Week 2: Cluster Analysis

CMPS 320: Machine Learning

Introduction

- In today's lectures, we discuss unsupervised learning.
  - a set of statistical tools intended for the setting in which we have only a set of features $X_1, X_2, \cdots, X_p$ measured on n observations.
- We are not interested in prediction, because we do not have an associated response variable Y.
- The goal is to discover interesting things about the measurements on $X_1, X_2, \cdots, X_p$.
  - Is there an informative way to visualize the data?
  - Can we discover subgroups among the variables or among the observations?
- We will focus on **clustering**, a broad class of methods for discovering unknown subgroups in data.

Clustering

- Clustering refers to a very broad set of techniques for finding homogeneous subgroups, or clusters, in a data set.
- Big idea: partition observations into distinct groups such that
  - observations within each group are similar to each other
  - observations in different groups are different from each other
- What we need: a clear idea of what it means for two or more observations to be similar or different.

Applications of Clustering

- Business intelligence (Market Segmentation)
  - Clustering can be used to organize a large number of customers into groups, where customers within a group share strong similar characteristics.
  - This facilitates the development of business strategies for enhanced customer relationship management.
- Image recognition
  - Clustering can be used to discover clusters or “subclasses” in handwritten character recognition systems.
- Web search
  - Clustering techniques have been developed to cluster documents into topics, which are commonly used in information retrieval practice.
- Outlier detection
  - Detection of credit card fraud and the monitoring of criminal activities in electronic commerce

Clustering

- There exist a great number of clustering methods.
- In this course we will focus on two best-known clustering approaches:
    - **K-means clustering**: we seek to partition the observations into a pre-specified number of clusters
    - **Hierarchical clustering**: we do not know in advance how many clusters we want; we end up with a tree-like visual representation of the observations, called a **dendrogram**.
        - ★ Dendrogram allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to n.
- In general,
    - we can cluster observations on the basis of the features in order to identify subgroups among the observations, or
    - we can cluster features on the basis of the observations in order to discover subgroups among the features.

K-Means Clustering

- K-means clustering is an approach for partitioning a data set into K distinct, non-overlapping clusters.
- To perform K-means clustering, we must first specify the desired number of clusters K.
- Partition a data set into K distinct, non-overlapping clusters.

"apples"

"oranges"

K-Means Clustering– cont.

- Big idea: good clustering is one for which within-cluster variation is small
- The within-cluster variation for cluster C_k is a measure $W(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Mathematically, we want to solve the problem:

$$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

  where $C_1, \cdots, C_K$ denote sets containing the indices of the observations in each cluster.
- We often use Euclidean distance:

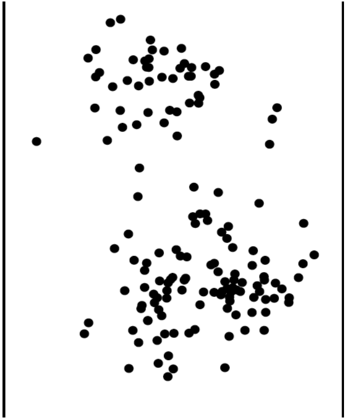
$$W(C_k) = \underbrace{\frac{1}{|C_k|} \sum_{i,i' \in C_k}}_{\substack{\text{Average over} \\ \text{all pairs of obs.} \\ \text{in cluster}}} \underbrace{\sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2}_{\substack{\text{Euclidean} \\ \text{distance}}}$$

K-Means algorithm

1. Randomly assign a number, from 1 to K, to each of the observations. each observation to a cluster.
   - These serve as initial cluster assignments for the observation
2. Iterate until the cluster assignments stop changing:
   - For each of the K clusters, compute the cluster **centroid**.
     - The kth cluster **centroid** is the vector of the p feature means for the observations in the kth cluster.
   - Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

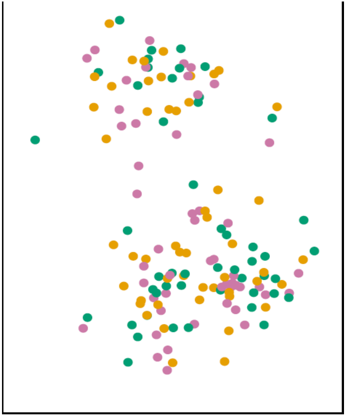
K-Means– Example

- Data/ Observation



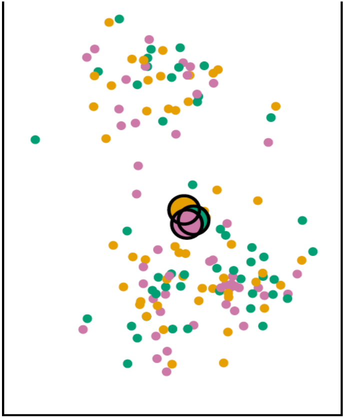
K-Means– Example

- Each data/observation is randomly assigned to a cluster ($K = 3$)



K-Means– Example

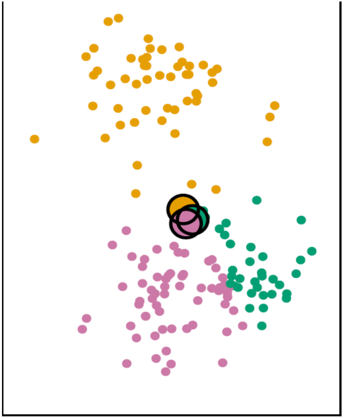
- Compute cluster centroids. These are shown as large colored disks.



