Advanced Data Analysis: K-Means Clustering

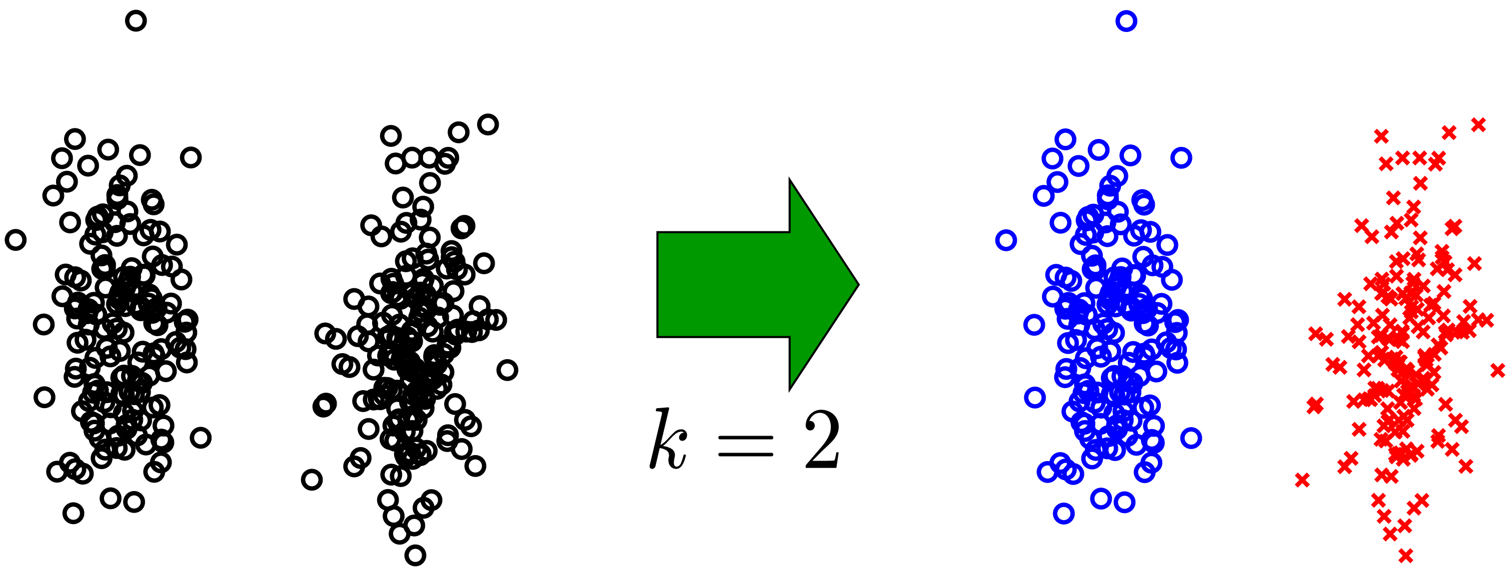
Masashi Sugiyama (Computer Science)

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

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

Data Clustering

- We want to divide data samples $\{\boldsymbol{x}_i\}_{i=1}^n$ into k $(1 \leq k \leq n)$ disjoint clusters so that samples in the same cluster are similar.
- We assume that k is prefixed.



Within-Cluster Scatter Criterion

- Idea: Cluster the samples so that within-cluster scatter is minimized
- $\mathcal{C}_i$: Set of samples in cluster i

$$\bigcup_{i=1}^{k} \mathcal{C}_i = \{\boldsymbol{x}_j\}_{j=1}^{n}$$

$$\mathcal{C}_i \cap \mathcal{C}_j = \phi$$

- Criterion:

$$\min_{\{\mathcal{C}_i\}_{i=1}^{k}} \left[\sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathcal{C}_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2 \right]$$

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} \boldsymbol{x}'$$

Within-Cluster Scatter Minimization

$$\min_{\{\mathcal{C}_i\}_{i=1}^{k}} \left[\sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathcal{C}_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2 \right]$$

- When all possible cluster assignment is tested in a greedy manner, computation time is proportional to k^n .
- Actually, the above optimization problem is NP-hard, i.e., we do not yet have a polynomial-time algorithm.

