Unsupervised Learning and Clustering

Selim Aksoy

Department of Computer Engineering
Bilkent University
saksoy@cs.bilkent.edu.tr

GE 461, Spring 2023

Introduction

- There are many learning scenarios for modeling data.
- Supervised learning: a teacher provides a category label for each pattern in the training set.
- Unsupervised learning: the system forms clusters (groupings) of the input patterns.
- Reinforcement learning: no desired category is given but the teacher provides feedback to the system such as the decision is right or wrong.
- Other scenarios

Introduction

- Unsupervised procedures are useful for several reasons:
  - Collecting and labeling a large set of sample patterns can be costly or may not be feasible.
  - One can train with large amount of unlabeled data, and then use supervision to label the groupings found.
  - Unsupervised methods can be used for feature extraction.
  - Exploratory data analysis can provide insight into the nature or structure of the data.

Data Description

- Assume that we have a set of unlabeled multi-dimensional patterns.
- One way of describing this set of patterns is to compute their sample mean and covariance.
- This description uses the assumption that the patterns form a cloud that can be modeled with a hyperellipsoidal shape.
- However, we must be careful about any assumptions we make about the structure of the data.

Data Description

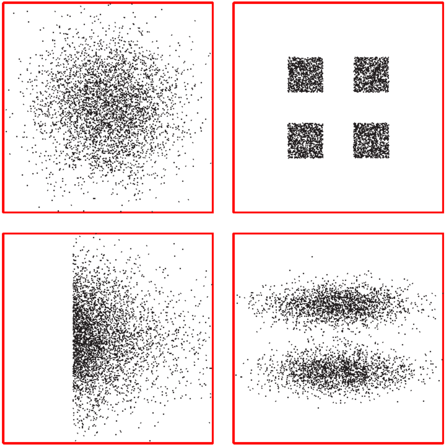


Figure 1: These four data sets have identical first-order and second-order statistics. We need to find other ways of modeling their structure. Clustering is an alternative way of describing the data in terms of groups of patterns.

Clusters

- A *cluster* is comprised of a number of similar objects collected or grouped together.
- *Cluster analysis* organizes data by abstracting the underlying structure either as a grouping of individuals or as a hierarchy of groups.
- Clustering is unsupervised. Category labels and other information about the source of data influence the interpretation of the clusters, not their formation.

Clustering

- Clustering is a very difficult problem because data can reveal clusters with different shapes and sizes.

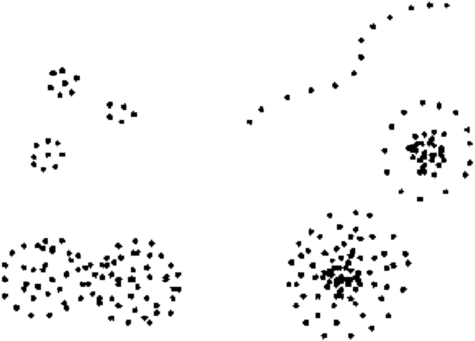


Figure 2: The number of clusters in the data often depend on the resolution (fine vs. coarse) with which we view the data. How many clusters do you see in this figure? 5, 8, 10, more?

Clustering

- Clustering algorithms can be divided into several groups:
  - *Exclusive* (each pattern belongs to only one cluster) vs. *nonexclusive* (each pattern can be assigned to several clusters).
  - *Hierarchical* (nested sequence of partitions) vs. *partitional* (a single partition).
  - *Agglomerative* (merging atomic clusters into larger clusters) vs. *divisive* (subdividing large clusters into smaller ones).

Clustering

- Thousands of clustering algorithms have been proposed in the literature.
- Many of these algorithms are based on the following popular techniques:
  - Iterative squared-error partitioning,
  - Density-based methods,
  - Agglomerative hierarchical clustering.
- One of the main challenges is to select an appropriate measure of similarity to define clusters that is often both data (cluster shape) and context dependent.

Similarity Measures

- The most obvious measure of *similarity* (or dissimilarity) between two patterns is the distance between them.
- If distance is a good measure of dissimilarity, then we can expect the distance between patterns in the same cluster to be significantly less than the distance between patterns in different clusters.
- Then, a very simple way of doing clustering would be to choose a threshold on distance and group the patterns that are closer than this threshold.

