Clustering: K-means and Kernel K-means

Piyush Rai

Machine Learning (CS771A)

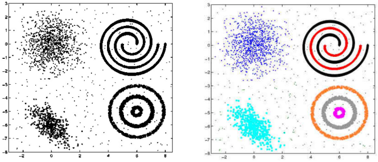
Aug 31, 2016

Clustering

- Usually an unsupervised learning problem
- Given: N unlabeled examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; no. of desired partitions K
- Goal: Group the examples into K "homogeneous" partitions

Clustering

- Usually an unsupervised learning problem
- Given: N unlabeled examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; no. of desired partitions K
- Goal: Group the examples into K "homogeneous" partitions

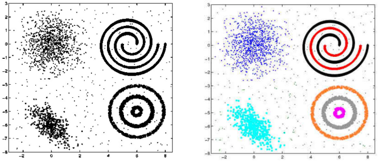


(a) Input data (b) Desired clustering

Picture courtesy: "Data Clustering: 50 Years Beyond K-Means", A.K. Jain (2008)

Clustering

- Usually an unsupervised learning problem
- Given: N unlabeled examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; no. of desired partitions K
- Goal: Group the examples into K "homogeneous" partitions



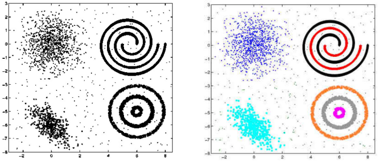
(a) Input data (b) Desired clustering

Picture courtesy: "Data Clustering: 50 Years Beyond K-Means", A.K. Jain (2008)

- Loosely speaking, it is classification without ground truth labels

Clustering

- Usually an unsupervised learning problem
- Given: N unlabeled examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; no. of desired partitions K
- Goal: Group the examples into K "homogeneous" partitions



(a) Input data (b) Desired clustering

Picture courtesy: "Data Clustering: 50 Years Beyond K-Means", A.K. Jain (2008)

- Loosely speaking, it is classification without ground truth labels
- A good clustering is one that achieves:
  - High within-cluster similarity
  - Low inter-cluster similarity

Similarity can be Subjective

- Clustering only looks at similarities, no labels are given
- Without labels, similarity can be hard to define

Similarity can be Subjective

- Clustering only looks at similarities, no labels are given
- Without labels, similarity can be hard to define

- Thus using the right distance/similarity is very important in clustering

Similarity can be Subjective

- Clustering only looks at similarities, no labels are given
- Without labels, similarity can be hard to define

- Thus using the right distance/similarity is very important in clustering
- Also important to define/ask: "Clustering based on what"?

Similarity can be Subjective

- Clustering only looks at similarities, no labels are given
- Without labels, similarity can be hard to define

- Thus using the right distance/similarity is very important in clustering
- Also important to define/ask: "Clustering based on what"?

Picture courtesy: `http://www.guy-sports.com/humor/videos/powerpoint_presentation_dogs.htm`

Clustering: Some Examples

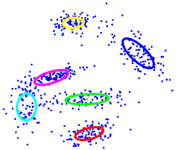
- Document/Image/Webpage Clustering
- Image Segmentation (clustering pixels)

- Clustering web-search results
- Clustering (people) nodes in (social) networks/graphs
- .. and many more..

Picture courtesy: `http://people.cs.uchicago.edu/~pff/segment/`

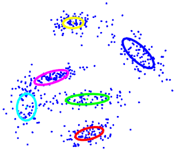
Types of Clustering

1. **Flat or Partitional clustering**
   - Partitions are independent of each other

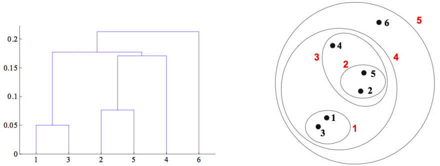


Types of Clustering

1. **Flat or Partitional clustering**
   - Partitions are independent of each other

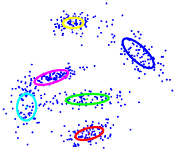


2. **Hierarchical clustering**
   - Partitions can be visualized using a tree structure (a dendrogram)

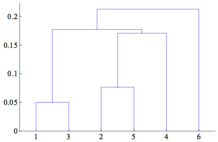


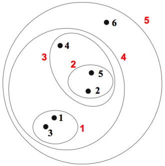
Types of Clustering

1. **Flat or Partitional clustering**
   - Partitions are independent of each other



