Cluster Analysis (a)

Lijun Zhang
zlj@nju.edu.cn
http://cs.nju.edu.cn/zlj

Outline

- **Introduction**
- Feature Selection for Clustering
- Representative-Based Algorithms
- Hierarchical Clustering Algorithms
- Probabilistic Model-Based Algorithms
- Summary

Introduction

An Informal Definition

Given a set of data points, partition them into groups containing very similar data points.

Applications

- Data summarization
- Customer segmentation
  - ✓ Collaborative filtering
- Social network analysis
  - ✓ Community detection
- Relationship to other mining problems

Outline

- Introduction
- **Feature Selection for Clustering**
- Representative-Based Algorithms
- Hierarchical Clustering Algorithms
- Probabilistic Model-Based Algorithms
- Summary

Feature Selection for Clustering

- The Goal
  - Remove the noisy attributes that do not cluster well
- Unsupervised
  - Determine the inherent clustering tendency of a set of features
- Two Primary Classes of Models
  - Filter models: a score is associated with each feature or a combination
  - Wrapper models: a clustering algorithm is used to evaluate a subset of features

Filter Models—Term Strength (1)

- Suitable for Sparse Domains
  - Text data
- Similar Document Pairs
  - Document pairs with similarity greater than some threshold
- The Definition
  - The fraction of similar document pairs, in which the term occurs in both the documents, conditional on the fact that it appears in the first

Filter Models—Term Strength (2)

- A Probabilistic Definition

$$\text{Term Strength} = P(t \in \overline{Y} | t \in \overline{X}).$$

  - $\bar{X}$ and $\bar{Y}$ are similar documents

- The Procedure
  - Sample document pairs
  - Record T_1, the number of similar document pairs in which t appears in both of them
  - Record T_2, the number of similar document pairs in which t appears in the first of them

$$\text{Term Strength} = \frac{T_1}{T_2}$$

Predictive Attribute Dependence

- Motivation
  - Correlated features result in better clusters
  - Correlated feature can be predicted
- The Approach for Quantifying Relevance
  - Use a classification algorithm on all attributes, except attribute i, to predict the value of attribute i
  - Report the classification accuracy as the relevance of attribute
- Regression can also be Used

Entropy

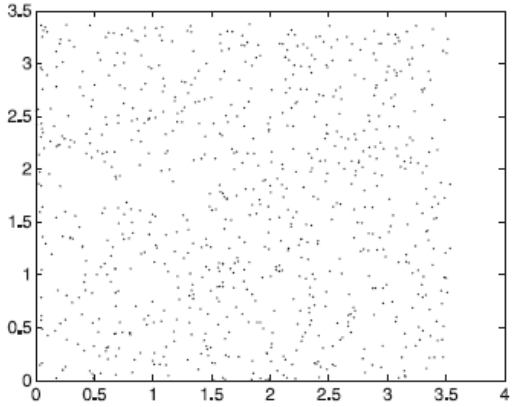
Motivation

- Highly clustered data reflects some of its clustering characteristics on the underlying distance distributions

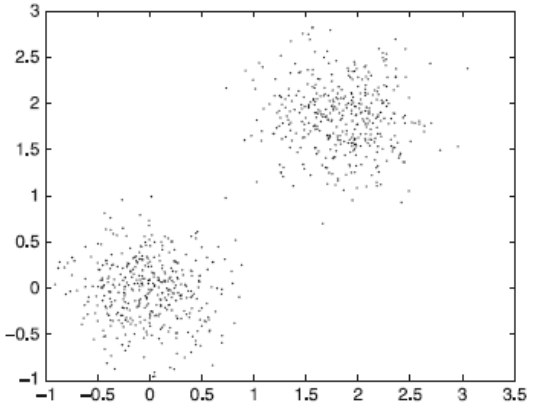
Entropy

Motivation

- H … of its … erlying …

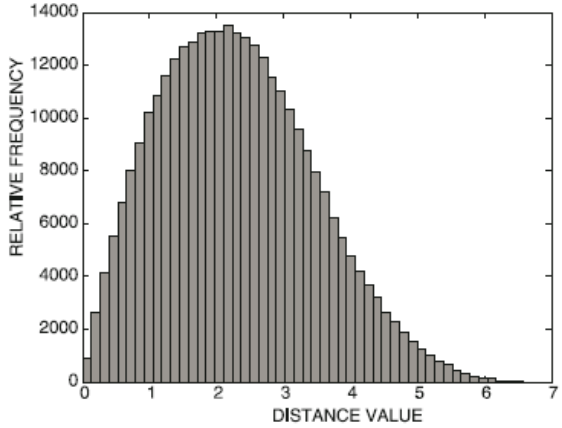


(a) Uniform Data

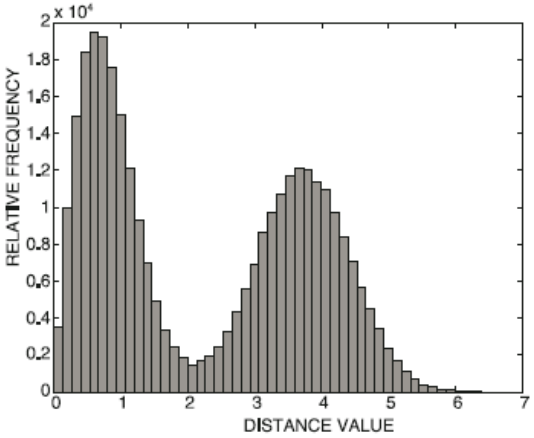


(b) Clustered data

High Entropy ⇨



(c) Distance distribution (uniform)



(d) Distance distribution (clustered)

⇦ **Low Entropy**

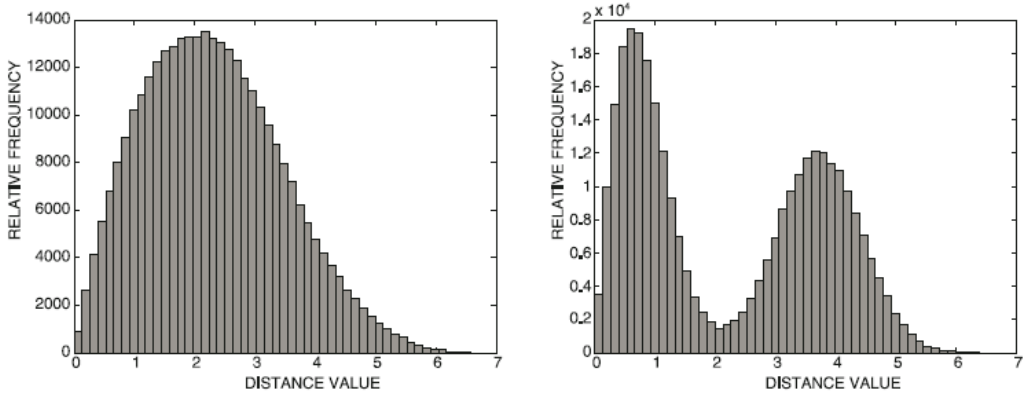
Entropy (1)