Similarity Measures

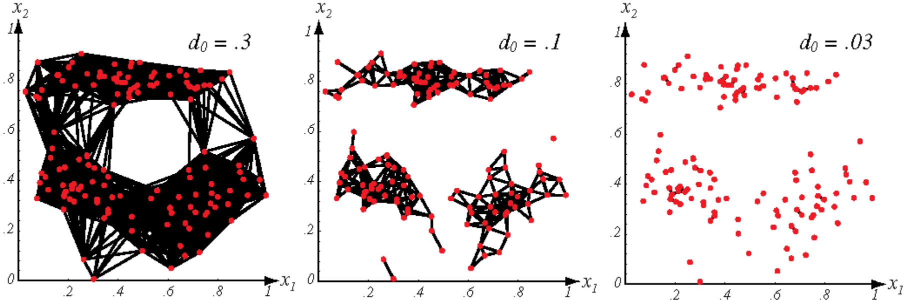


Figure 3: The distance threshold affects the number and size of clusters that are shown by lines drawn between points closer than the threshold.

Criterion Functions

- The next challenge after selecting the similarity measure is the choice of the criterion function to be optimized.
- Suppose that we have a set $\mathcal{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ of n samples that we want to partition into exactly k disjoint subsets $\mathcal{D}_1, \ldots, \mathcal{D}_k$.
- Each subset is to represent a cluster, with samples in the same cluster being somehow more similar to each other than they are to samples in other clusters.
- The simplest and most widely used criterion function for clustering is the *sum-of-squared-error* criterion.

Squared-error Partitioning

- Suppose that the given set of n patterns has somehow been partitioned into k clusters $\mathcal{D}_1, \dots, \mathcal{D}_k$.
- Then, the sum of squared errors is defined by

  $$J_e = \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{D}_i} \|\mathbf{x} - \mathbf{c}_{\boldsymbol{i}}\|^2$$

  where $\mathbf{c}_{\boldsymbol{i}}$ is the point representative of cluster i.
- For a given cluster $\mathcal{D}_i$ with n_i samples, the mean vector

  $$\mathbf{m}_{\boldsymbol{i}} = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}$$

  is the best representative ($\mathbf{c}_{\boldsymbol{i}}$) of the samples in $\mathcal{D}_i$.
- The mean vector corresponds to the centroid of the samples.

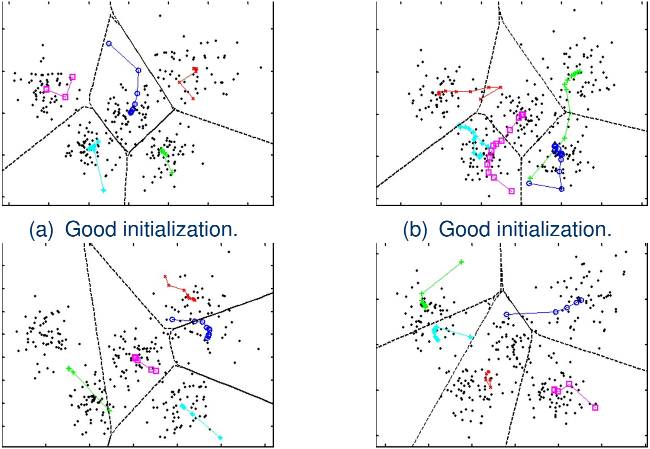
Squared-error Partitioning

- A general algorithm for iterative squared-error partitioning:
  1. Select an initial partition with k clusters. Repeat steps 2 through 5 until the cluster membership stabilizes.
  2. Generate a new partition by assigning each pattern to its closest cluster center.
  3. Compute new cluster centers as the centroids of the clusters.
  4. Repeat steps 2 and 3 until an optimum value of the criterion function is found (e.g., when a local minimum is found or a predefined number of iterations are completed).
  5. Adjust the number of clusters by merging and splitting existing clusters or by removing small or outlier clusters.
- This algorithm, without step 5, is also known as the *k-means* algorithm.