2. **Hierarchical clustering**
   - Partitions can be visualized using a tree structure (a dendrogram)





   - Possible to view partitions at different levels of granularities (i.e., can refine/coarsen clusters) using different K

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**
  - (Re)-Assign each example $\boldsymbol{x}_n$ to its closest cluster center (based on the smallest Euclidean distance)

$$\boxed{\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2\}}$$

  ($\mathcal{C}_k$ is the set of examples assigned to cluster k with center $\boldsymbol{\mu}_k$)

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**
  - (Re)-Assign each example $\boldsymbol{x}_n$ to its closest cluster center (based on the smallest Euclidean distance)

$$\boxed{\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2\}}$$

  ($\mathcal{C}_k$ is the set of examples assigned to cluster k with center $\boldsymbol{\mu}_k$)

  - Update the cluster means

$$\boxed{\mu_k = \text{mean}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \boldsymbol{x}_n}$$

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**
  - (Re)-Assign each example $\boldsymbol{x}_n$ to its closest cluster center (based on the smallest Euclidean distance)

$$\boxed{\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2\}}$$

    ($\mathcal{C}_k$ is the set of examples assigned to cluster k with center $\boldsymbol{\mu}_k$)

  - Update the cluster means

$$\boxed{\mu_k = \text{mean}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \boldsymbol{x}_n}$$

  - Repeat while not converged

Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$; $\boldsymbol{x}_n \in \mathbb{R}^D$; the number of partitions K
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, each $\boldsymbol{\mu}_k \in \mathbb{R}^D$
  - Usually initialized randomly, but good initialization is crucial; many smarter initialization heuristics exist (e.g., K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**
  - (Re)-Assign each example $\boldsymbol{x}_n$ to its closest cluster center (based on the smallest Euclidean distance)

$$\boxed{\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2\}}$$

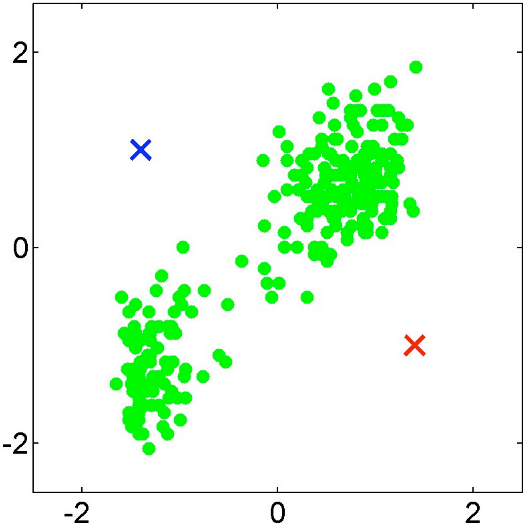
  ($\mathcal{C}_k$ is the set of examples assigned to cluster k with center $\boldsymbol{\mu}_k$)

  - Update the cluster means

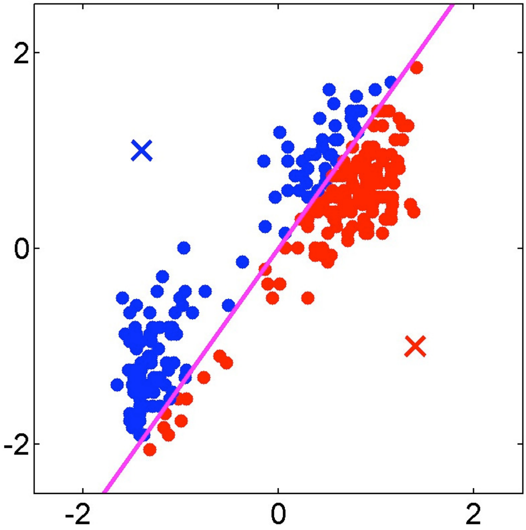
$$\boxed{\mu_k = \text{mean}(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \boldsymbol{x}_n}$$

  - Repeat while not converged
- Stop when cluster means or the "loss" does not change by much

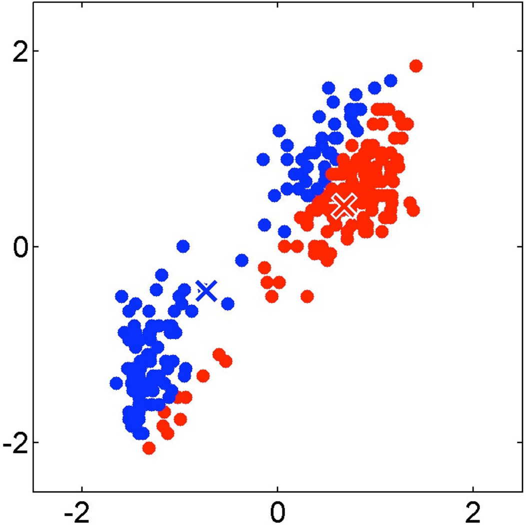
K-means: Initialization (assume $K = 2$)