- Motivation
  - Highly clustered data reflects some of its clustering characteristics on the underlying distance distribution
- The Key Idea
  - Find subset of features such that the distance distribution has low entropy
- Quantify the Entropy (1st Approach)
  - Discretize the data using ϕ grid regions for each dimension, and obtain $m = \phi^k$ grid

$$E = -\sum_{i=1}^{m} [p_i \log(p_i) + (1 - p_i) \log(1 - p_i)].$$

Entropy (2)

- Quantify the Entropy (1st Approach)
  - If data is sparse, then p_i is inaccurate
  - Hard to fix ϕ^k for different k
- Quantify the Entropy (2nd Approach)
  - Compute the entropy of the 1-dimensional point-to-point distance distribution



$$E = -\sum_{i=1}[p_i \log(p_i) + (1 - p_i)\log(1 - p_i)].$$

Entropy (2)

- Quantify the Entropy (1st Approach)
  - If data is sparse, then p_i is inaccurate
  - Hard to fix ϕ^k for different k
- Quantify the Entropy (2nd Approach)
  - Compute the entropy of the 1-dimensional point-to-point distance distribution
- Find the optimal subset
  - Brute Force Algorithms
  - Greedy Algorithms
    - ✓ Start from the full set of features, and drop the feature that leads to the greatest reduction in the entropy

Hopkins Statistic (1)

Notations

- $\mathcal{D}$ is the data set, whose clustering tendency needs to be evaluated
- $\mathcal{R}$ is a set of r data points from $\mathcal{D}$
- $\alpha_1, \ldots, \alpha_r$ are distances of points in $\mathcal{R}$ to their nearest neighbors in $\mathcal{D}$

- $\mathcal{S}$ is a set of r synthetic data points, which are generated randomly
- $\beta_1, \ldots, \beta_r$ are distances of points in $\mathcal{S}$ to their nearest neighbors in $\mathcal{D}$

Hopkins Statistic (2)

- Definition

$$H = \frac{\sum_{i=1}^{r} \beta_i}{\sum_{i=1}^{r} (\alpha_i + \beta_i)} \in (0,1)$$

  - Uniformly distributed data will have a Hopkins statistic of 0.5
  - Clustered data will result in a value of the Hopkins statistic that is closer to 1
- Random sampling can be repeated
- Can be combined with a greedy algorithm

Wrapper Models (1)

- The Key Idea
  - Use a clustering algorithm with a subset of features
  - Evaluate the quality of this clustering with a cluster validity criterion
- Find the optimal subset
  - Brute Force Algorithms
  - Greedy Algorithms
- Limitation
  - Sensitive to the validity criterion

Wrapper Models (2)

- Another Approach based on Supervised Feature Selection
  - Use a clustering algorithm on the current subset of selected features F, in order to fix cluster labels L for the data points
  - Use any supervised criterion to quantify the quality of the individual features with respect to labels L
    - ✓ Class-based Entropy, Fisher Score
  - Select the top-k features on the basis of this quantification

Outline

- Introduction
- Feature Selection for Clustering
- **Representative-Based Algorithms**
- Hierarchical Clustering Algorithms
- Probabilistic Model-Based Algorithms
- Summary

Partitioning Representatives

- What are Representatives?
  - A function of the data points in the clusters
  - Existing data points in the cluster
- How to use Representatives?
  - Assign data points to their closest representatives
- How to find Representatives?

$$\min_{\overline{Y_1},\ldots,\overline{Y_k}} O = \sum_{i=1}^{n} \left[\min_{j} Dist\left(\overline{X_i}, \overline{Y_j}\right) \right]$$

  - $\overline{X_1}, \ldots, \overline{X_n}$ are data points

Optimization

$$\min_{\overline{Y_1},\ldots,\overline{Y_k}} O = \sum_{i=1}^{n} \left[\min_j Dist(\overline{X_i}, \overline{Y_j}) \right]$$

- If the optimal representatives are known, then the optimal assignment is easy to determine, and vice versa.
- An Iterative Approach

- (Assign step) Assign each data point to its closest representative in S using distance function $Dist(\cdot,\cdot)$, and denote the corresponding clusters by $\mathcal{C}_1 \ldots \mathcal{C}_k$.
- (Optimize step) Determine the optimal representative $\overline{Y_j}$ for each cluster $\mathcal{C}_j$ that minimizes its *local* objective function $\sum_{\overline{X_i} \in \mathcal{C}_j} [Dist(\overline{X_i}, \overline{Y_j})]$.