K-Means Clustering Algorithm

- Randomly initialize cluster centroids: $\{\boldsymbol{\mu}_i\}_{i=1}^k$
- Repeat the following steps until convergence:
  - Update cluster assignments: $j = 1, 2, \ldots, n$

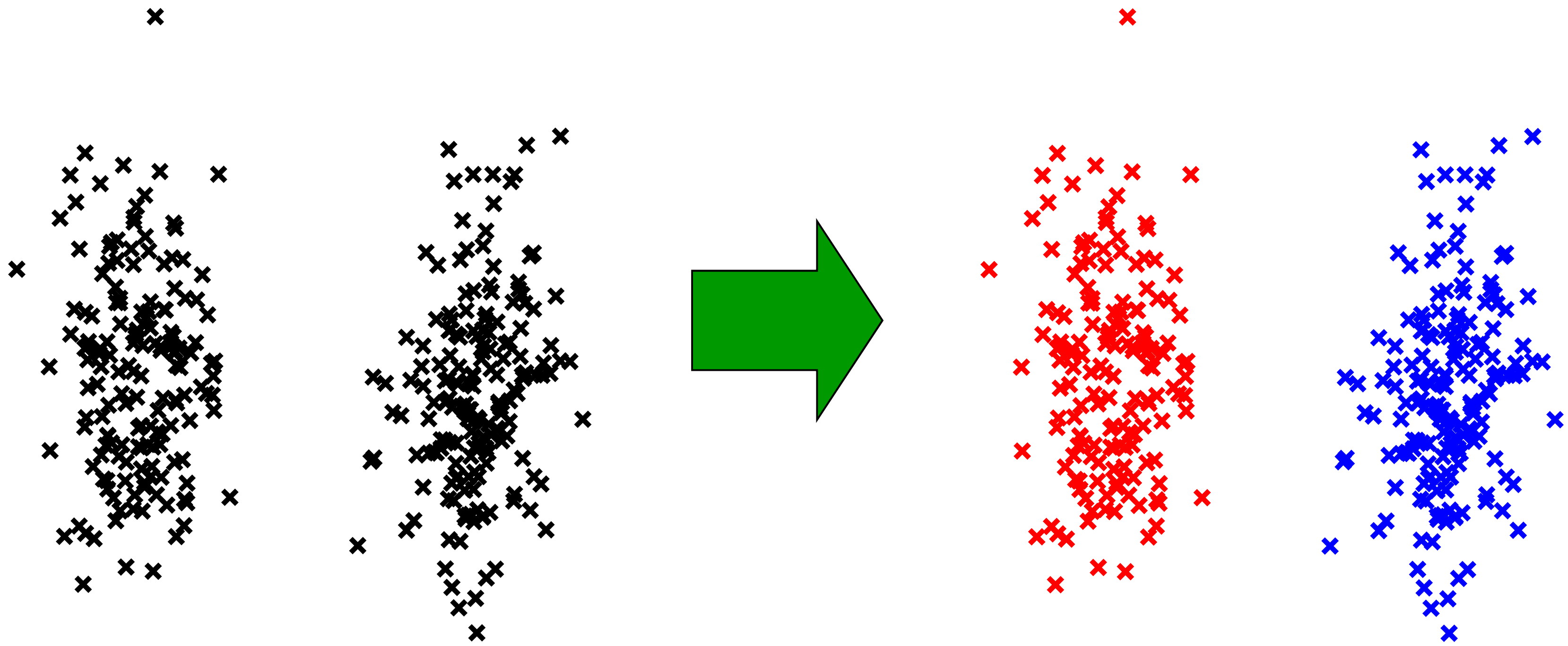
$$\boldsymbol{x}_j \rightarrow \mathcal{C}_{t_j} \qquad t_j = \underset{i}{\mathrm{argmin}} \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|^2$$

