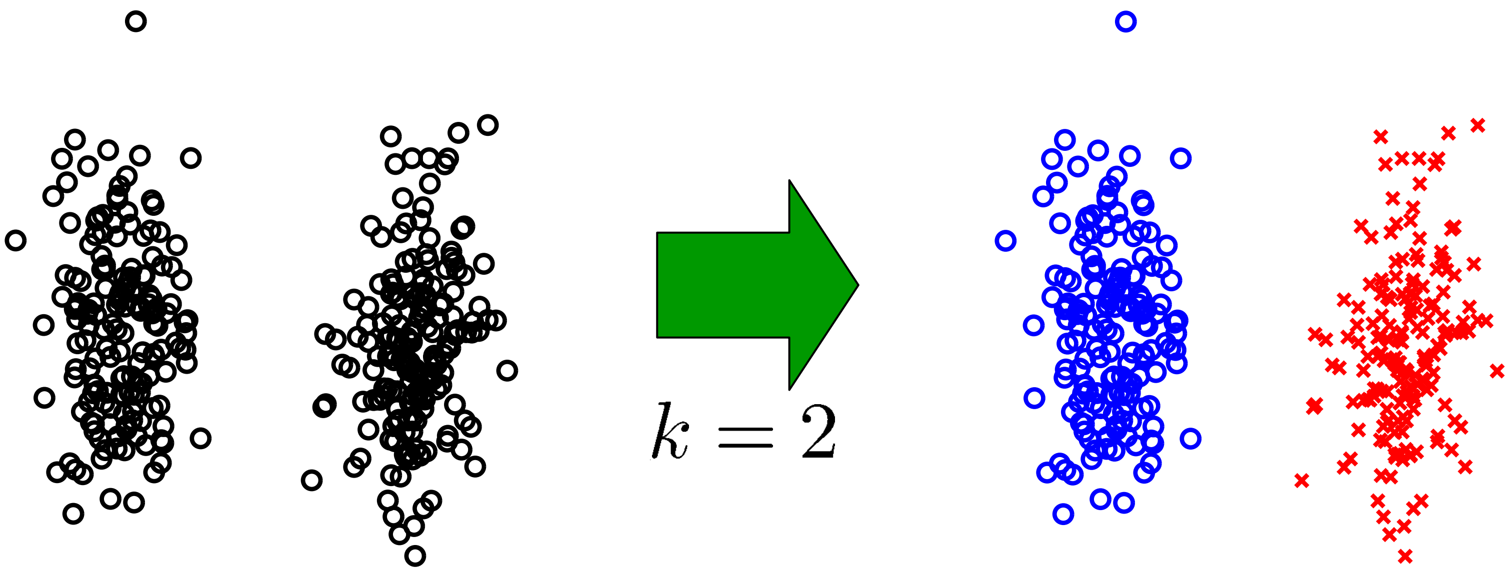
Data Clustering

- We want to divide data samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ into k $(1 \leq k \leq n)$ disjoint groups, s.t. samples in the same group have similar characteristics.



Within-Class Scatter Criterion

- Basic idea: Divide the samples so that within-class scatter is minimized.
- $\mathcal{C}_i$: Set of samples in class i

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

- Criterion:

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

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

Within-Class 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 simply 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.