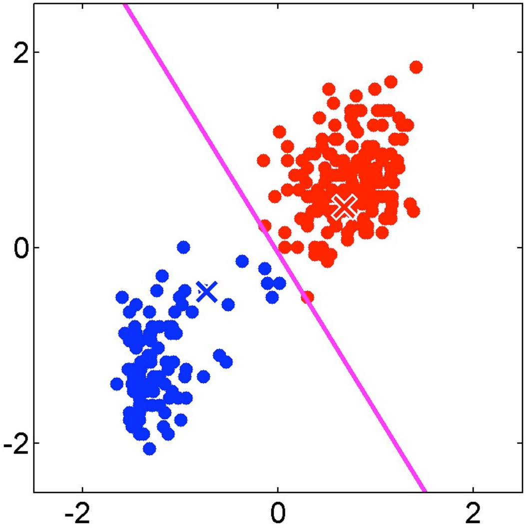
K-means iteration 1: Assigning points



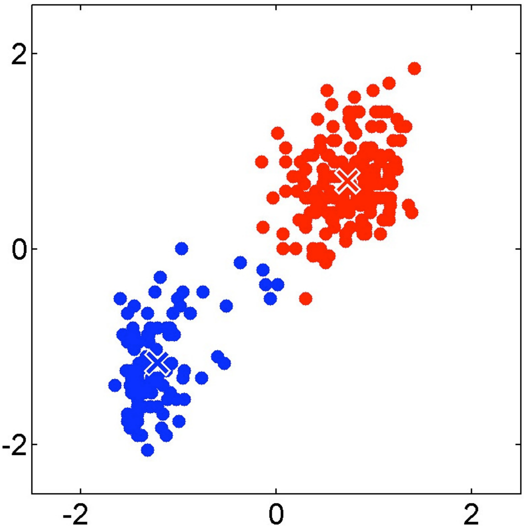
K-means iteration 1: Recomputing the centers



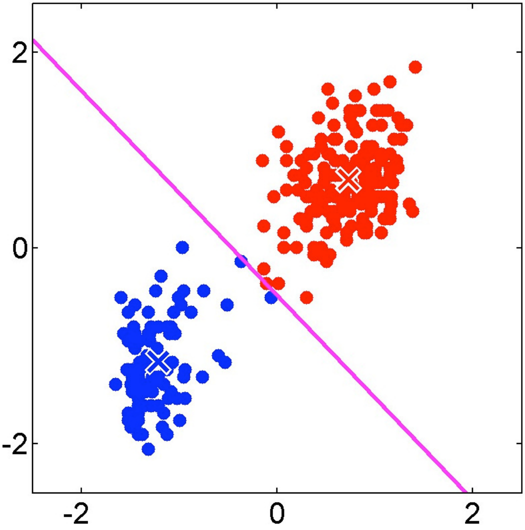
K-means iteration 2: Assigning points



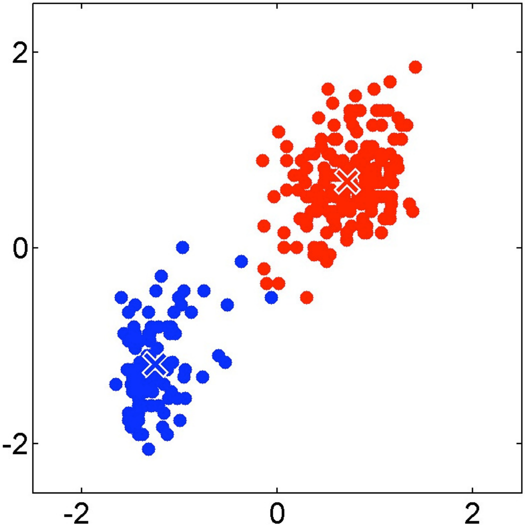
K-means iteration 2: Recomputing the centers



