Pattern Recognition

Unsupervised Learning and Clustering

Supervised and Unsupervised Learning

- In ***supervised learning*** a set of correctly labeled training samples for training the classifier.

- The procedures using labeled training samples are part of supervised classification.

- Classification using unlabeled samples is called ***unsupervised classification*** or ***clustering***.

Clustering

- Clustering is either useful or necessary is the following cases.
  - The collection and classification of training data can be costly and time consuming. Therefore, it can be impossible to collect a training set. Also, when there are very many training samples, it can be that all of these cannot be hand-labeled.
  - For data mining, it can be useful to search for natural clusters/groupings among the data, and then recognize the clusters.
  - The properties of feature vectors can change over time. Then, supervised classification is not reasonable, because sooner or later the test feature vectors would have completely different properties than the training data had. (Medical images for example)
  - The clustering can be useful when searching for good parametric families for the class conditional densities for the supervised classification.

Applications

- In grouping of shopping items clustering can be used to group all the shopping items available on the web into a set of unique products.
- For example, all the similar items can be grouped into unique products.

Applications

- Social network analysis In the study of social networks, clustering may be used to recognize communities within large groups of people.
- In the process of intelligent grouping of the files and websites, clustering may be used to create a more relevant set of search results compared to normal search engines like Google.

Applications

- In image segmentation, clustering can be used to divide a digital image into distinct regions for border detection or object recognition.
- Recommender systems are designed to recommend new items based on a user's tastes. They sometimes use clustering algorithms to predict a user's preferences based on the preferences of other users in the user's cluster.

Applications

- Biology
  - In plant and animal ecology cluster analysis is used to describe and to make spatial and temporal comparisons of communities (assemblages) of organisms in heterogeneous environments.
  - In human genetic clustering, the similarity of genetic data is used in clustering to infer population structures.

Clustering Problem

- Given a set of feature vectors D = { x_1 , . . , x_n }, the task is to place each of these feature vectors into one of the c classes.
- Find c sets D_1 , . . . , D_c so that

$$\bigcup_{i=1}^{c} \mathcal{D}_i = \mathcal{D} \qquad \text{and} \qquad \mathcal{D}_j \cap \mathcal{D}_i = \emptyset$$

for all i ≠ j.

Clustering Problem

- Besides, the task is to maximize the similarity of feature vectors within a class.

- For this, we need to define how similar are the feature vectors belonging to some set D_i.

Clustering Methods

- K-Means Clustering
- Fuzzy C-Means Clustering
- Hierarchical Clustering
- Mixture of Gaussian Clustering

K-Means Clustering

- K-Means uses minimum distance classifier to assign a test point to the class with the nearest mean to the test point.
- Similarity within a class D_i is defined as:

$$s(\mathcal{D}_i) = -\sum_{\mathbf{x}\in\mathcal{D}_i} ||\mathbf{x} - \mu_i||^2,$$

$$\mu_i = \frac{1}{n_i}\sum_{\mathbf{x}\in\mathcal{D}_i} \mathbf{x}.$$

K-Means Clustering

- K-Means finds the unknown mean vectors by maximizing the sum of similarities of all classes.
- Equivalently, we can minimize the negative of the sum of similarities as:

$$J(\mathcal{D}_1, \ldots, \mathcal{D}_c) = \sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathcal{D}_i} ||\mathbf{x} - \mu_i||^2.$$