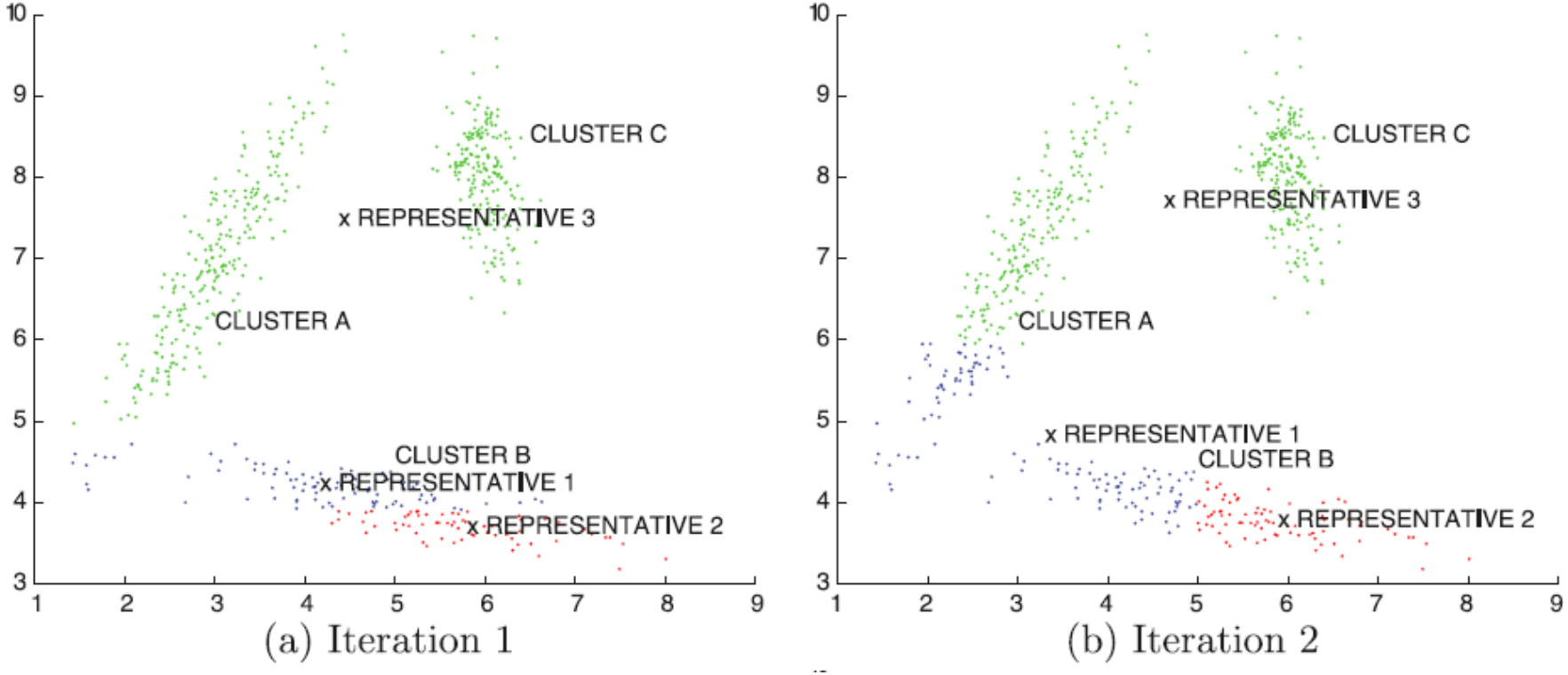
Generic Representative Algorithm

Algorithm *GenericRepresentative*(Database: $\mathcal{D}$, Number of Representatives: k)
begin
  Initialize representative set S;
  repeat
    Create clusters $(\mathcal{C}_1 \ldots \mathcal{C}_k)$ by assigning each
      point in $\mathcal{D}$ to closest representative in S
      using the distance function $Dist(\cdot, \cdot)$;
    Recreate set S by determining one representative $\overline{Y_j}$ for
      each $\mathcal{C}_j$ that minimizes $\sum_{\overline{X_i} \in \mathcal{C}_j} Dist(\overline{X_i}, \overline{Y_j})$;
  until convergence;
  return $(\mathcal{C}_1 \ldots \mathcal{C}_k)$;
end

- Time Complexity per Iteration $O(knd)$
- Local Optimal Solution
  - Repeat multiple times and chooses the one with smallest objective value

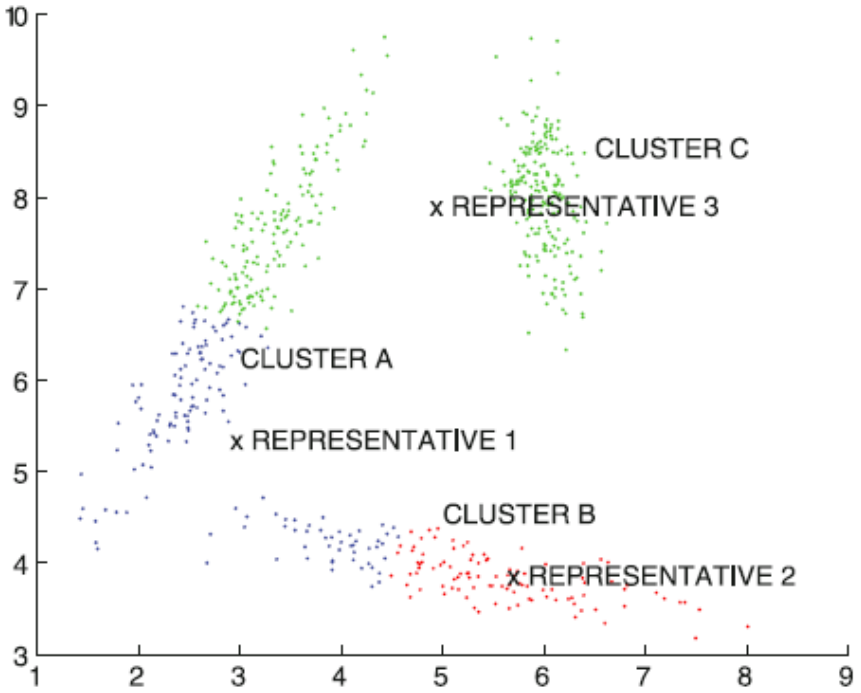
An Example with Euclidean distance function (1)

- A bad initial result

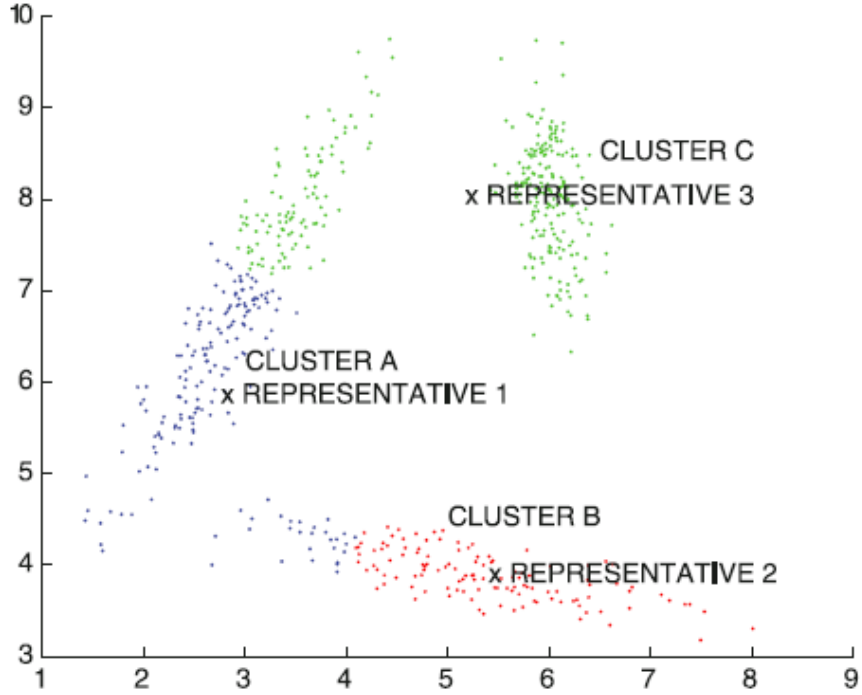


An Example with Euclidean distance function (2)