  - Update cluster centroids: $i = 1, 2, \ldots, k$

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} \boldsymbol{x}'$$

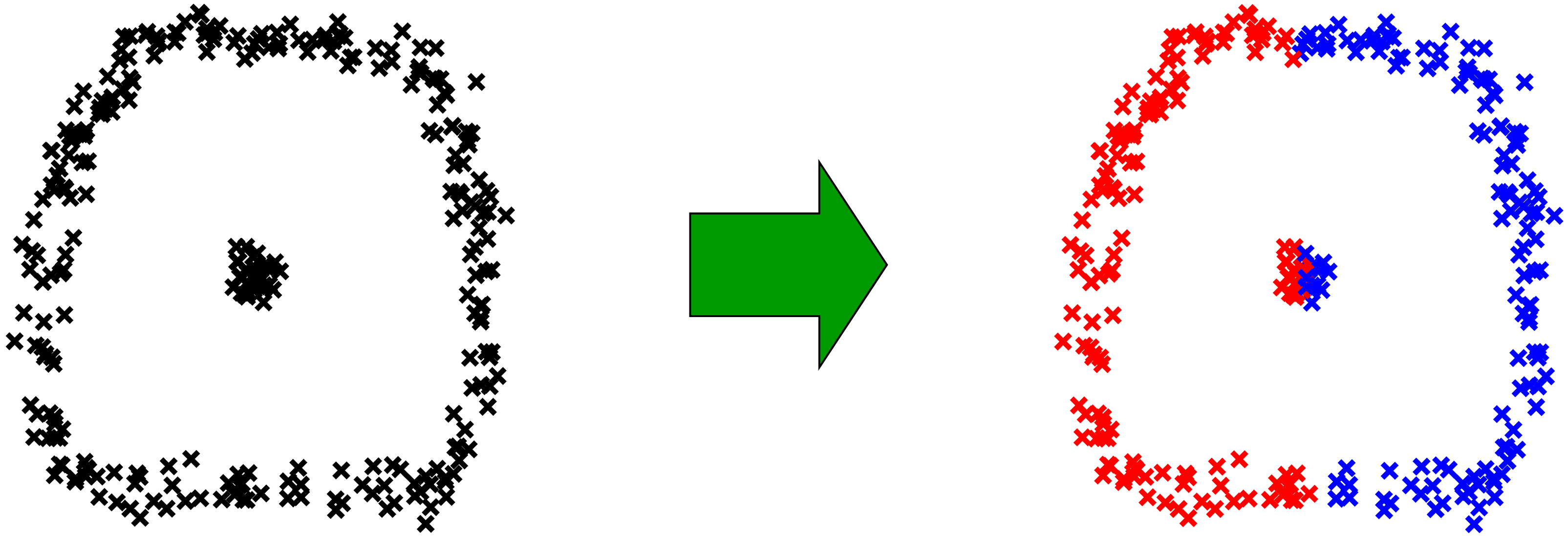
Note: Only local optimality is guaranteed

Examples



- K-means method can successfully separate the two data crowds from each other.

Examples (cont.)



- However, it does not work well if the data crowds have non-convex shapes.

Non-Linearizing K-Means

- Map the original data to a feature space by a non-linear transformation:

$$\phi : \boldsymbol{x} \rightarrow \boldsymbol{f} \qquad \{\boldsymbol{f}_i \mid \boldsymbol{f}_i = \phi(\boldsymbol{x}_i)\}_{i=1}^{n}$$

- Run the k-means algorithm in the feature space.

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

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

Kernel K-Means Algorithm

- Randomly initialize cluster partition: $\{\mathcal{C}_j\}_{j=1}^{k}$
- Update cluster assignments until convergence:

$$\boldsymbol{x}_j \rightarrow \mathcal{C}_{t_j} \qquad j = 1, 2, \ldots, n$$

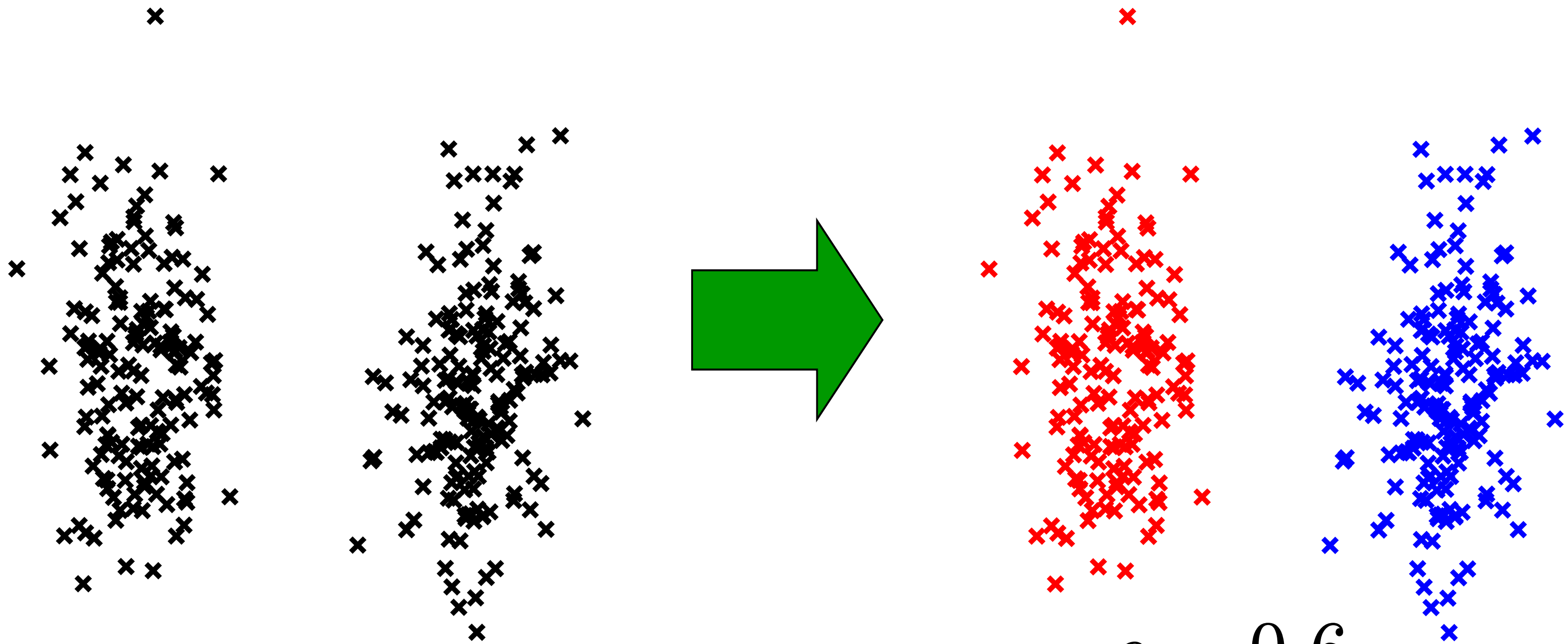
$$t_j = \operatorname*{argmin}_i \left[-\frac{2}{|\mathcal{C}_i|} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} K(\boldsymbol{x}_j, \boldsymbol{x}') + \frac{1}{|\mathcal{C}_i|^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} K(\boldsymbol{x}', \boldsymbol{x}'') \right]$$

$$\|\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i\|^2 = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}) \rangle - 2\langle \phi(\boldsymbol{x}), \boldsymbol{\mu}_i \rangle + \langle \boldsymbol{\mu}_i, \boldsymbol{\mu}_i \rangle$$

$$= \underbrace{K(\boldsymbol{x}, \boldsymbol{x})}_{\text{constant}} - \frac{2}{|\mathcal{C}_i|} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} K(\boldsymbol{x}, \boldsymbol{x}') + \frac{1}{|\mathcal{C}_i|^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} K(\boldsymbol{x}', \boldsymbol{x}'')$$