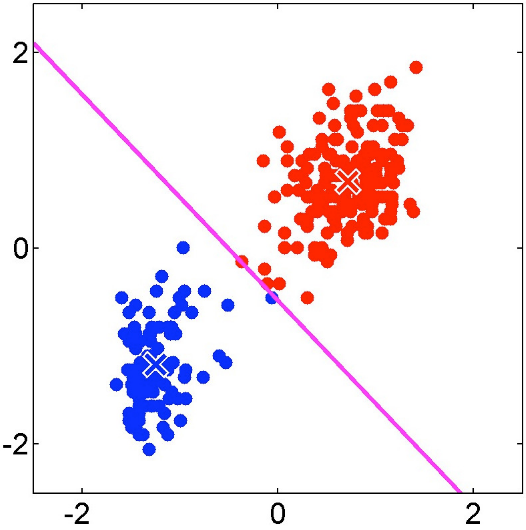
K-means iteration 3: Assigning points



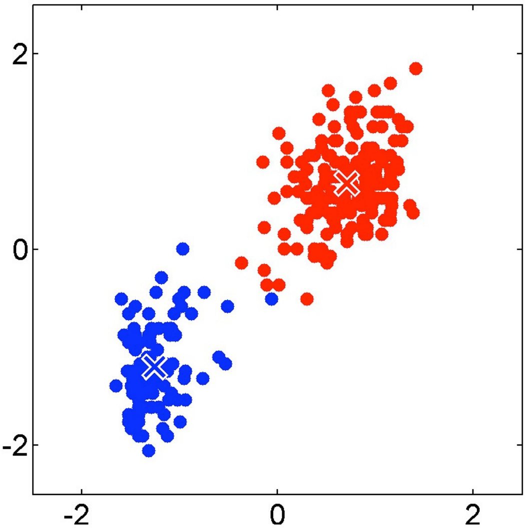
K-means iteration 3: Recomputing the centers



K-means iteration 4: Assigning points



K-means iteration 4: Recomputing the centers



What Loss Function is K-means Optimizing?

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise
  - Note: $\boldsymbol{z}_n = [z_{n1} \; z_{n2} \; \ldots \; z_{nK}]$ represents a length K one-hot encoding of $\boldsymbol{x}_n$

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise
  - Note: $\boldsymbol{z}_n = [z_{n1} \; z_{n2} \; \ldots \; z_{nK}]$ represents a length K one-hot encoding of $\boldsymbol{x}_n$
- Define the distortion or "loss" for the cluster assignment of $\boldsymbol{x}_n$

$$\ell(\boldsymbol{\mu}, \boldsymbol{x}_n, \boldsymbol{z}_n) = \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise
  - Note: $\boldsymbol{z}_n = [z_{n1} \; z_{n2} \; \ldots \; z_{nK}]$ represents a length K one-hot encoding of $\boldsymbol{x}_n$
- Define the distortion or "loss" for the cluster assignment of $\boldsymbol{x}_n$

$$\ell(\boldsymbol{\mu}, \boldsymbol{x}_n, \boldsymbol{z}_n) = \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Total distortion over all points defines the K-means "loss function"

$$L(\mu, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2$$

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise
  - Note: $\boldsymbol{z}_n = [z_{n1} \; z_{n2} \; \ldots \; z_{nK}]$ represents a length K one-hot encoding of $\boldsymbol{x}_n$
- Define the distortion or "loss" for the cluster assignment of $\boldsymbol{x}_n$

$$\ell(\boldsymbol{\mu}, \boldsymbol{x}_n, \boldsymbol{z}_n) = \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Total distortion over all points defines the K-means "loss function"

$$L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2 = ||\mathbf{X} - \mathbf{Z}\boldsymbol{\mu}||^2$$

where $\mathbf{Z}$ is $N \times K$ (row n is $\boldsymbol{z}_n$) and $\boldsymbol{\mu}$ is $K \times D$ (row k is $\boldsymbol{\mu}_k$)

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise
  - Note: $\boldsymbol{z}_n = [z_{n1} \; z_{n2} \; \ldots \; z_{nK}]$ represents a length K one-hot encoding of $\boldsymbol{x}_n$
- Define the distortion or "loss" for the cluster assignment of $\boldsymbol{x}_n$

$$\ell(\boldsymbol{\mu}, \boldsymbol{x}_n, \boldsymbol{z}_n) = \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Total distortion over all points defines the K-means "loss function"

$$L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2 = ||\mathbf{X} - \mathbf{Z}\boldsymbol{\mu}||^2$$

  where $\mathbf{Z}$ is $N \times K$ (row n is $\boldsymbol{z}_n$) and $\boldsymbol{\mu}$ is $K \times D$ (row k is $\boldsymbol{\mu}_k$)
- The K-means **problem** is to minimize this objective w.r.t. $\boldsymbol{\mu}$ and $\mathbf{Z}$

What Loss Function is K-means Optimizing?

- Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ be the K cluster centroids (means)
- Let $z_{nk} \in \{0, 1\}$ be s.t. $z_{nk} = 1$ if $\boldsymbol{x}_n$ belongs to cluster k, and 0 otherwise
  - Note: $\boldsymbol{z}_n = [z_{n1} \; z_{n2} \; \ldots \; z_{nK}]$ represents a length K one-hot encoding of $\boldsymbol{x}_n$
- Define the distortion or "loss" for the cluster assignment of $\boldsymbol{x}_n$

$$\ell(\boldsymbol{\mu}, \boldsymbol{x}_n, \boldsymbol{z}_n) = \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Total distortion over all points defines the K-means "loss function"

$$L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 = ||\mathbf{X} - \mathbf{Z}\boldsymbol{\mu}||^2$$

where $\mathbf{Z}$ is $N \times K$ (row n is $\boldsymbol{z}_n$) and $\boldsymbol{\mu}$ is $K \times D$ (row k is $\boldsymbol{\mu}_k$)

- The K-means **problem** is to minimize this objective w.r.t. $\boldsymbol{\mu}$ and $\mathbf{Z}$
  - Note that the objective only minimizes within-cluster distortions

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

- It is a non-convex objective function
  - Many local minima possible
- Also NP-hard to minimize in general (note that $\mathbf{Z}$ is discrete)

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

- It is a non-convex objective function
  - Many local minima possible
- Also NP-hard to minimize in general (note that $\mathbf{Z}$ is discrete)
- The K-means algorithm we saw is a heuristic to optimize this function

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

- It is a non-convex objective function
  - Many local minima possible
- Also NP-hard to minimize in general (note that $\mathbf{Z}$ is discrete)
- The K-means algorithm we saw is a heuristic to optimize this function
- K-means algorithm alternated between the following two steps

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

- It is a non-convex objective function
  - Many local minima possible
- Also NP-hard to minimize in general (note that $\mathbf{Z}$ is discrete)
- The K-means algorithm we saw is a heuristic to optimize this function
- K-means algorithm alternated between the following two steps
  - Fix $\boldsymbol{\mu}$, minimize w.r.t. $\mathbf{Z}$ (assign points to closest centers)

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

- It is a non-convex objective function
  - Many local minima possible
- Also NP-hard to minimize in general (note that $\mathbf{Z}$ is discrete)
- The K-means algorithm we saw is a heuristic to optimize this function
- K-means algorithm alternated between the following two steps
  - Fix $\boldsymbol{\mu}$, minimize w.r.t. $\mathbf{Z}$ (assign points to closest centers)
  - Fix $\mathbf{Z}$, minimize w.r.t. $\boldsymbol{\mu}$ (recompute the center means)

K-means Objective

- Consider the K-means objective function

$$\boxed{L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||\boldsymbol{x}_n - \mu_k||^2}$$

- It is a non-convex objective function
  - Many local minima possible
- Also NP-hard to minimize in general (note that $\mathbf{Z}$ is discrete)
- The K-means algorithm we saw is a heuristic to optimize this function
- K-means algorithm alternated between the following two steps
  - Fix $\boldsymbol{\mu}$, minimize w.r.t. $\mathbf{Z}$ (assign points to closest centers)
  - Fix $\mathbf{Z}$, minimize w.r.t. $\boldsymbol{\mu}$ (recompute the center means)
- Note: The algorithm usually converges to a local minima (though may not always, and it may just convergence "somewhere"). Multiple runs with different initializations can be tried to find a good solution.