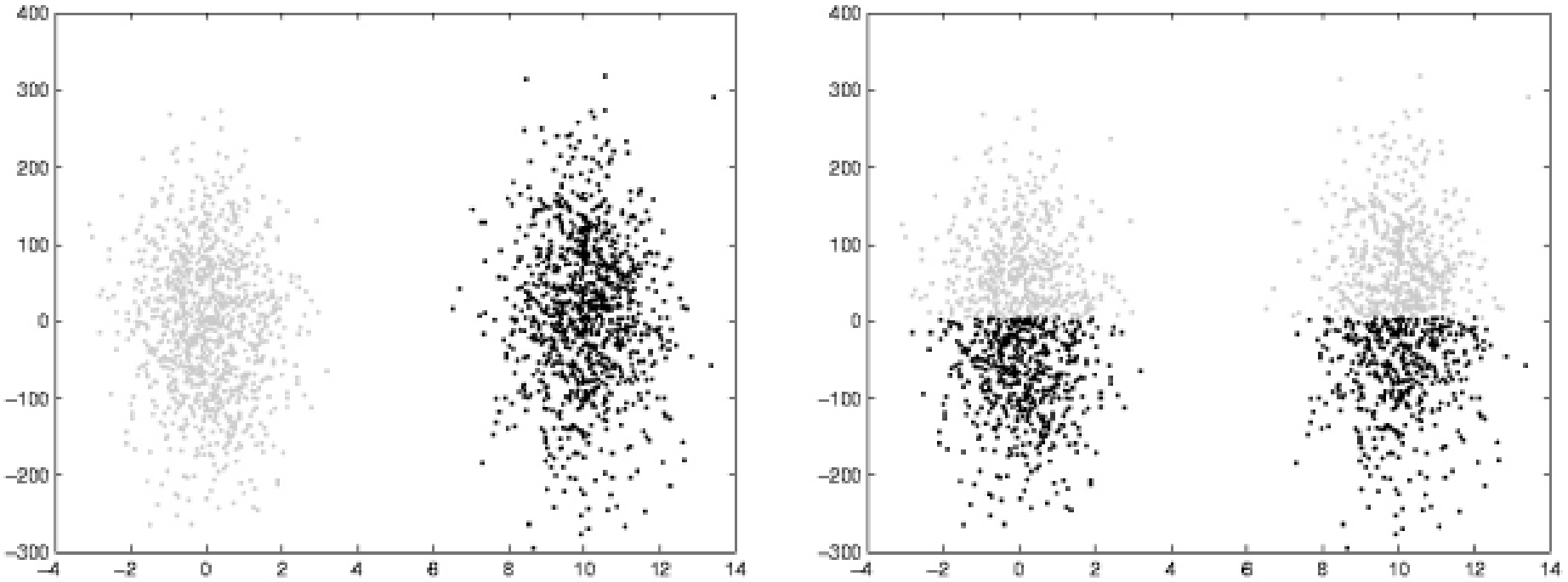
K-Means Algorithm

k-means algorithm
Initialize $\mu_1(0), \ldots, \mu_c(0)$, set $t \leftarrow 0$
repeat
  Classify each $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to the class $\mathcal{D}_j(t)$ whose mean vector $\mu_j(t)$ is the nearest to $\mathbf{x}_i$.
  for $k = 1$ to c **do**
    update the mean vectors $\mu_k(t+1) = \frac{1}{|\mathcal{D}_k(t)|} \sum_{\mathbf{x} \in \mathcal{D}_k(t)} \mathbf{x}$
  end for
  Set $t \leftarrow t + 1$
until clustering did not change
Return $\mathcal{D}_1(t-1), \ldots, \mathcal{D}_c(t-1)$.