Examples of Kernel K-Means

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$

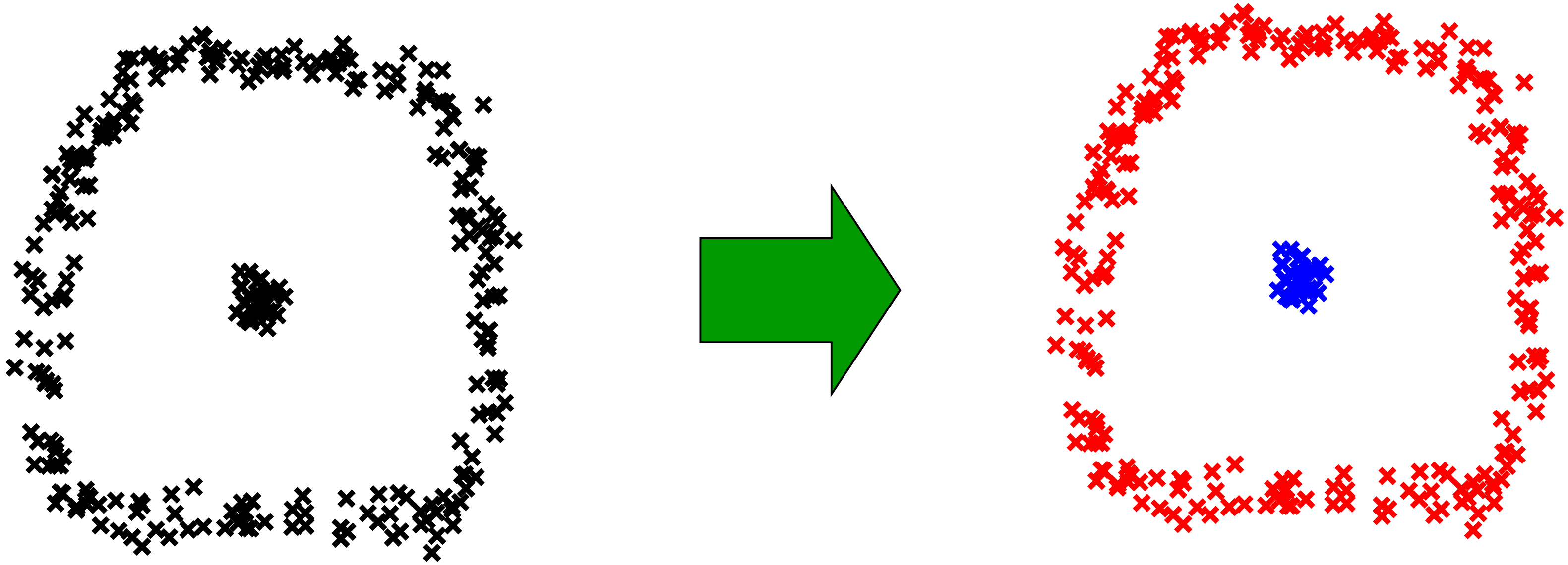


$$c = 0.6$$

- Kernel k-means method can separate the two data crowds successfully.

Examples of Kernel K-Means (cont.)