Convergence of K-means Algorithm

- Each step (updating $\mathbf{Z}$ or $\boldsymbol{\mu}$) can never increase the objective

Convergence of K-means Algorithm

- Each step (updating $\mathbf{Z}$ or $\boldsymbol{\mu}$) can never increase the objective
- When we update $\mathbf{Z}$ from $\mathbf{Z}^{(t-1)}$ to $\mathbf{Z}^{(t)}$

$$L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t-1)})$$

Convergence of K-means Algorithm

- Each step (updating $\mathbf{Z}$ or $\boldsymbol{\mu}$) can never increase the objective
- When we update $\mathbf{Z}$ from $\mathbf{Z}^{(t-1)}$ to $\mathbf{Z}^{(t)}$

$$L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t-1)})$$

because the new $\mathbf{Z}^{(t)} = \arg\min_{\mathbf{Z}} L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z})$

Convergence of K-means Algorithm

- Each step (updating $\mathbf{Z}$ or $\boldsymbol{\mu}$) can never increase the objective
- When we update $\mathbf{Z}$ from $\mathbf{Z}^{(t-1)}$ to $\mathbf{Z}^{(t)}$

$$L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t-1)})$$

  because the new $\mathbf{Z}^{(t)} = \arg\min_{\mathbf{Z}} L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z})$

- When we update $\boldsymbol{\mu}$ from $\boldsymbol{\mu}^{(t-1)}$ to $\boldsymbol{\mu}^{(t)}$

$$L(\boldsymbol{\mu}^{(t)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)})$$

Convergence of K-means Algorithm

- Each step (updating $\mathbf{Z}$ or $\boldsymbol{\mu}$) can never increase the objective
- When we update $\mathbf{Z}$ from $\mathbf{Z}^{(t-1)}$ to $\mathbf{Z}^{(t)}$

$$L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t-1)})$$

  because the new $\mathbf{Z}^{(t)} = \arg\min_{\mathbf{Z}} L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z})$
- When we update $\boldsymbol{\mu}$ from $\boldsymbol{\mu}^{(t-1)}$ to $\boldsymbol{\mu}^{(t)}$

$$L(\boldsymbol{\mu}^{(t)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)})$$

  because the new $\boldsymbol{\mu}^{(t)} = \arg\min_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}^{(t)})$

Convergence of K-means Algorithm

- Each step (updating $\mathbf{Z}$ or $\boldsymbol{\mu}$) can never increase the objective
- When we update $\mathbf{Z}$ from $\mathbf{Z}^{(t-1)}$ to $\mathbf{Z}^{(t)}$

$$L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t-1)})$$

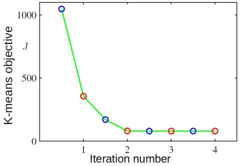
  because the new $\mathbf{Z}^{(t)} = \arg\min_{\mathbf{Z}} L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z})$

- When we update $\boldsymbol{\mu}$ from $\boldsymbol{\mu}^{(t-1)}$ to $\boldsymbol{\mu}^{(t)}$

$$L(\boldsymbol{\mu}^{(t)}, \mathbf{X}, \mathbf{Z}^{(t)}) \leq L(\boldsymbol{\mu}^{(t-1)}, \mathbf{X}, \mathbf{Z}^{(t)})$$

  because the new $\boldsymbol{\mu}^{(t)} = \arg\min_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \mathbf{X}, \mathbf{Z}^{(t)})$

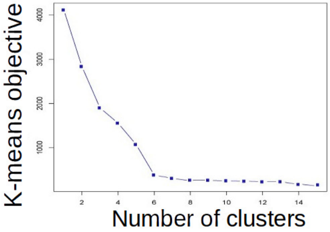
- Thus the K-means algorithm monotonically decreases the objective



(blue: after Z updated, red: after μ updated)

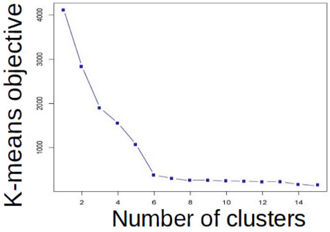
K-means: Choosing K

- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point"



K-means: Choosing K

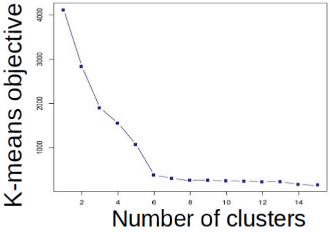
- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point"



- For the above plot, $K = 6$ is the elbow point

K-means: Choosing K

- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point"



- For the above plot, $K = 6$ is the elbow point
- Can also information criterion such as AIC (Akaike Information Criterion)

$$AIC = 2L(\hat{\mu}, \mathbf{X}, \hat{\mathbf{Z}}) + K \log D$$

.. and choose the K that has the smallest AIC (discourages large K)

K-means: Some Limitations

- Makes hard assignments of points to clusters

K-means: Some Limitations

- Makes hard assignments of points to clusters
  - A point either completely belongs to a cluster or doesn't belong at all