- Better and better



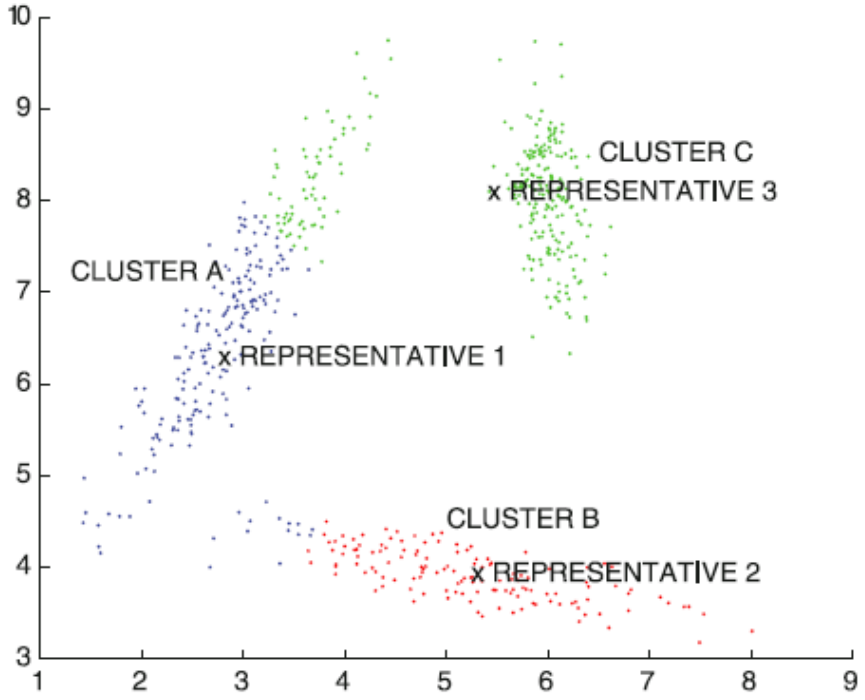
(c) Iteration 3



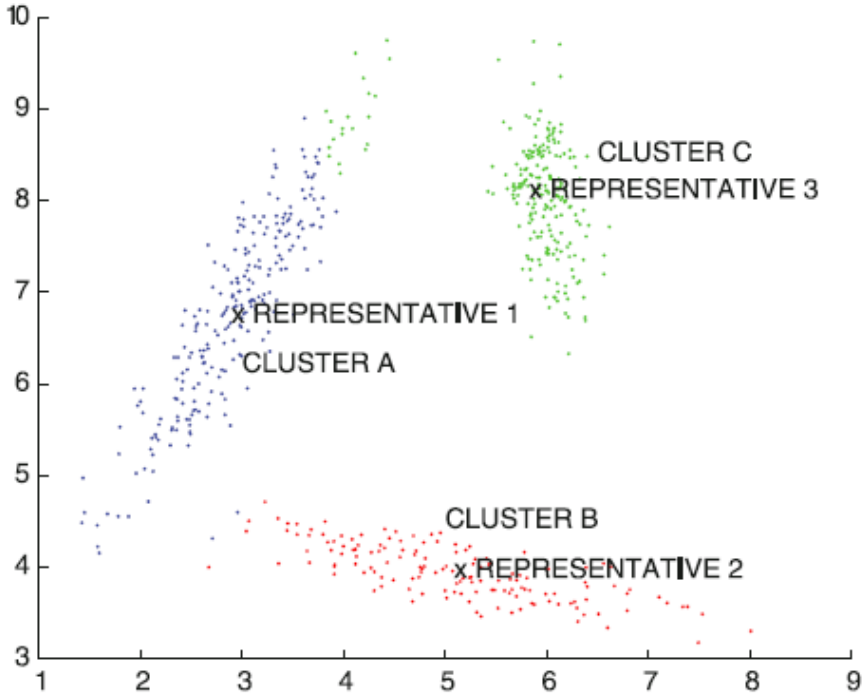
(d) Iteration 4

An Example with Euclidean distance function (3)

- A good result after 10 iterations



(e) Iteration 5



(f) Iteration 10

The k-Means Algorithm

- Optimization with Euclidean distances

$$Dist(\overline{X}_i, \overline{Y}_j) = \left\| \overline{X}_i - \overline{Y}_j \right\|_2^2$$

- Sum of Square Errors

$$\min_{\overline{Y}_1,\ldots,\overline{Y}_k} O = \sum_{i=1}^{n} \left[\min_j \left\| \overline{X}_i - \overline{Y}_j \right\|_2^2 \right]$$

- Assign Step: determine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$

The k-Means Algorithm

- Optimization with Euclidean distances

$$Dist(\overline{X}_i, \overline{Y}_j) = \left\|\overline{X}_i - \overline{Y}_j\right\|_2^2$$

- Sum of Square Errors

$$\min_{\overline{Y}_1,\ldots,\overline{Y}_k} O = \sum_{i=1}^{n}\left[\min_j \left\|\overline{X}_i - \overline{Y}_j\right\|_2^2\right]$$

- Assign Step: determine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$
- Optimize Step

$$\overline{Y}_j = \text{argmin}_{\bar{Y}} \sum_{\overline{X}_i \in \mathcal{C}_j} \|\overline{X}_i - \bar{Y}\|_2^2$$

The k-Means Algorithm

- Optimization with Euclidean distances

$$Dist(\overline{X}_i, \overline{Y}_j) = \left\|\overline{X}_i - \overline{Y}_j\right\|_2^2$$

- Sum of Square Errors

$$\min_{\overline{Y}_1,\ldots,\overline{Y}_k} O = \sum_{i=1}^{n}\left[\min_j \left\|\overline{X}_i - \overline{Y}_j\right\|_2^2\right]$$

- Assign Step: determine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$
- Optimize Step

$$\overline{Y}_j = \text{argmin}_{\bar{Y}} \sum_{\overline{X}_i \in \mathcal{C}_j} \|\overline{X}_i - \bar{Y}\|_2^2 = \frac{1}{|\mathcal{C}_j|}\sum_{\overline{X}_i \in \mathcal{C}_j} \overline{X}_i$$