Numerical Method: K-Means Clustering Algorithm

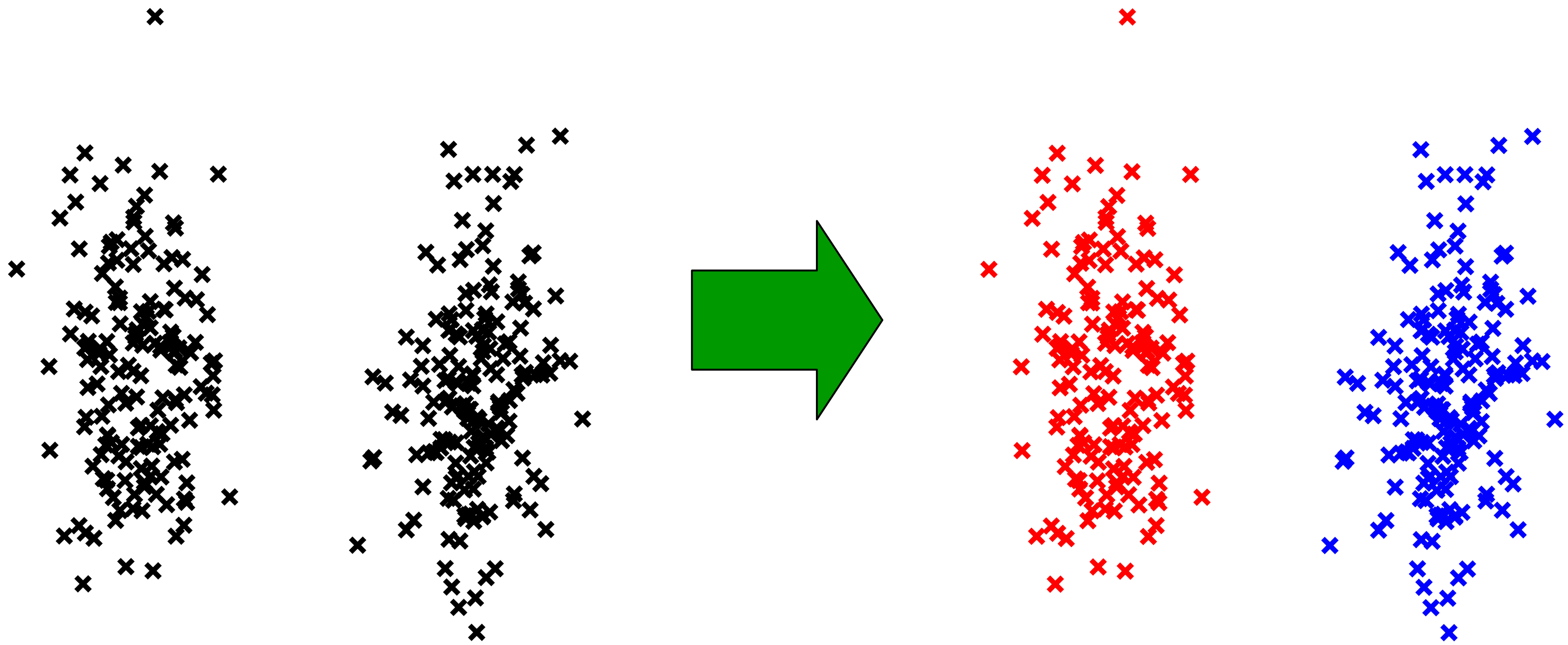
- Randomly initialize partition: $\{\mathcal{C}_i\}_{i=1}^{k}$
- Update class assignments until convergence:

$$\boldsymbol{x}_j \rightarrow \mathcal{C}_t \qquad t = \underset{i}{\operatorname{argmin}} \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|^2$$