K-means: Some Limitations

- Makes hard assignments of points to clusters
  - A point either completely belongs to a cluster or doesn't belong at all
  - No notion of a soft assignment (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point $\boldsymbol{x}_n$, $p_1 = 0.7, p_2 = 0.2, p_3 = 0.1$)

K-means: Some Limitations

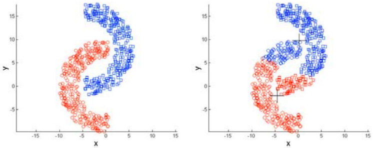
- Makes hard assignments of points to clusters
  - A point either completely belongs to a cluster or doesn't belong at all
  - No notion of a soft assignment (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point $\boldsymbol{x}_n$, $p_1 = 0.7, p_2 = 0.2, p_3 = 0.1$)
- Works well only is the clusters are roughly of equal sizes

K-means: Some Limitations

- Makes hard assignments of points to clusters
  - A point either completely belongs to a cluster or doesn't belong at all
  - No notion of a soft assignment (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point $\boldsymbol{x}_n$, $p_1 = 0.7, p_2 = 0.2, p_3 = 0.1$)
- Works well only is the clusters are roughly of equal sizes
- Probabilistic clustering methods such as Gaussian mixture models can handle both these issues (model each cluster using a Gaussian distribution)

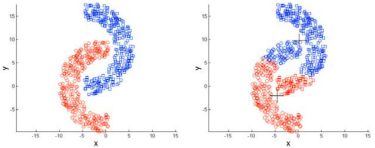
K-means: Some Limitations

- Makes hard assignments of points to clusters
  - A point either completely belongs to a cluster or doesn't belong at all
  - No notion of a soft assignment (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point $\boldsymbol{x}_n$, $p_1 = 0.7, p_2 = 0.2, p_3 = 0.1$)
- Works well only is the clusters are roughly of equal sizes
- Probabilistic clustering methods such as Gaussian mixture models can handle both these issues (model each cluster using a Gaussian distribution)
- K-means also works well only when the clusters are round-shaped and does badly if the clusters have non-convex shapes



K-means: Some Limitations

- Makes hard assignments of points to clusters
  - A point either completely belongs to a cluster or doesn't belong at all
  - No notion of a soft assignment (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point $\boldsymbol{x}_n$, $p_1 = 0.7, p_2 = 0.2, p_3 = 0.1$)
- Works well only is the clusters are roughly of equal sizes
- Probabilistic clustering methods such as Gaussian mixture models can handle both these issues (model each cluster using a Gaussian distribution)
- K-means also works well only when the clusters are round-shaped and does badly if the clusters have non-convex shapes



- Kernel K-means or Spectral clustering can handle non-convex

Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2$$

Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 = ||\phi(\boldsymbol{x}_n)||^2 + ||\phi(\boldsymbol{\mu}_k)||^2 - 2\phi(\boldsymbol{x}_n)^\top \phi(\boldsymbol{\mu}_k)$$

Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$\begin{aligned} ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 &= ||\phi(\boldsymbol{x}_n)||^2 + ||\phi(\boldsymbol{\mu}_k)||^2 - 2\phi(\boldsymbol{x}_n)^\top \phi(\boldsymbol{\mu}_k) \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) + k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k) - 2k(\boldsymbol{x}_n, \boldsymbol{\mu}_k) \end{aligned}$$

- Here $k(.,.)$ denotes the kernel function and ϕ is its (implicit) feature map

Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$
\begin{aligned}
||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 &= ||\phi(\boldsymbol{x}_n)||^2 + ||\phi(\boldsymbol{\mu}_k)||^2 - 2\phi(\boldsymbol{x}_n)^\top \phi(\boldsymbol{\mu}_k) \\
&= k(\boldsymbol{x}_n, \boldsymbol{x}_n) + k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k) - 2k(\boldsymbol{x}_n, \boldsymbol{\mu}_k)
\end{aligned}
$$

- Here $k(.,.)$ denotes the kernel function and ϕ is its (implicit) feature map
- Note: ϕ doesn't have to be computed/stored for data $\{\boldsymbol{x}_n\}_{n=1}^N$ or the cluster means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ because computations only depend on kernel evaluations