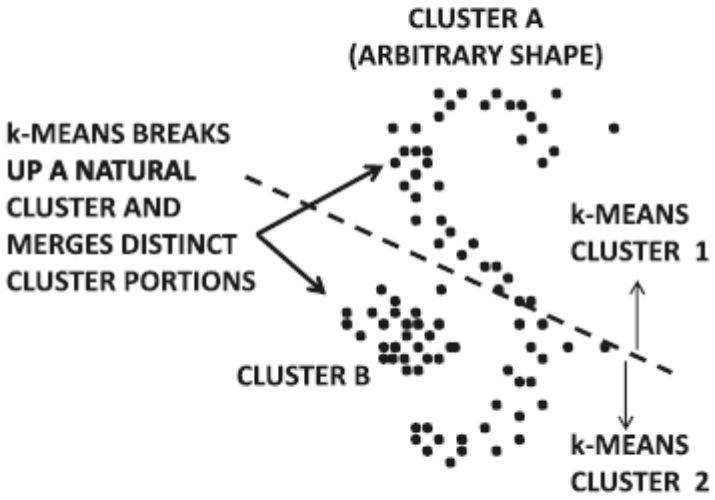
Mahalanobis k-Means Algorithm

- Optimization with Local Mahalanobis Distance

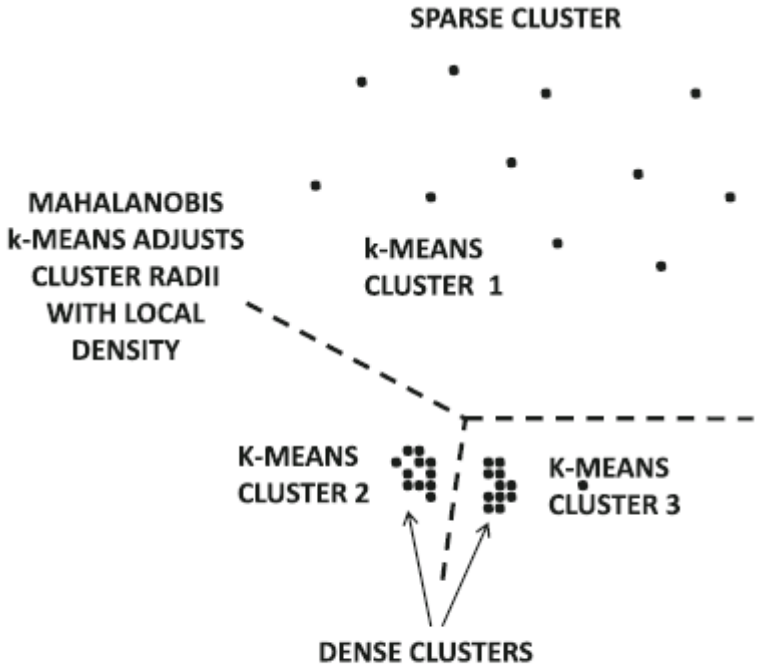
$$Dist(\overline{X}_i, \overline{Y}_j) = (\overline{X}_i - \overline{Y}_j)\Sigma_r^{-1}(\overline{X}_i - \overline{Y}_j)^{\mathsf{T}}$$

  - Σ_r is the $d \times d$ covariance matrix of $\mathcal{C}_r$
  - Σ_r is computed based on data points assigned to $\mathcal{C}_r$ in the previous iteration.
- Assign Step: determine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ based on the new distance
- Optimize Step

Strengths and Weaknesses



(a) Varying cluster shape
(Bad for k-means)



(b) Varying cluster density
(Good for Mahalanobis k-means)

The Kernel k-Means Algorithm

Kernel Trick

- Replace inner product with kernel functions
- The Original Distance

$$||\overline{X}-\overline{\mu}||^2 = ||\overline{X}-\frac{\sum_{\overline{X_i}\in\mathcal{C}}\overline{X_i}}{|\mathcal{C}|}||^2 = \overline{X}\cdot\overline{X}-2\frac{\sum_{\overline{X_i}\in\mathcal{C}}\overline{X}\cdot\overline{X_i}}{|\mathcal{C}|}+\frac{\sum_{\overline{X_i},\overline{X_j}\in\mathcal{C}}\overline{X_i}\cdot\overline{X_j}}{|\mathcal{C}|^2}.$$

- The New Distance

$$\kappa(\bar{X},\bar{X}) - 2\frac{\sum_{\overline{X_i}\in\mathcal{C}}\kappa(\bar{X},\bar{X}_i)}{|\mathcal{C}|}+\frac{\sum_{\overline{X_i},\overline{X_j}\in\mathcal{C}}\kappa\left(\bar{X}_i,\bar{X}_j\right)}{|\mathcal{C}|^2}$$

where $\kappa(\cdot,\cdot)\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a kernel function

An Implicit Mapping

$$\bar{X} \to \phi(\bar{X}) \text{ and } \phi(\bar{X})\cdot\phi(\bar{Y}) = \kappa(\bar{X},\bar{Y})$$

The k-Medians Algorithm

- Optimization with the Manhattan Distances

$$Dist(\overline{X}_i, \overline{Y}_j) = \left\| \overline{X}_i - \overline{Y}_j \right\|_1 = \sum_{p=1}^{d} \left| X_i^p - Y_j^p \right|$$

- Assign Step: determine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$
- Optimize Step

$$Y_j^p = \text{argmin}_Y \sum_{\overline{X}_i \in \mathcal{C}_j} \left| X_i^p - Y \right|$$

The k-Medians Algorithm

- Optimization with the Manhattan Distances

$$Dist(\overline{X}_i, \overline{Y}_j) = \left\|\overline{X}_i - \overline{Y}_j\right\|_1 = \sum_{p=1}^{d} \left|X_i^p - Y_j^p\right|$$

- Assign Step: determine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$
- Optimize Step

$$Y_j^p = \text{argmin}_Y \sum_{\overline{X}_i \in \mathcal{C}_j} \left|X_i^p - Y\right| = \text{median}\left\{X_i^p \middle| \overline{X}_i \in \mathcal{C}_j\right\}$$

  - $\bar{Y} = \left[Y_j^1, \ldots, Y_j^d\right]$ may not belong to $\mathcal{D}$

The k-Medoids Algorithm (1)

- Representatives are Selected from $\mathcal{D}$

$$\min_{\overline{Y_1},\ldots,\overline{Y_k}\in\mathcal{D}} O = \sum_{i=1}^{n}\left[\min_{j} Dist\left(\overline{X_i},\overline{Y_j}\right)\right]$$

- Why $\overline{Y_1},\ldots,\overline{Y_k}\in\mathcal{D}$?
  - The representative of a k-means cluster may be distorted by outliers
  - k-means can not be applied to heterogeneous data
  - Good for summarization

The k-Medoids Algorithm (2)

- Optimization based on Hill-climbing
  - The representative set $\mathcal{S}$ is initialized to a set of points from $\mathcal{D}$
  - $\mathcal{S}$ is iteratively improved by exchanging a single point from $\mathcal{S}$ with a point from $\mathcal{D}$
- How to perform the exchange?
  - Try all $|\mathcal{S}| \cdot |\mathcal{D}|$ possible exchanges
  - Try a randomly select set of r pairs $(\overline{X}_i, \overline{Y}_j)$ and select the best one