Squared-error Partitioning

- k-means is computationally efficient and gives good results if the clusters are compact and well-separated in the feature space.
- However, choosing k and choosing the initial partition are the main drawbacks of this algorithm.
- The value of k is often chosen empirically or by prior knowledge about the data.
- The initial partition is often chosen by generating k random points uniformly distributed within the range of the data, or by randomly selecting k points from the data.

Squared-error Partitioning



(a) Good initialization.

(b) Good initialization.

(c) Bad initialization.

(d) Bad initialization.

Figure 4: Examples for k-means with different initializations of five clusters for the same data.

Gaussian Mixture Models

- Other problems with k-means include
  - the restriction that each sample is assigned to only one cluster, and
  - the implicit assumption that the clusters are hyperspherical in shape.
- Gaussian mixture models are popular density estimation models that can also be used for clustering.

Gaussian Mixture Models

- The Gaussian mixture model represents a multi-modal distribution as a mixture of uni-modal distributions.
- Each mode is assumed to be a Gaussian that is represented by its mean vector and covariance matrix.
- The parameters of the mixture consist of the parameters of the individual Gaussians and a set of mixture weights.
- These parameters can be estimated using the Expectation-Maximization (EM) algorithm that iteratively
  - computes the expected value of the data likelihood using the current parameter estimates, and
  - finds a new set of parameter values that maximizes this expectation.

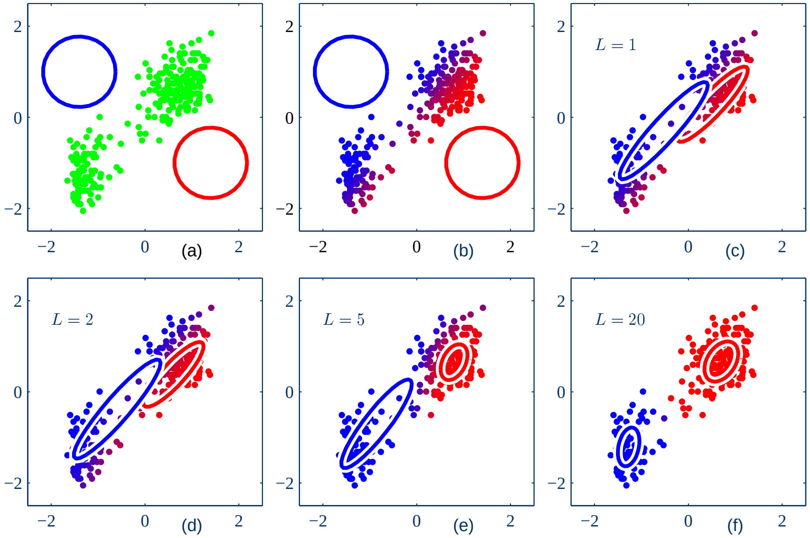
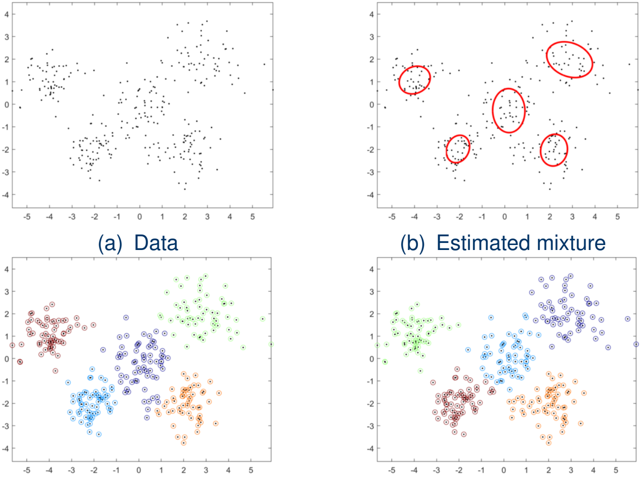


Figure 5: Illustration of the EM algorithm iterations for a mixture of two Gaussians.

Gaussian Mixture Models

- The resulting parameters can be used to compute the probability of each sample being generated by each individual Gaussian.
- These probabilities can be interpreted as soft assignments.
- We can assign each sample to the Gaussian that gives the highest probability if a hard clustering is needed.
- Using arbitrary covariance matrices enables the modeling of clusters with elliptical shapes.
- The k-means algorithm can be considered as a special case where the covariance matrix of each Gaussian is assumed to be spherical.