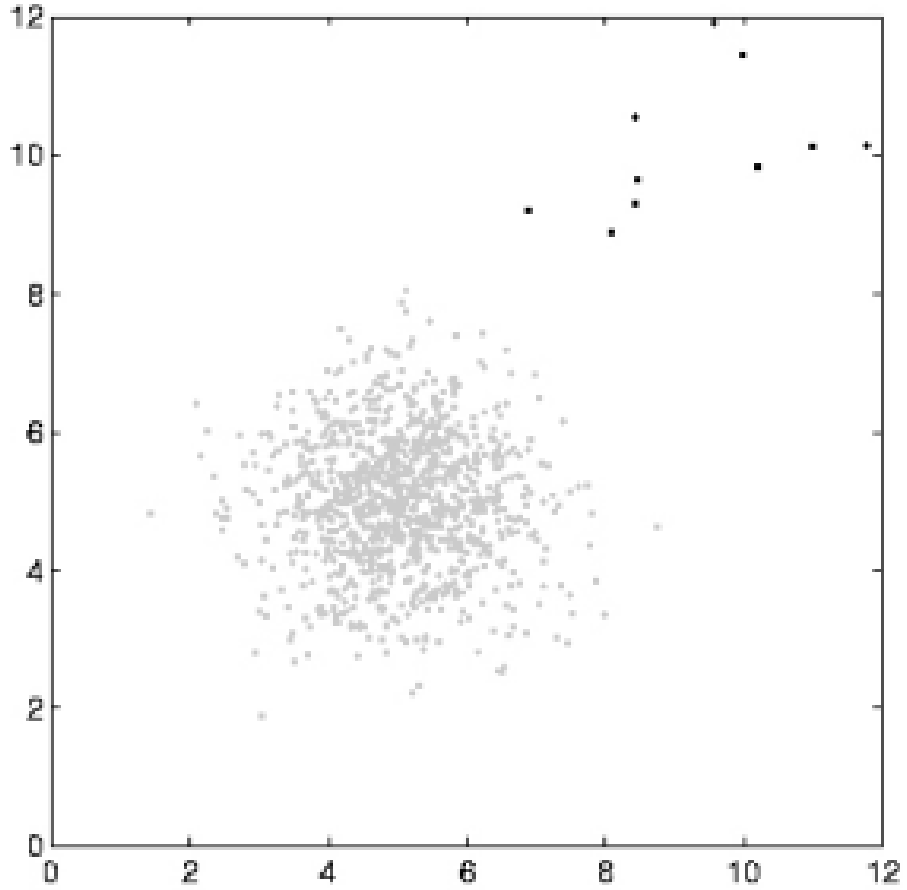
Properties of the K-means Clustering

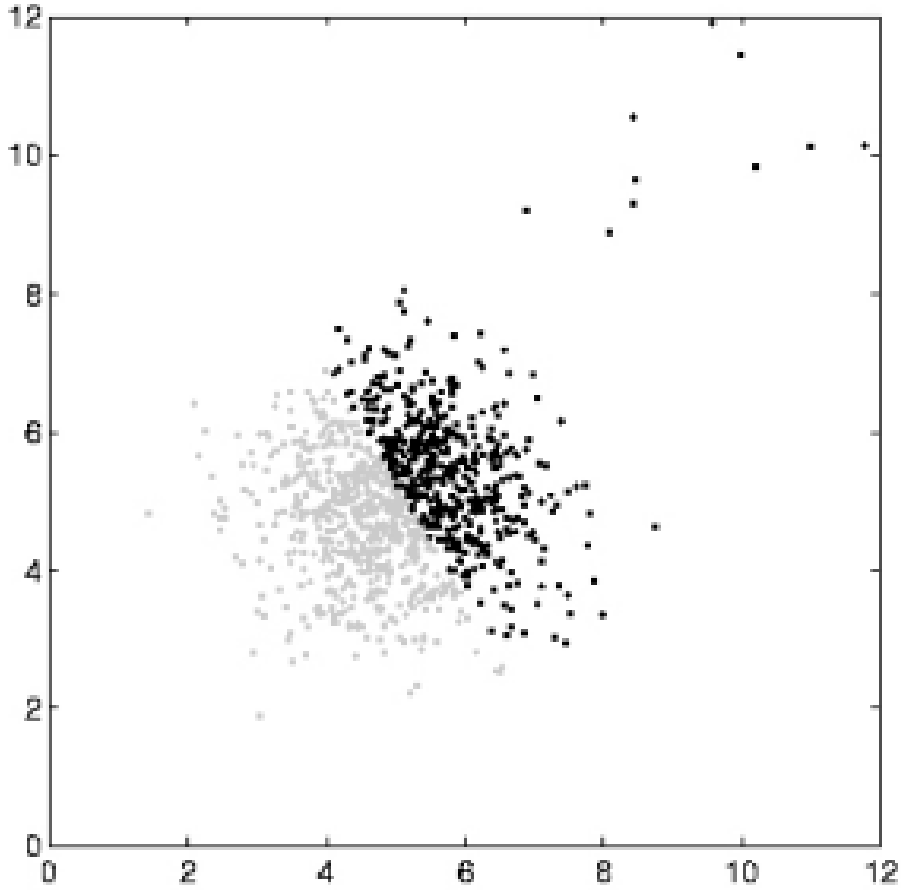
- K-Means is not successful in the following cases:
  - If the clusters are of different sizes (i.e. They contain a different number of samples).
  - If the clusters have very different covariances or scales of different features are different.