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$

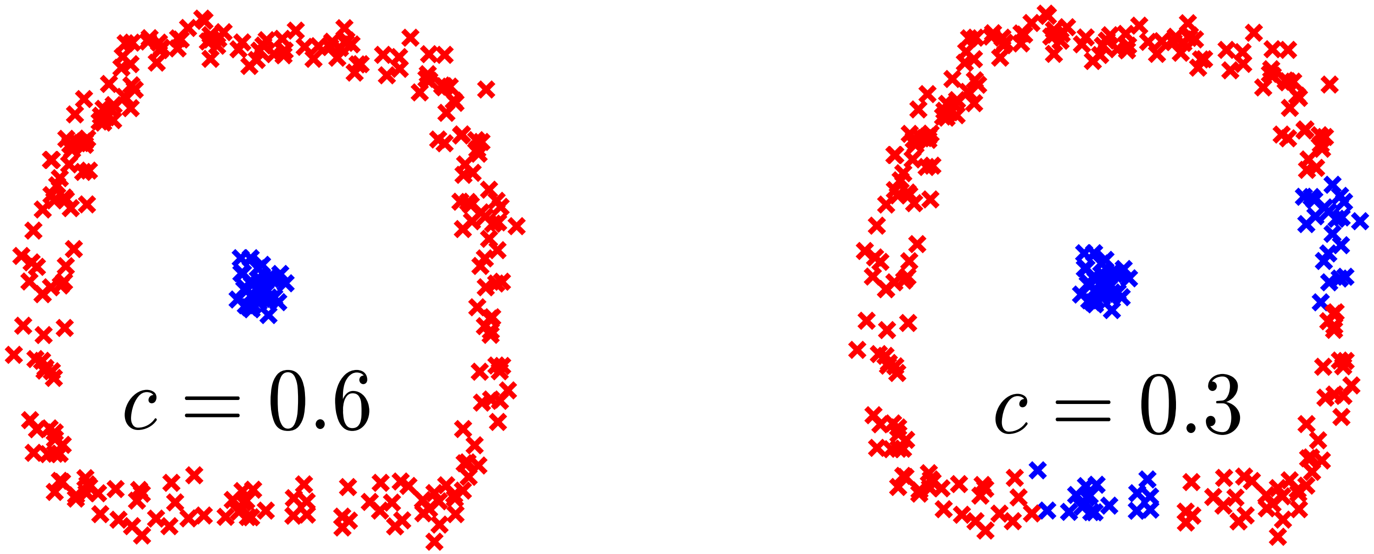


$c = 0.6$

- It also works well for data with non-convex shapes.

Examples of Kernel K-Means (cont.)

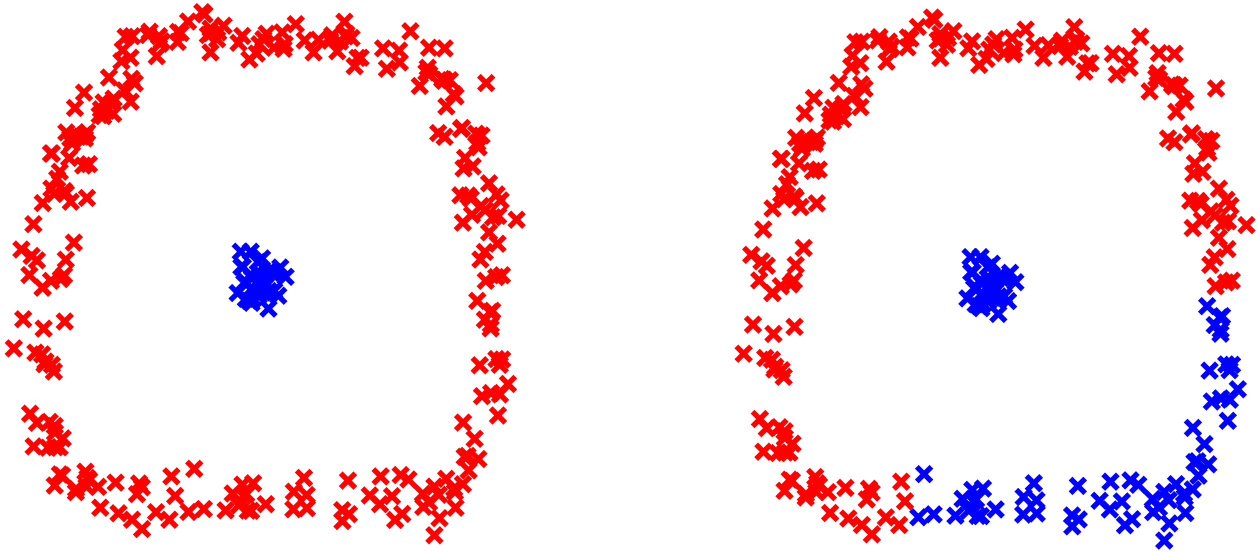
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$



- Choice of kernels (type and parameter) depends on the result.
- Appropriately choosing kernels is not easy in practice.

Examples of Kernel K-Means (cont.)

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$



- Solution depends crucially on the initial cluster assignments since clustering is carried out in a high-dimensional feature space.

Weighted Scatter Criterion

- We assign a positive weight $d(\boldsymbol{x})$ for each sample $\boldsymbol{x}$:

$$\min_{\{\mathcal{C}_i\}_{i=1}^{k}} [J_{WS}]$$

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

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

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

Exercise

- Prove that

$$\underset{i}{\operatorname{argmin}} \left[d(\boldsymbol{x}) \| \phi(\boldsymbol{x}) - \boldsymbol{\mu}_i \|^2 \right]$$

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

is equivalent to

$$\underset{i}{\operatorname{argmin}} \left[-\frac{2}{s_i} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} d(\boldsymbol{x}') K(\boldsymbol{x}_j, \boldsymbol{x}') \right.$$

$$\left. + \frac{1}{s_i^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} d(\boldsymbol{x}') d(\boldsymbol{x}'') K(\boldsymbol{x}', \boldsymbol{x}'') \right]$$