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

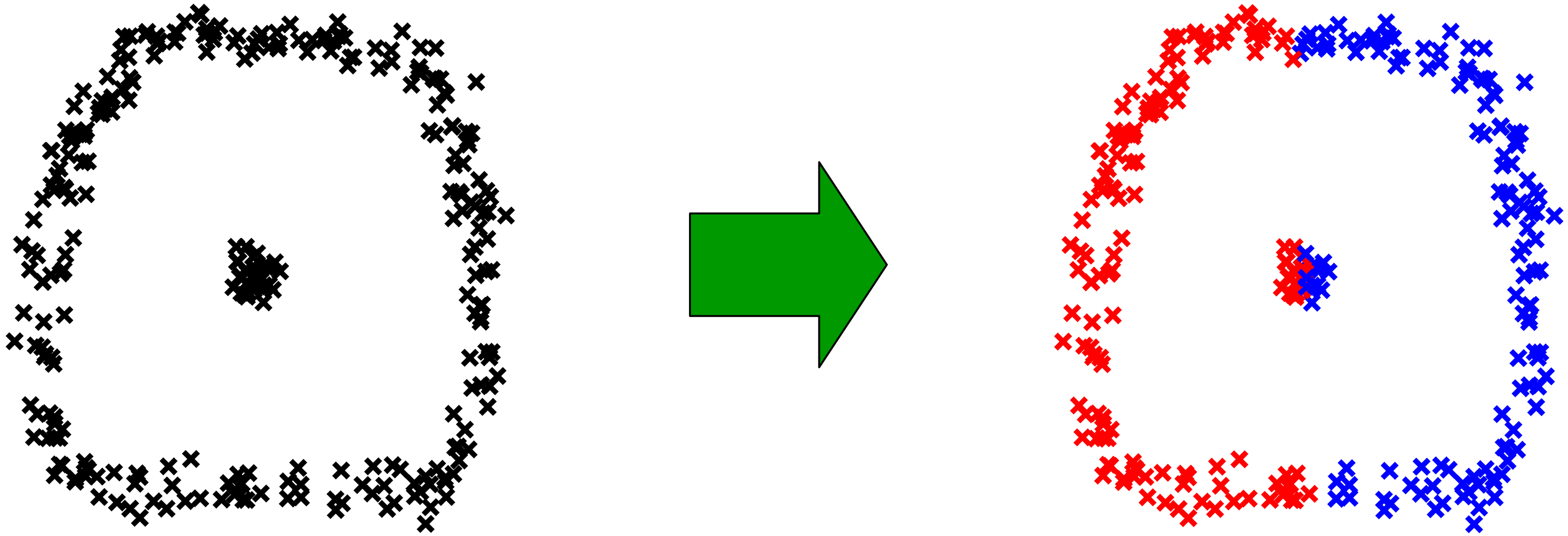
Note: Only local optimality is guaranteed

Examples: demo(1)



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

Examples: demo(2)



- 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

$$
\begin{aligned}
&\|\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i\|^2 \\
&\quad = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}) \rangle - 2\langle \phi(\boldsymbol{x}), \boldsymbol{\mu}_i \rangle + \langle \boldsymbol{\mu}_i, \boldsymbol{\mu}_i \rangle \\