Properties of the K-means Clustering



Properties of the K-means Clustering





K-Means Clustering

- The way to initialize the means is not specified. One popular way to start is to randomly choose k of the samples.
- The results produced depend on the initial values for the means, and it frequently happens that suboptimal partitions are found. The standard solution is to try a number of different starting points.
- It can happen that the set of samples closest to $\mathbf{m}_i$ is empty, so that $\mathbf{m}_i$ cannot be updated.
- The results depend on the metric used to measure $||\mathbf{x} - \mathbf{m}_i||$. A popular solution is to normalize each variable by its standard deviation, though this is not always desirable.
- The results depend on the value of k.

Fuzzy C-Means

- Fuzzy c-means (FCM) is a method of clustering which allows one sample to belong to two or more clusters.
- The objective function to minimize is:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2$$

m is any real number greater than 1

U_{ij} is the membership value of x_i in cluster j

Fuzzy C-Means Algorithm

1. *Initialize $U=[u_{ij}]$ matrix, $U^{(0)}$*

2. *Calculate the centers vectors $C^{(k)}=[c_j]$ with $U^{(k)}$*

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

3. *Update $U^{(k)}$, $U^{(k+1)}$*

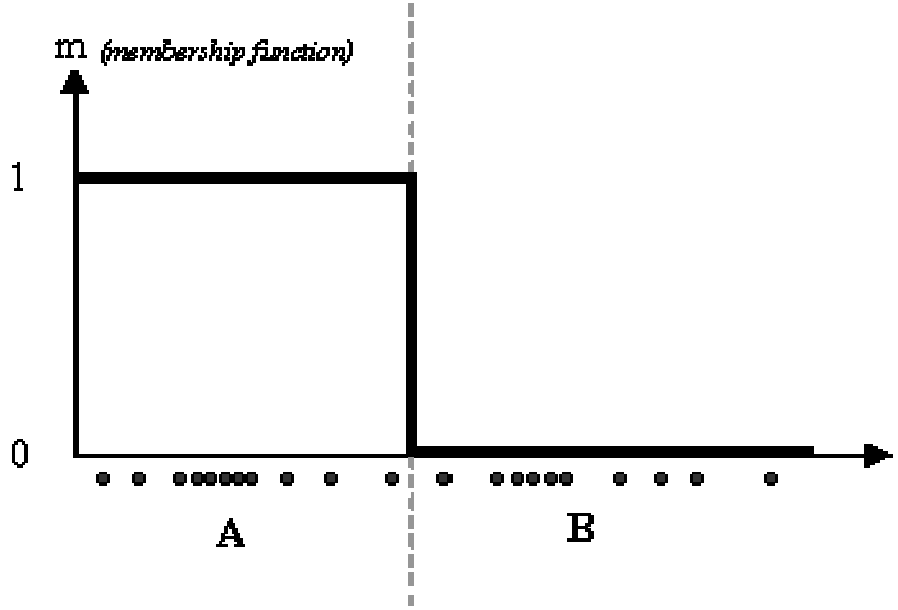
$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

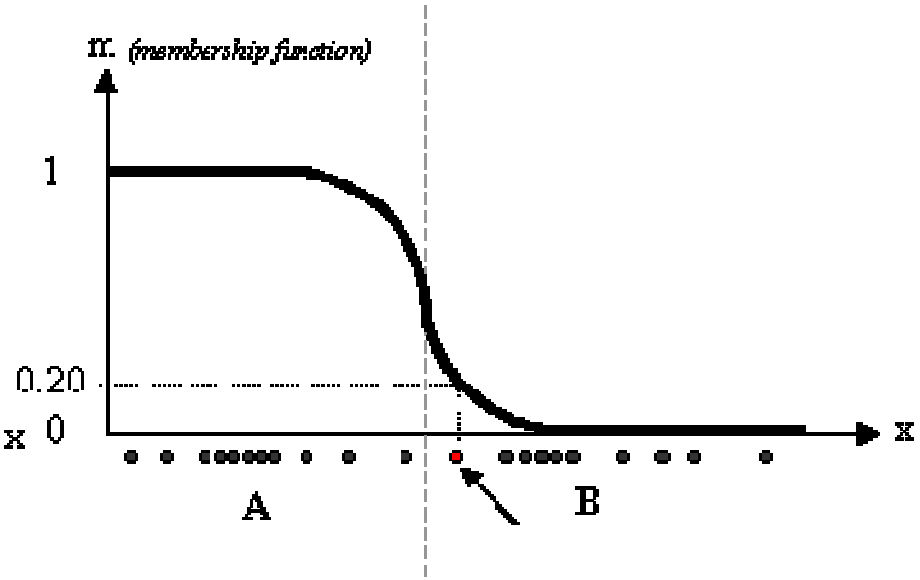
4. *If $|| U^{(k+1)} - U^{(k)} || < \epsilon$ then STOP; otherwise return to step 2.*