Practical and Implementation Issues

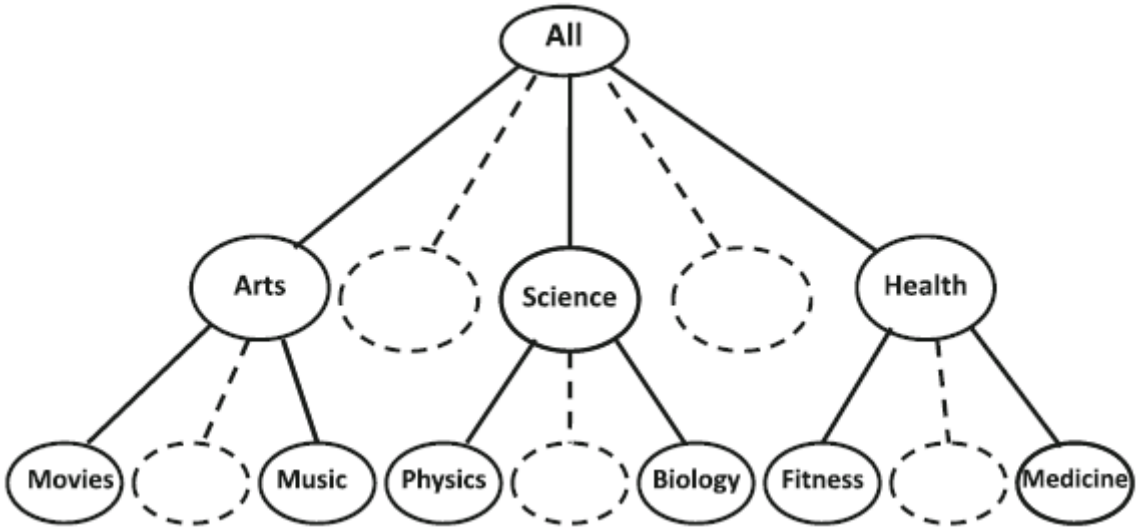
- The initialization criteria
  - Select points randomly from the data space or from the data set $\mathcal{D}$
  - Sample more data points from $\mathcal{D}$, and use a hierarchical clustering approach to create k centroids
- The choice of k
  - In practice, it is better to use large k first, and then post-process
- The presence of outlier
  - Discard centers with small clusters

Outline

- Introduction
- Feature Selection for Clustering
- Representative-Based Algorithms
- **Hierarchical Clustering Algorithms**
- Probabilistic Model-Based Algorithms
- Summary

Hierarchical Clustering

Taxonomy of Clusters



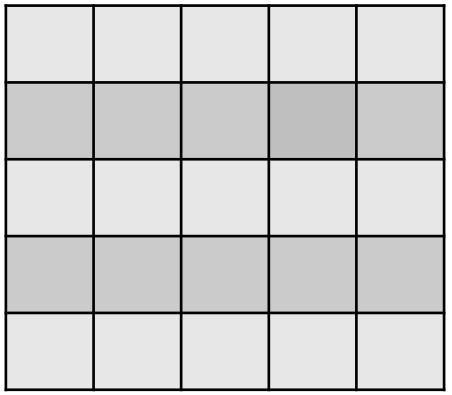
- Different levels of clustering granularity provide different application-specific insights

Types of Hierarchical Algorithms

- Bottom-up (agglomerative) methods
  - The individual data points are successively agglomerated into higher-level clusters
- Top-down (divisive) methods
  - Successively partition the data points into a tree-like structure
  - Flexible in in terms of choosing the trade-off between the balance in the tree structure and the balance in the number of data points in each node

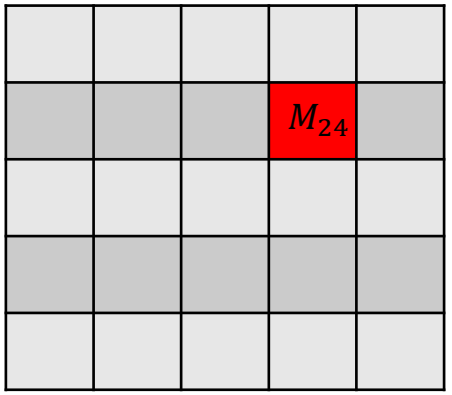
Bottom-Up Agglomerative Methods

- The Procedure in the t-th iteration
  - A distance matrix M between n_t clusters
    - ✓ It is symmetric



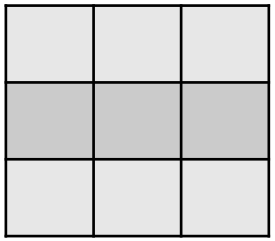
Bottom-Up Agglomerative Methods

- The Procedure in the t-th iteration
  - A distance matrix M between n_t clusters
    - ✓ It is symmetric
  - Find the smallest entry M_{ij}



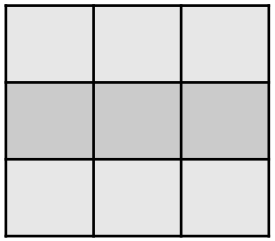
Bottom-Up Agglomerative Methods

- The Procedure in the t-th iteration
  - A distance matrix M between n_t clusters
    - ✓ It is symmetric
  - Find the smallest entry M_{ij}
  - Delete rows and columns i, j



Bottom-Up Agglomerative Methods

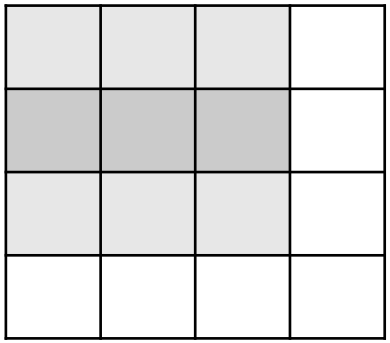
- The Procedure in the t-th iteration
  - A distance matrix M between n_t clusters
    - ✓ It is symmetric
  - Find the smallest entry M_{ij}
  - Delete rows and columns i, j
  - Merge clusters C_i and C_j



Bottom-Up Agglomerative Methods

The Procedure in the t-th iteration

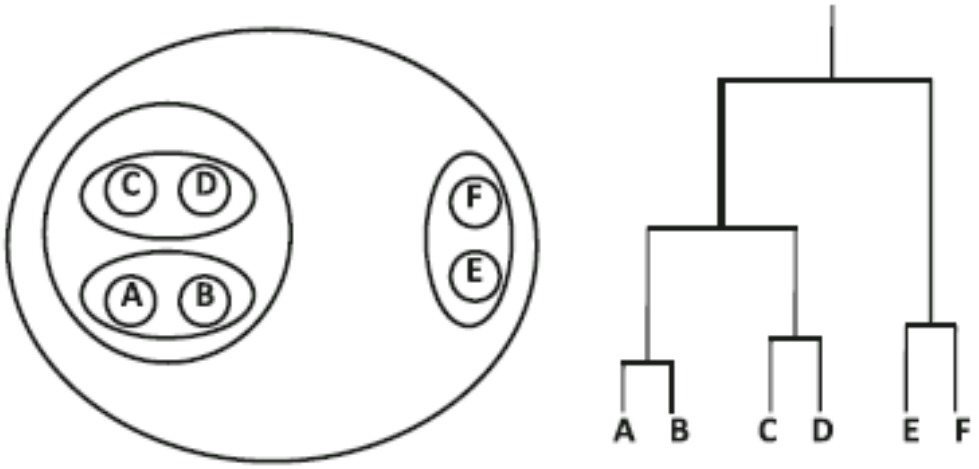
- A distance matrix M between n_t clusters
  - ✓ It is symmetric
- Find the smallest entry M_{ij}
- Delete rows and columns i, j
- Merge clusters C_i and C_j