&\quad = K(\boldsymbol{x}, \boldsymbol{x}) - \frac{2}{|\mathcal{C}_i|} \sum_{\boldsymbol{x}' \in \mathcal{C}_i} K(\boldsymbol{x}, \boldsymbol{x}') + \frac{1}{|\mathcal{C}_i|^2} \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} K(\boldsymbol{x}', \boldsymbol{x}'')
\end{aligned}
$$

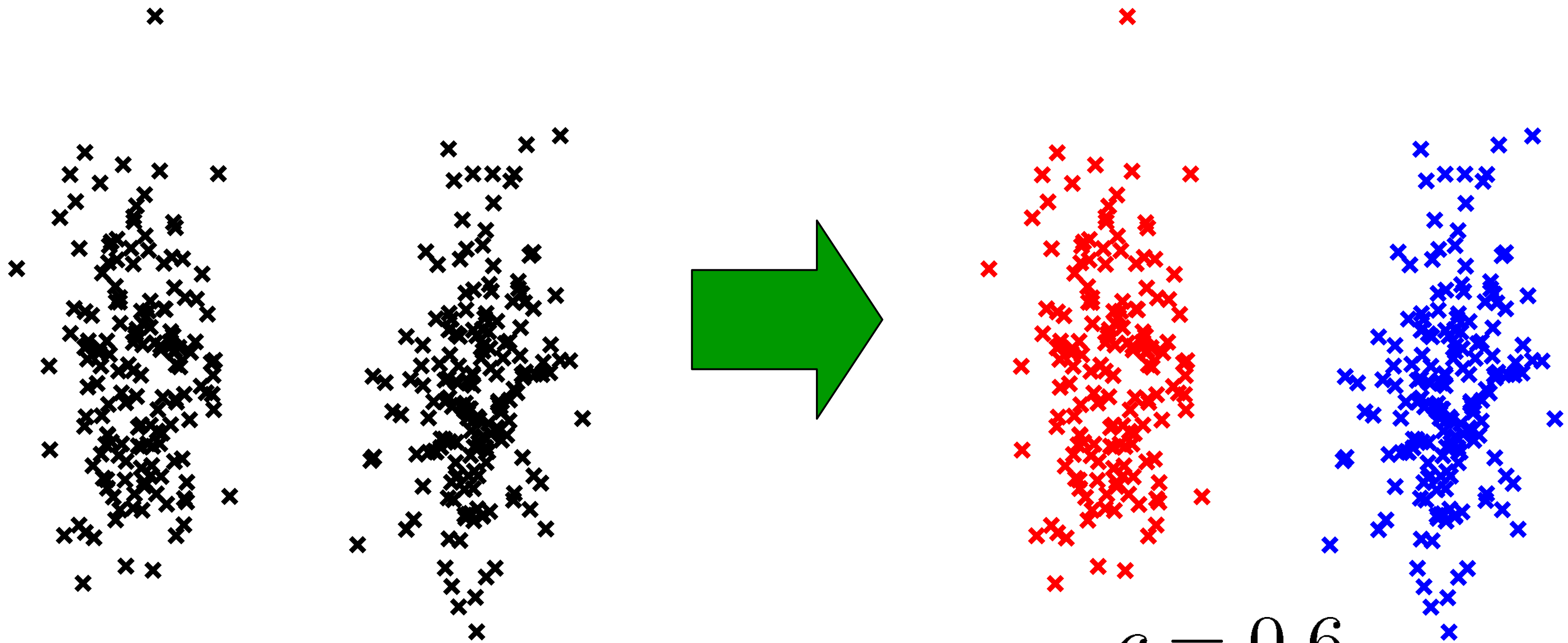
1. Randomly initialize partition: $\{\mathcal{C}_j\}_{j=1}^{k}$
2. Update class assignments until convergence:

