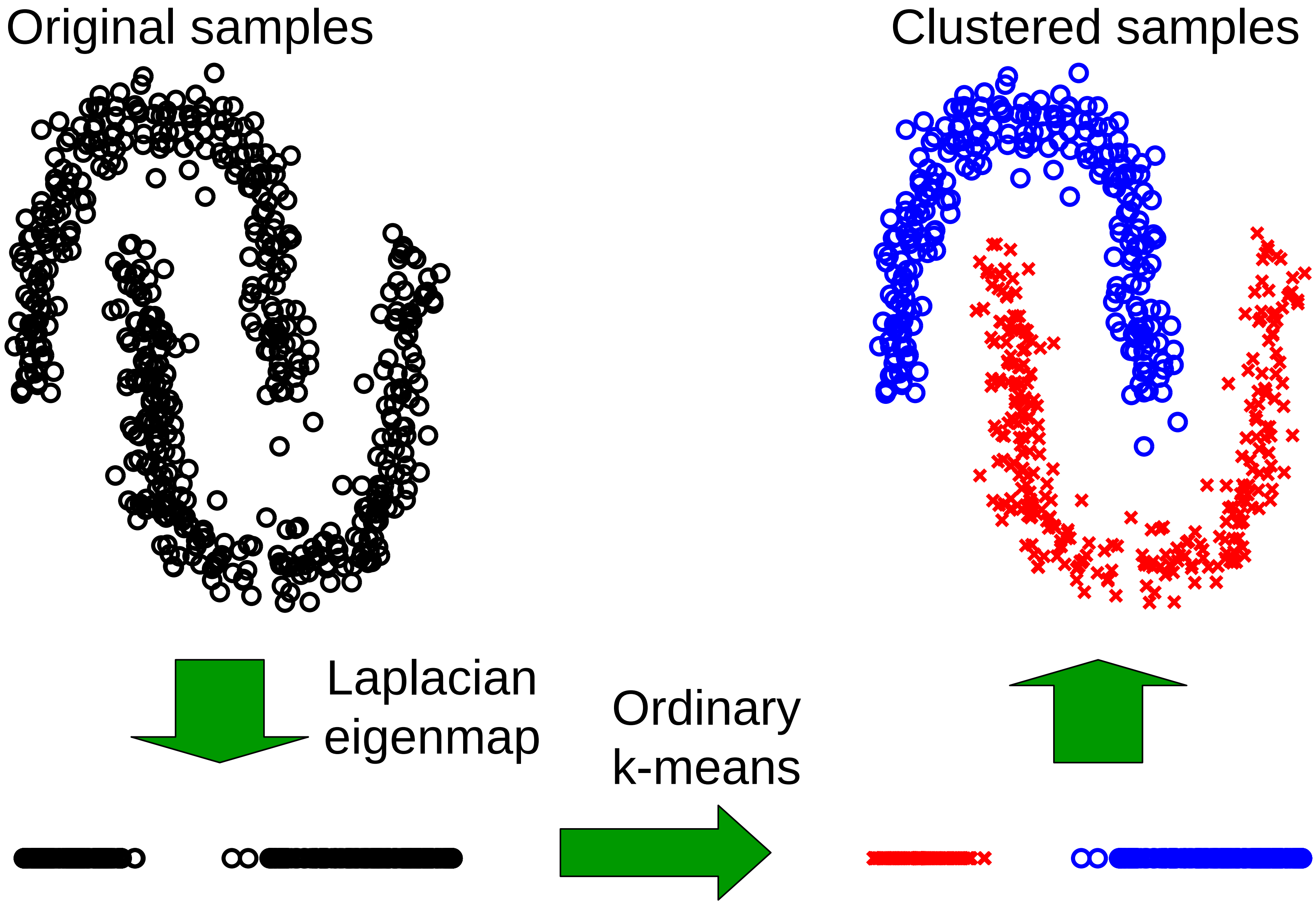
1. Embed $\{\boldsymbol{x}_i\}_{i=1}^{n}$ into $k-1$ dimensional space by Laplacian eigenmap embedding.
2. Cluster the embedded samples by (non-kernelized) k-means clustering algorithm.

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

Examples


Original samples
Clustered samples
Laplacian eigenmap
Ordinary k-means

Examples (cont.)



Notification of Final Assignment

- Data mining: Apply dimensionality reduction or clustering techniques to your own data set and find something interesting.

Mini-Conference on Data Analysis

- On July 10th (final class), we have a mini-conference on data analysis, instead of a regular lecture.
- Some of the students (5-10?) may present their data analysis results.
- Those who give a talk at the conference will have very good grades!

Mini-Conference on Data Analysis

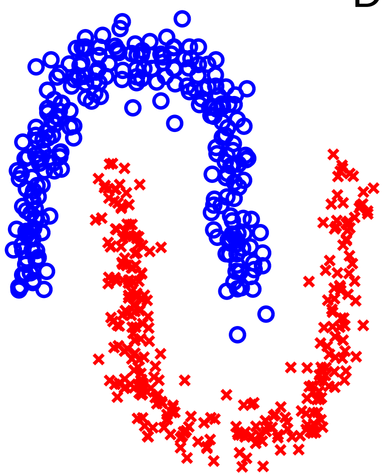
- Application: On July 3rd, just say to me "I want to give a talk!".
- Presentation: approx. 10 min?
  - Description of your data
  - Methods to be used
  - Outcome
- Slides should be in English.
- Better to speak in English, but Japanese may also be allowed (perhaps your friends will provide simultaneous translation!).

Homework

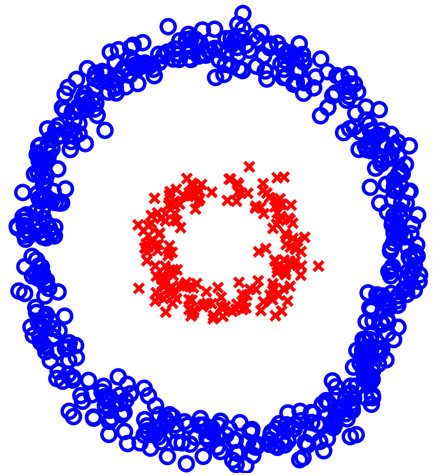
1. Implement Algorithm 2 (spectral clustering) and reproduce the 2-dimensional examples shown in the class.

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

Data Set 8



Data Set 9



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

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

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

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

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

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

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

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.,

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

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