- Add a new row and column in M
- Set the values in the new row and column
  - ✓ Sometimes, the value can be obtained from the deleted rows and columns

Dendrogram

- The order of merging naturally creates a hierarchical tree-like structure



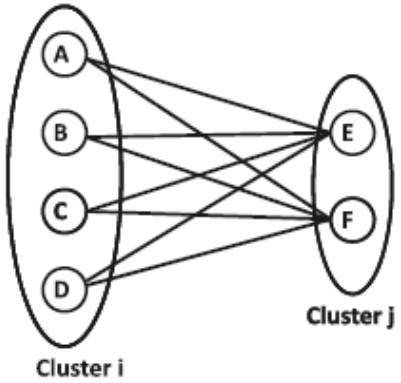
(a) Dendrogram

Generic Agglomerative Merging Algorithm

Algorithm *AgglomerativeMerge*(Data: $\mathcal{D}$)
begin
  Initialize $n \times n$ distance matrix M using $\mathcal{D}$;
  repeat
    Pick closest pair of clusters i and j using M;
    Merge clusters i and j;
    Delete rows/columns i and j from M and create
      a new row and column for newly merged cluster;
    Update the entries of new row and column of M;
  until termination criterion;
  return current merged cluster set;
end

Distance between Clusters

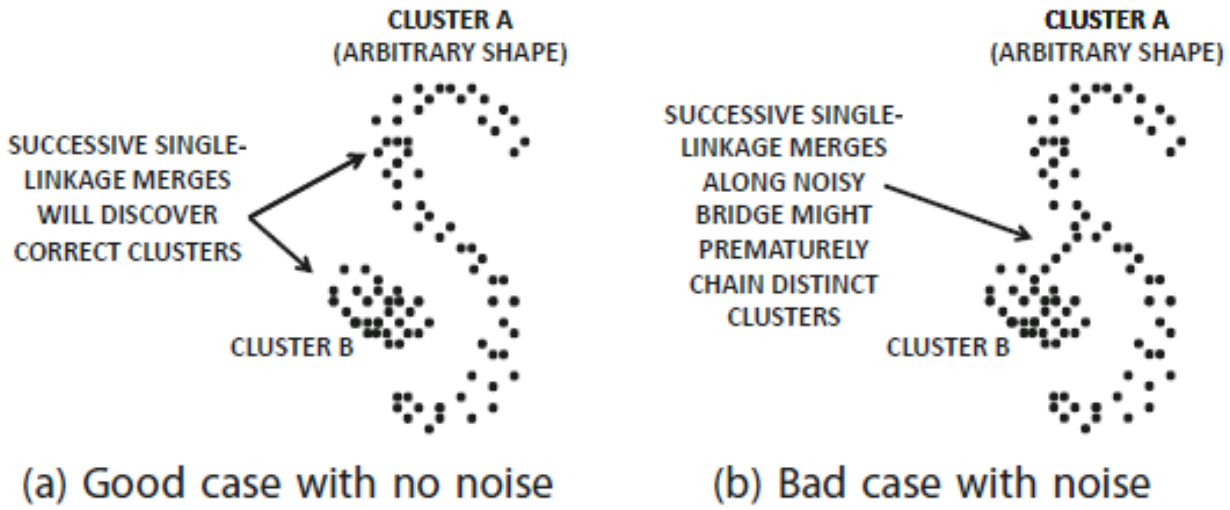
- Distances between Elements in Clusters $\mathcal{C}_i$ and $\mathcal{C}_j$
  - $|\mathcal{C}_i| \cdot |\mathcal{C}_j|$ pairs of distances



- Distances between Clusters $\mathcal{C}_i$ and $\mathcal{C}_j$
  - As a function of those $|\mathcal{C}_i| \cdot |\mathcal{C}_j|$ pairs

Group-Based Statistics

- Best (single) Linkage
  - Minimum distance between $|\mathcal{C}_i| \cdot |\mathcal{C}_j|$ pairs



(a) Good case with no noise (b) Bad case with noise

Group-Based Statistics

- Best (single) Linkage
  - Minimum distance between $|\mathcal{C}_i| \cdot |\mathcal{C}_j|$ pairs
- Worst (complete) Linkage
  - Maximum distance between $|\mathcal{C}_i| \cdot |\mathcal{C}_j|$ pairs
- Group-average linkage
  - Average distance between $|\mathcal{C}_i| \cdot |\mathcal{C}_j|$ pairs
- Closest Centroid
  - Distance between centroids

Practical Considerations

- Difficult to control the structure of the hierarchical tree
- Sensitive to mistakes made during the merging process
  - There is no way to undo it
- High computational cost
  - Space complexity: $O(n^2)$
  - Time complexity: $O(n^2 d + n^2 \log n)$

Top-Down Divisive Methods

The Algorithm

Algorithm *GenericTopDownClustering*(Data: $\mathcal{D}$, Flat Algorithm: $\mathcal{A}$)
begin
  Initialize tree $\mathcal{T}$ to root containing $\mathcal{D}$;
  repeat
    Select a leaf node L in $\mathcal{T}$ based on pre-defined criterion;
    Use algorithm $\mathcal{A}$ to split L into $L_1 \ldots L_k$;
    Add $L_1 \ldots L_k$ as children of L in $\mathcal{T}$;
  until termination criterion;
end

- $\mathcal{A}$ can be any clustering algorithm
- Many possible criteria for node selection
  - ✓ Size, depth

Outline

- Introduction
- Feature Selection for Clustering
- Representative-Based Algorithms
- Hierarchical Clustering Algorithms
- **Probabilistic Model-Based Algorithms**
- Summary

Two Types of Clustering

- Hard Clustering
  - Each data point is deterministically assigned to a particular cluster
- Soft Clustering
  - Each data point may have a nonzero assignment probability to many (typically all) clusters

Mixture-based Generative Model

- Data was generated from a mixture of k distributions with probability distribution $\mathcal{G}_1, \ldots, \mathcal{G}_k$
- $\mathcal{G}_i$ represents a cluster/mixture component
- Each point $\bar{X}$ is generated as follows
  - Select a mixture component with probability $\alpha_i = P(\mathcal{G}_i),\ i = 1, \ldots, k$
  - Assume the r-th component is selected
  - Generate a data point from G_r