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

$$t = \operatorname*{argmax}_i \left[2|\mathcal{C}_i| \sum_{\boldsymbol{x}' \in \mathcal{C}_i} K(\boldsymbol{x}_j, \boldsymbol{x}') - \sum_{\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{C}_i} K(\boldsymbol{x}', \boldsymbol{x}'') \right]$$

Examples: demo(3)

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

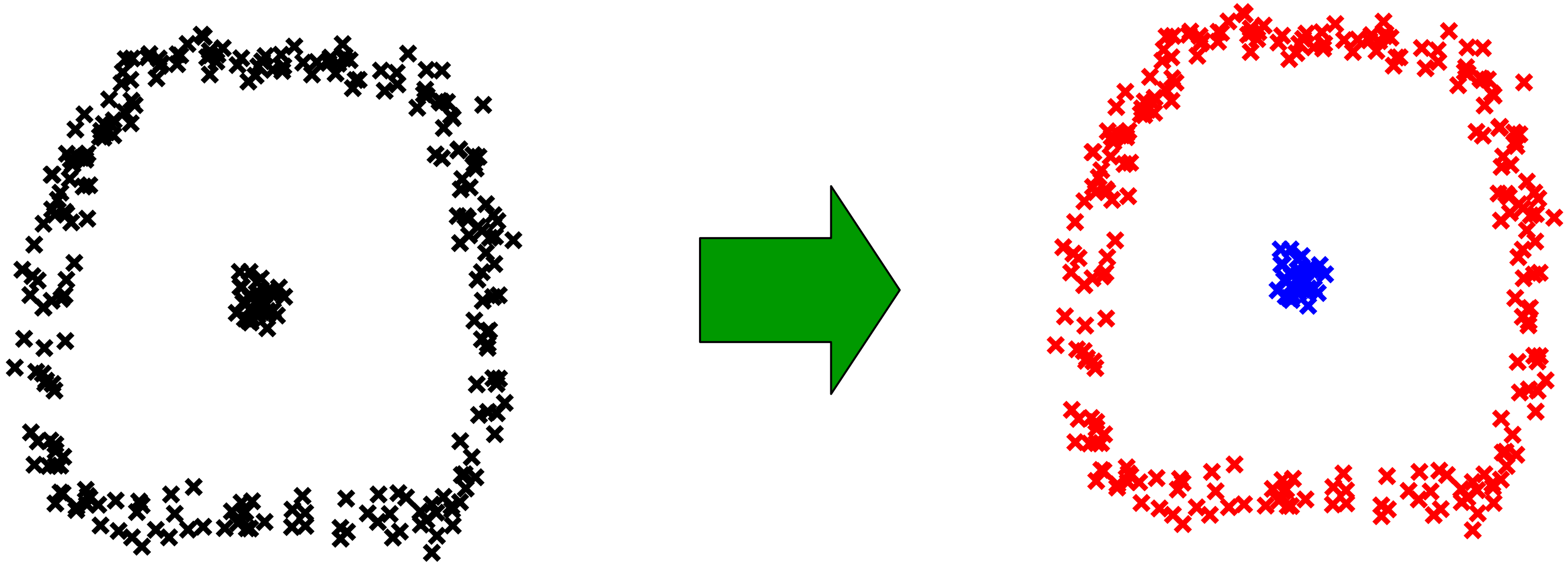


$c = 0.6$

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

Examples: demo(4)

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

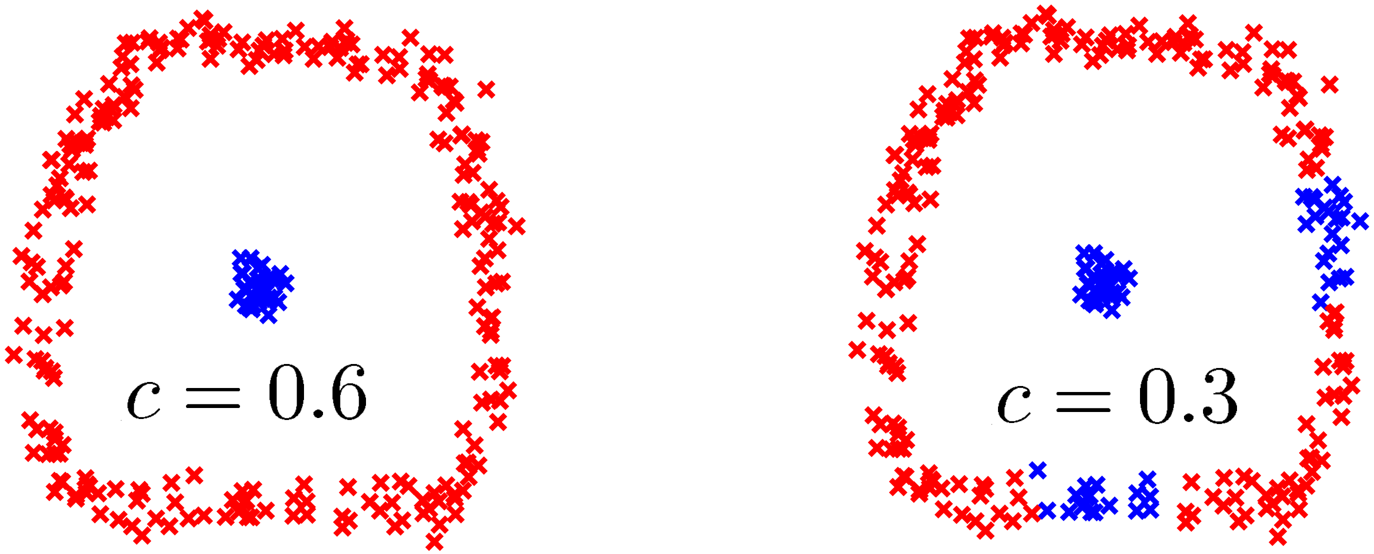


$c = 0.6$

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

Examples: demo(5)