Proof

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

$$d(\boldsymbol{x})\|\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i\|^2$$

$$= d(\boldsymbol{x})\Big(\langle\phi(\boldsymbol{x}), \phi(\boldsymbol{x})\rangle - 2\langle\phi(\boldsymbol{x}), \boldsymbol{\mu}_i\rangle + \langle\boldsymbol{\mu}_i, \boldsymbol{\mu}_i\rangle\Big)$$

$$= d(\boldsymbol{x})\left(K(\boldsymbol{x}, \boldsymbol{x}) - \frac{2}{s_i} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} d(\boldsymbol{x}')K(\boldsymbol{x}, \boldsymbol{x}') + \frac{1}{s_i^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} d(\boldsymbol{x}')d(\boldsymbol{x}'')K(\boldsymbol{x}', \boldsymbol{x}'')\right)$$

independent of i

Weighted Kernel K-Means

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

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

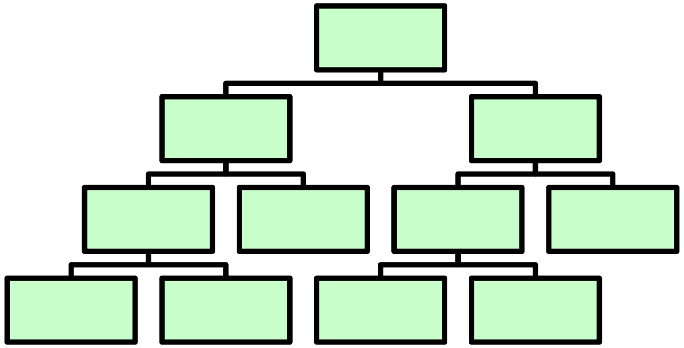
$$t = \mathop{\mathrm{argmin}}_i \Bigg[-\frac{2}{s_i} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} d(\boldsymbol{x}') K(\boldsymbol{x}_j, \boldsymbol{x}')$$

$$+\frac{1}{s_i^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} d(\boldsymbol{x}') d(\boldsymbol{x}'') K(\boldsymbol{x}', \boldsymbol{x}'') \Bigg]$$

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

Hierarchical Clustering

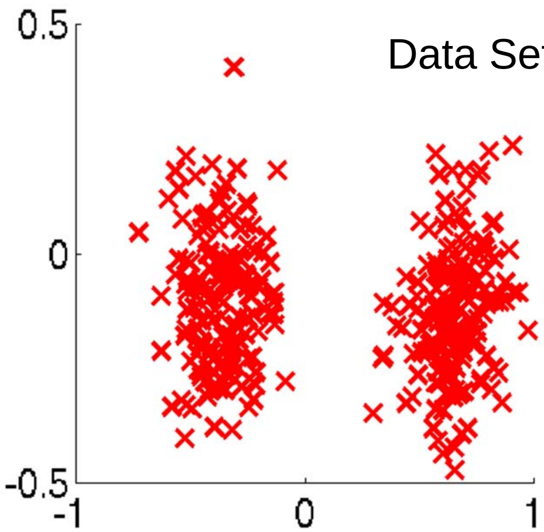
- Hierarchical cluster structure can be obtained recursively clustering the data.
- We may fix $k = 2$.

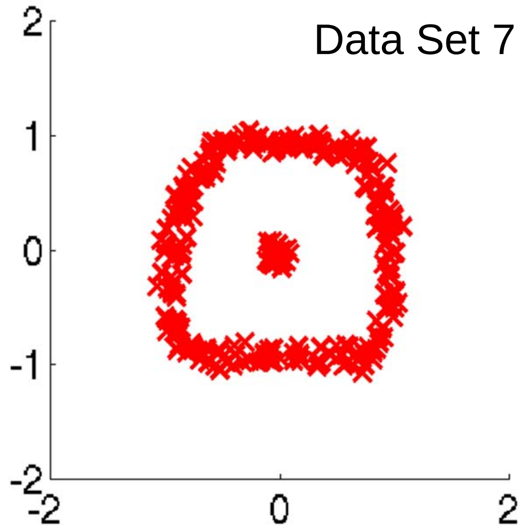


Homework

- Implement linear/kernel k-means algorithms and reproduce the 2-dimensional examples shown in the class.

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





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