Gaussian Mixture Models



(a) Data

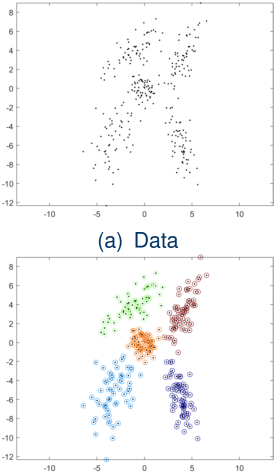
(b) Estimated mixture

(c) Clustering result

(d) k-means result

Figure 6: Gaussian mixture clustering example.

Gaussian Mixture Models



(a) Data

(b) Estimated mixture

(c) Clustering result

(d) k-means result

Figure 7: Gaussian mixture clustering example.

Hierarchical Clustering

- The k-means algorithm produces a *flat* data description where the clusters are disjoint and are at the same level.
- In some applications, groups of patterns share some characteristics when looked at a particular level.
- Hierarchical clustering tries to capture these multi-level groupings using *hierarchical* representations rather than flat partitions.

Hierarchical Clustering

- In hierarchical clustering, for a set of n samples,
  - the first level consists of n clusters (each cluster containing exactly one sample),
  - the second level contains $n - 1$ clusters,
  - the third level contains $n - 2$ clusters,
  - and so on until the last (n'th) level at which all samples form a single cluster.
- Given any two samples, at some level they will be grouped together in the same cluster and remain together at all higher levels.
- A natural representation of hierarchical clustering is a tree, also called a *dendrogram*.

Hierarchical Clustering

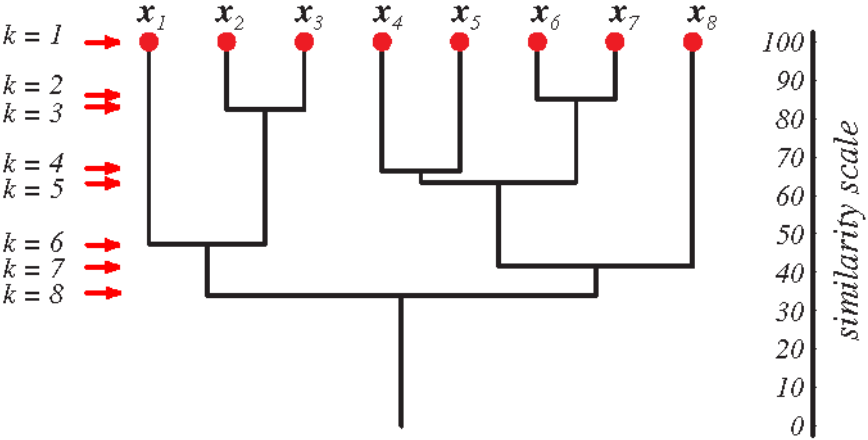


Figure 8: A dendrogram can represent the results of hierarchical clustering algorithms. The vertical axis shows a generalized measure of similarity among clusters.

Hierarchical Clustering

- *Agglomerative Hierarchical Clustering:*
  1. Specify the number of clusters. Place every pattern in a unique cluster and repeat steps 2 and 3 until a partition with the required number of clusters is obtained.
  2. Find the closest clusters according to a distance measure.
  3. Merge these two clusters.
  4. Return the resulting clusters.

Hierarchical Clustering

- Popular distance measures (for two clusters $\mathcal{D}_i$ and $\mathcal{D}_j$):

$$d_{\text{min}}(\mathcal{D}_i, \mathcal{D}_j) = \min_{\substack{\mathbf{x} \in \mathcal{D}_i \\ \mathbf{x'} \in \mathcal{D}_j}} \|\mathbf{x} - \mathbf{x'}\|$$

$$d_{\text{max}}(\mathcal{D}_i, \mathcal{D}_j) = \max_{\substack{\mathbf{x} \in \mathcal{D}_i \\ \mathbf{x'} \in \mathcal{D}_j}} \|\mathbf{x} - \mathbf{x'}\|$$

$$d_{\text{avg}}(\mathcal{D}_i, \mathcal{D}_j) = \frac{1}{\#\mathcal{D}_i \, \#\mathcal{D}_j} \sum_{\mathbf{x} \in \mathcal{D}_i} \sum_{\mathbf{x'} \in \mathcal{D}_j} \|\mathbf{x} - \mathbf{x'}\|$$

$$d_{\text{mean}}(\mathcal{D}_i, \mathcal{D}_j) = \|\mathbf{m}_{\boldsymbol{i}} - \mathbf{m}_{\boldsymbol{j}}\|$$