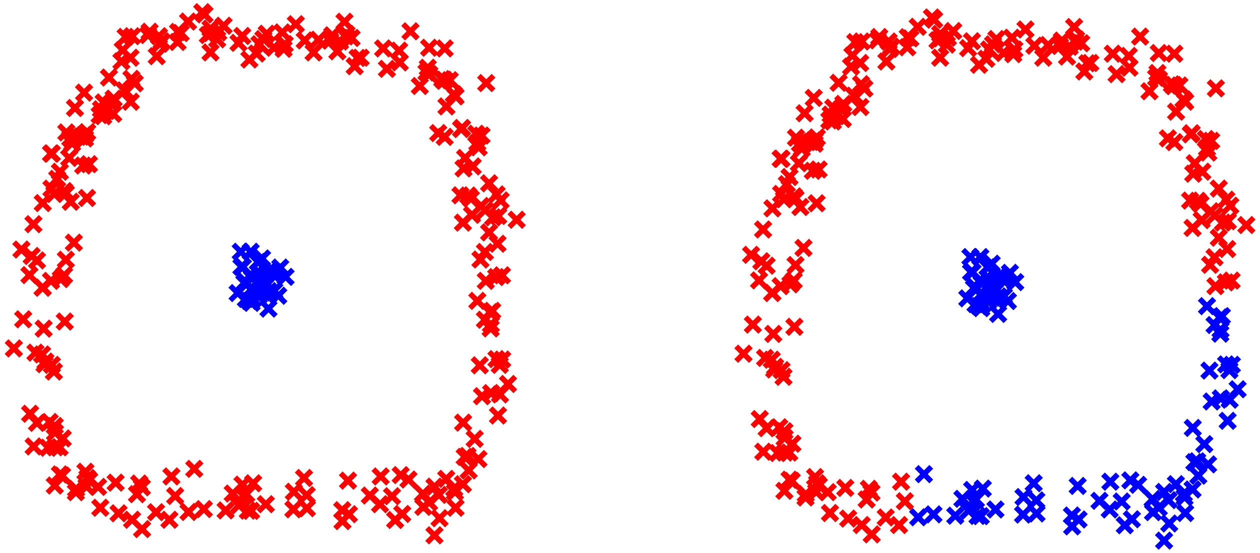
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$



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

Examples: demo(6)

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



- Solution depends on the initial cluster assignments.

Weighted Scatter Criterion

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

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

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

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

Weighted Kernel K-Means

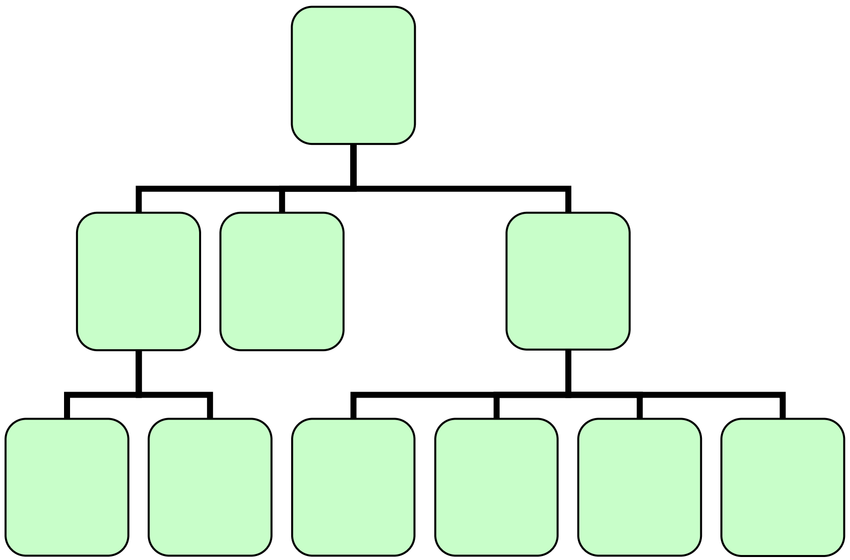
1. Randomly initialize partition: $\{\mathcal{C}_i\}_{i=1}^{k}$
2. Update class assignments until convergence:

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

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

Hierarchical Clustering

- Obtain hierarchical cluster structure.
- It may be achieved by recursively clustering the data.



Notification: Final Assignment

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

Mini-Conference on Data Mining

- In July 12th (final class), we have a mini-conference on data mining, instead of regular lecture.
- Some of you (5-10 students?) may present their data mining results.
- Those who gave a talk at the conference will have very good grades!

Mini-Conference on Data Mining

- Application deadline: July 5^{th}
- Presentation: 10-15 min.
  - Description of your data
  - Methods to be used
  - Outcome
- OHP or projector may be used.
- Slides should be in English.
- Better to speak in English, but Japanese is also allowed.