The Clustering Process

- Learning: determine $\alpha_1, \ldots, \alpha_k$ and parameters of distributions $\mathcal{G}_1, \ldots, \mathcal{G}_k$ from the observed data
  - Denote all the parameters by Θ

- Testing: decide the probability of $\bar{X}$ belong to cluster $\mathcal{G}_i$

$$P(\mathcal{G}_i|\bar{X},\Theta) = \frac{P(\mathcal{G}_i,\bar{X}|\Theta)}{P(\bar{X}|\Theta)} = \frac{P(\mathcal{G}_i,\bar{X}|\Theta)}{\sum_{r=1}^{k} P(\mathcal{G}_r,\bar{X}|\Theta)}$$

$$P(\mathcal{G}_i,\bar{X}|\Theta) = P(\mathcal{G}_i)P(\bar{X}|\mathcal{G}_i,\Theta) = \alpha_i P(\bar{X}|\mathcal{G}_i,\Theta)$$

The Objective of Learning (1)

- Denote the probability density function of $\mathcal{G}_i$ by f^i
- The probability that $\overline{X_j}$ generated by the mixture model $\mathcal{M}$ is given by

$$f^{point}(\overline{X_j}|\mathcal{M}) = \sum_{i=1}^{k} P(\mathcal{G}_i, \overline{X_j}) = \sum_{i=1}^{k} P(\mathcal{G}_i)P(\overline{X_j}|\mathcal{G}_i) = \sum_{i=1}^{k} \alpha_i \cdot f^i(\overline{X_j})$$

- The probability of the data set $\mathcal{D} = \{\overline{X_1}, \ldots, \overline{X_n}\}$ generated by $\mathcal{M}$

$$f^{data}(\mathcal{D}|\mathcal{M}) = \prod_{j=1}^{n} f^{point}(\overline{X_j}|\mathcal{M}).$$

The Objective of Learning (2)

Log-likelihood

$$\mathcal{L}(\mathcal{D}|\mathcal{M}) = \log(\prod_{j=1}^{n} f^{point}(\overline{X_j}|\mathcal{M})) = \sum_{j=1}^{n} \log(\sum_{i=1}^{k} \alpha_i f^i(\overline{X_j}))$$

The Optimization Problem

$$\max_{\mathcal{M}} \mathcal{L}(\mathcal{D}|\mathcal{M})$$

- Let Θ be the parameters of $\mathcal{M}$

$$\max_{\Theta} \mathcal{L}(\mathcal{D}|\Theta)$$

Expectation-maximization (EM) Algorithm

- Observation
  - If the soft assignments $P(\mathcal{G}_i|\bar{X}_j, \Theta)$ is known, then it is easy to estimate Θ
  - Similar to the representative-based algorithms
- The Algorithm
  - E-step: use the current Θ to estimate the posterior probability $P(\mathcal{G}_i|\bar{X}_j, \Theta)$
  - M-step: fix the posterior probability, and find Θ to maximize the log-likelihood

An Example for Gaussian Mixture Model

E-step

$$P(\mathcal{G}_i|\overline{X_j},\Theta) = \frac{P(\mathcal{G}_i)\cdot P(\overline{X_j}|\mathcal{G}_i,\Theta)}{\sum_{r=1}^{k} P(\mathcal{G}_r)\cdot P(\overline{X_j}|\mathcal{G}_r,\Theta)} = \frac{\alpha_i \cdot f^{i,\Theta}(\overline{X_j})}{\sum_{r=1}^{k} \alpha_r \cdot f^{r,\Theta}(\overline{X_j})}$$

$$f^{i,\Theta}(\overline{X_j}) = \frac{1}{\sqrt{|\Sigma_i|}(2\cdot\pi)^{(d/2)}} e^{-\frac{1}{2}(\overline{X_j}-\overline{\mu_i})\Sigma_i^{-1}(\overline{X_j}-\overline{\mu_i})}.$$

M-step

$$\alpha_i = P(\mathcal{G}_i) = \frac{\sum_{j=1}^{n} P(\mathcal{G}_i|\overline{X_j},\Theta)}{n}$$

$$\overline{\mu_i} = \frac{1}{\sum_{j=1}^{n} P(\mathcal{G}_i|\overline{X_j},\Theta)} \sum_{j=1}^{n} P(\mathcal{G}_i|\overline{X_j},\Theta)\overline{X_j}$$

$$\Sigma_i = \frac{1}{\sum_{j=1}^{n} P(\mathcal{G}_i|\overline{X_j},\Theta)} \sum_{j=1}^{n} P(\mathcal{G}_i|\overline{X_j},\Theta)\left(\overline{X_j}-\overline{\mu_i}\right)\left(\overline{X_j}-\overline{\mu_i}\right)^{\top}$$

Relation of EM to k-Means

A Simple Mixture Models

- Fix $\alpha_1 = \cdots = \alpha_k = 1/k$
- Choose a simple Gaussian distribution

$$f^{j,\Theta}(\overline{X_i}) = \frac{1}{(\sigma\sqrt{2\cdot\pi})^d} e^{-\left(\frac{||\overline{X_i}-\overline{Y_j}||^2}{2\sigma^2}\right)}$$

Comparisons

1. (E-step) Each data point i has a probability belonging to cluster j, which is proportional to the scaled and exponentiated Euclidean distance to each representative $\overline{Y_j}$. In the k-means algorithm, this is done in a hard way, by picking the *best* Euclidean distance to any representative $\overline{Y_j}$.

2. (M-step) The center $\overline{Y_j}$ is the weighted mean over all the data points where the weight is defined by the probability of assignment to cluster j. The hard version of this is used in k-means, where each data point is either assigned to a cluster or not assigned to a cluster (i.e., analogous to 0-1 probabilities).

Problems of Mixture Models

- Overfitting
  - Too many parameters in Θ
  - A small data set $\mathcal{D}$
  - Reduce the complexity of model
- Local Optimal Solution
  - Repeat many times, and choose the one with smallest objective value

Outline

- Introduction
- Feature Selection for Clustering
- Representative-Based Algorithms
- Hierarchical Clustering Algorithms
- Probabilistic Model-Based Algorithms
- **Summary**

Summary

- Feature Selection for Clustering
  - Filter Models, Wrapper Models
- Representative-Based Algorithms
  - k-Means, k-Medians, k-Medoids
- Hierarchical Clustering Algorithms
  - Bottom-Up Agglomerative Methods
    - ✓ Group-Based Statistics
  - Top-Down Divisive Methods
- Probabilistic Model-Based Algorithms
  - Mixture Model, EM Algorithm