Hierarchical Clustering

- When d_{min} is used to measure the distance between clusters, the algorithm is called the nearest neighbor clustering algorithm.
- It is also called the *single linkage algorithm* where merging two clusters corresponds to adding an edge between the nearest pair of nodes in these clusters.
- When d_{max} is used to measure the distance between clusters, the algorithm is called the farthest neighbor clustering algorithm.
- It is also called the *complete linkage algorithm* where merging two clusters corresponds to adding edges between every pair of nodes in these clusters.

Hierarchical Clustering

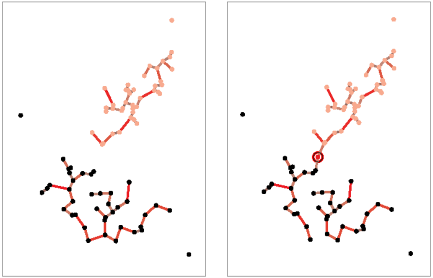


Figure 9: Examples for single linkage clustering.

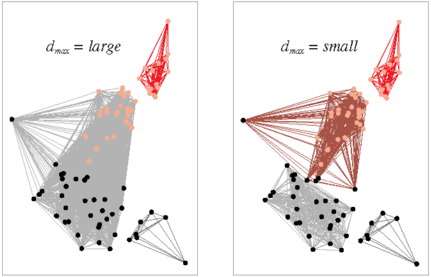
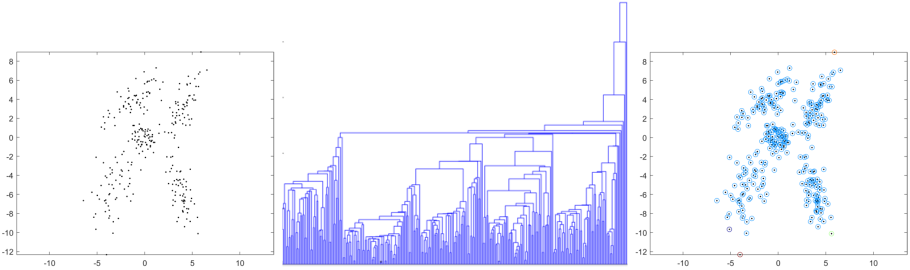


Figure 10: Examples for complete linkage clustering.

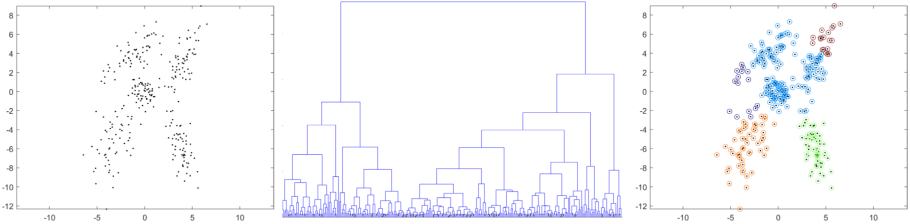
Hierarchical Clustering



(a) Data (b) Dendrogram (c) Clustering result

Figure 11: Single linkage clustering example.

Hierarchical Clustering



(a) Data (b) Dendrogram (c) Clustering result

Figure 12: Complete linkage clustering example.

Cluster Validity

- Methods for validating the results of a clustering algorithm include:
  - Repeating the clustering procedure for different values of the parameters, and examining the resulting values of the criterion function for large jumps or stable ranges.
  - Formulating hypothesis tests that check whether multiple clusters found have been formed by chance, and whether the observed change in the error criterion has any significance.
  - Comparing the groupings by the unsupervised clustering to the known labels in the ground truth.