- The centroids are almost completely overlapping because the initial cluster assignments were chosen at random.

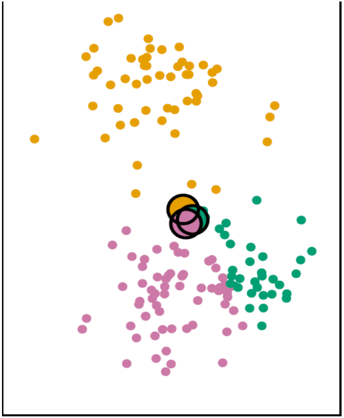
K-Means– Example

- Each observation is assigned to the nearest centroid.



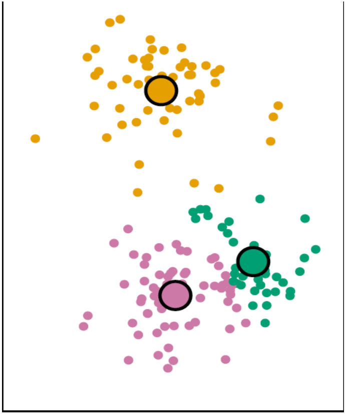
K-Means– Example

- Recompute centroids



K-Means– Example

- Repeat until clusters stabilize



K-Means – Remarks.

- Because the K-means algorithm finds a local rather than a global optimum, the results obtained will depend on the initial (random) cluster assignment of each observation in Step 1 of Algorithm the algorithm.
- Therefore it is important to run the algorithm multiple times from different random configurations.
- Then one selects the best solution, i.e. that for which the objective function is smallest.

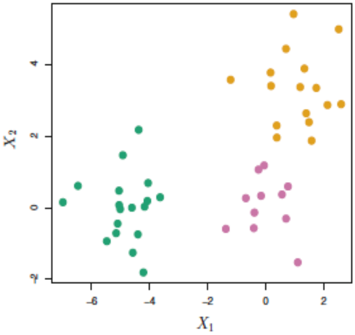
K-Means clustering

- To perform K-means clustering, we must decide how many clusters we expect in the data.
- The problem of selecting K is not simple.
  - This issue, along with other practical considerations that arise in performing K-means clustering is addressed next.

Hierarchical clustering (bottom-up or agglomerative)

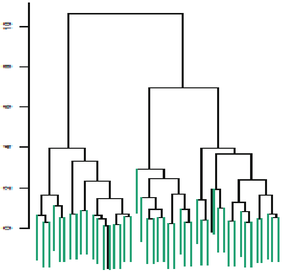
- Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K.
- It has an added advantage over K-means clustering in that it results in an attractive tree-based representation of the observations, called a **dendrogram**.
- Hierarchical clustering can be divided into two main types:
  - agglomerative: It works in a bottom-up manner– dendrogram is built starting from the leaves and combining clusters up to the trunk.
  - divisive: It works in a top-down manner (i.e. the inverse of agglomerative)

Interpreting dendrograms



- Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors.
- We will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.

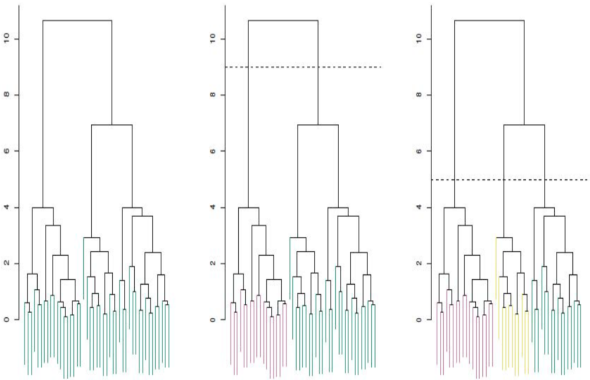
Interpreting dendrograms–cont.



- Each leaf of the dendrogram represents one of the 45 observations.
- As we move up the tree, some leaves begin to fuse into branches.
  - These correspond to observations that are similar to each other.
- As we move higher up the tree, branches themselves fuse, either with leaves or other branches.
  - Observations that fuse close to the top of the tree will tend to be quite different.

Interpreting dendrograms–cont.

- To go from a dendrogram to actual clusters, just cut



- The height of the cut serves the same role as the K in K-means clustering: it controls the number of clusters.

Selecting K clusters

- In practice, people often look at the dendrogram and select by eye a sensible number of clusters, based on the heights of the fusion and the number of clusters desired.

Algorithm–Hierarchical clustering

- Begin with n observations and a measure of all the (n choose 2) pairwise distances. Treat each observation as its own cluster.
- For $i = n, n-1, \cdots, 2$:
  - Examine all pairwise inter-cluster distances and identify the pair of clusters that are most similar.
  - Fuse these two clusters. The distances between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
  - Compute the new pairwise inter-cluster distances.

Algorithm–Hierarchical clustering

- How do we measure distance between clusters?
- Answer: Using linkage.
- Linkage Types:
  - **Complete**: maximal intercluster distance i.e. compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.
  - **Single**: minimal intercluster distance i.e. compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities.
  - **Average**: mean intercluster distance i.e. compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.
  - **Centroid**: distance between cluster means i.e. dissimilarity between the centroid for cluster A and the centroid for cluster B.