Fuzzy memberships

- The samples are bound to each cluster by means of a membership value, which represents the fuzzy behavior of the algorithm.
- The membership values are numbers between 0 and 1, and represent the degree of similarity between data and centers of clusters.

Fuzzy Memberships Example





Hierarchical Clustering

- Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is:
  1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
  2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
  3. Compute distances (similarities) between the new cluster and each of the old clusters.
  4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Measuring Similarity between Clusters

- Similarity (or dissimilarity) between clusters can be measured by one of the following methods:
  - *single-linkage*
  - *complete-linkage*
  - *average-linkage*

Single Linkage

- In *single-linkage* clustering (also called the connectedness or minimum method), we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster.

Complete Linkage

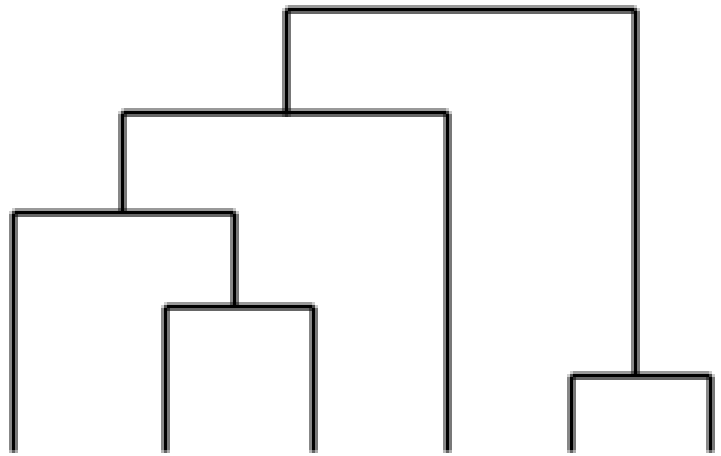
- In *complete-linkage* clustering (also called the *diameter* or *maximum* method), we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster.

Average Linkage

- In *average-linkage* clustering, we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.

Determining the Number of Clusters

- The hierarchical algorithm stops when we have only one cluster.
- Going back we may obtain as many clusters as we wish!



Mixture of Gaussian Clustering

- The task is now to learn the Bayes classifier in the unsupervised manner (without any training samples) when the parametric families of the class conditional densities are known.

Assumptions

- The class conditional densities are modeled by normal densities, i.e. $p(x \mid \omega_j) = p_{normal}(x \mid \mu_J, \Sigma_j)$ for all $j = 1, \ldots, c$.
- Parameters $\mu_1, \ldots, \mu_c, \Sigma_1, \ldots, \Sigma_c$ are not known.
- Priors $P(\omega_1), \ldots, P(\omega_c)$ are not known.
- There are no training samples: i.e. the classes of feature vectors $x_1, \ldots, x_n$ are not known, and the task is to classify the feature vectors.

Mixture of Gaussians

- The parametric mixture density is given as:

$$p(\mathbf{x}_i|\theta) = \sum_{j=1}^{c} p_{normal}(\mathbf{x}_i|\mu_j, \Sigma_j) P(\omega_j),$$

where

$$\theta = (\mu_1, \ldots, \mu_c, \Sigma_1, \ldots, \Sigma_c, P(\omega_1), \ldots, P(\omega_c))^T$$

Questions?