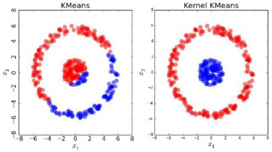
Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$\begin{aligned} ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 &= ||\phi(\boldsymbol{x}_n)||^2 + ||\phi(\boldsymbol{\mu}_k)||^2 - 2\phi(\boldsymbol{x}_n)^\top \phi(\boldsymbol{\mu}_k) \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) + k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k) - 2k(\boldsymbol{x}_n, \boldsymbol{\mu}_k) \end{aligned}$$

- Here $k(.,.)$ denotes the kernel function and ϕ is its (implicit) feature map
- Note: ϕ doesn't have to be computed/stored for data $\{\boldsymbol{x}_n\}_{n=1}^N$ or the cluster means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ because computations only depend on kernel evaluations

Kernel K-means vs. K-means



Pyclust: Open Source Data Clustering Pckage

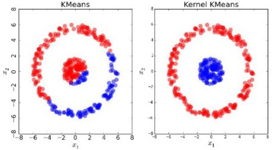
Kernel K-means

- Basic idea: Replace the Euclidean distance/similarity computations in K-means by the kernelized versions. E.g., $d(\boldsymbol{x}_n, \boldsymbol{\mu}_k) = ||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||$ by

$$
\begin{aligned}
||\phi(\boldsymbol{x}_n) - \phi(\boldsymbol{\mu}_k)||^2 &= ||\phi(\boldsymbol{x}_n)||^2 + ||\phi(\boldsymbol{\mu}_k)||^2 - 2\phi(\boldsymbol{x}_n)^\top \phi(\boldsymbol{\mu}_k) \\
&= k(\boldsymbol{x}_n, \boldsymbol{x}_n) + k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k) - 2k(\boldsymbol{x}_n, \boldsymbol{\mu}_k)
\end{aligned}
$$

- Here $k(.,.)$ denotes the kernel function and ϕ is its (implicit) feature map
- Note: ϕ doesn't have to be computed/stored for data $\{\boldsymbol{x}_n\}_{n=1}^N$ or the cluster means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ because computations only depend on kernel evaluations

Kernel K-means vs. K-means

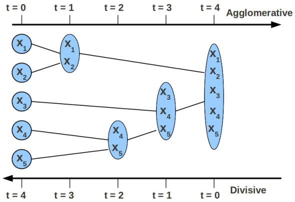


Pyclust: Open Source Data Clustering Pckage

  - A small technical note: When computing $k(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k)$ and $k(\boldsymbol{x}_n, \boldsymbol{\mu}_k)$, remember that $\phi(\boldsymbol{\mu}_k)$ is the average of ϕ's the data points assigned to cluster k

Extra Slides: Hierarchical Clustering

Hierarchical Clustering



- **Agglomerative (bottom-up) Clustering**
  1. Start with each example in its own singleton cluster
  2. At each time-step, greedily merge 2 most similar clusters
  3. Stop when there is a single cluster of all examples, else go to 2
- **Divisive (top-down) Clustering**
  1. Start with all examples in the same cluster
  2. At each time-step, remove the "outsiders" from the least cohesive cluster
  3. Stop when each example is in its own singleton cluster, else go to 2
- Agglomerative is more popular and simpler than divisive (but less accurarate)

(Dis)similarity between clusters

- We know how to compute the dissimilarity $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two examples
- How to compute the dissimilarity between two clusters R and S?
- **Min-link or single-link:** results in chaining (clusters can get very large)

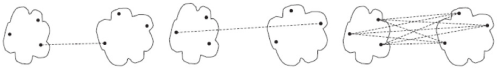
$$d(R,S) = \min_{\boldsymbol{x}_R \in R, \boldsymbol{x}_S \in S} d(\boldsymbol{x}_R, \boldsymbol{x}_S)$$

- **Max-link or complete-link:** results in small, round shaped clusters

$$d(R,S) = \max_{\boldsymbol{x}_R \in R, \boldsymbol{x}_S \in S} d(\boldsymbol{x}_R, \boldsymbol{x}_S)$$

- **Average-link:** compromise between single and complexte linkage

$$d(R,S) = \frac{1}{|R||S|} \sum_{\boldsymbol{x}_R \in R, \boldsymbol{x}_S \in S} d(\boldsymbol{x}_R, \boldsymbol{x}_S)$$



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

Flat vs Hierarchical Clustering

- Flat clustering produces a single partitioning
- Hierarchical Clustering can give different partitionings depending on the level-of-resolution we are looking at
- Flat clustering is usually more efficient run-time wise
- Hierarchical clustering can be slow (has to make several merge/split decisions)