Clustering

Predictive Modeling & Statistical Learning

Gaston Sanchez

Introduction

Clustering Examples

Clustering refers to a very broad set of techniques for finding groups, or clusters, in a data set.

Clustering Examples

- **Marketing**: discover groups of customers and used them for targeted marketing
- **Astronomy**: find groups of similar stars and galaxies
- **Genomics**: find groups of genes with similar expressions

Clustering Idea

Group a set of n objects in K groups such that:

- each group is as much homogeneous as possible (*within homogeneity*)
- groups are as distinct as possible among them (*between heterogeneity*)

Clustering Idea

- We seek a partition of the data into distinct groups.
- We want the observations within each group to be quite similar to each other.
- We must define what it means for two or more observations to be similar or different.
- This is often a domain-specific consideration that must be made based on knowledge of the data being studied.

Assupmtions

We will assume that the rows of the data matrix correspond to the individuals to be clustered (although you could also cluster variables).

We will assume that the individuals are embeded in an euclidean space (e.g. quantitative variables, or output of a PCA)

Proximity Measures

The proximity measure need to reflect the actual proximity between objects according to the final aim of the clustering.

Weighting the attributes might be necessary

Proximity Measures

Euclidean Distance between two individuals $\mathbf{x_i}$ and $\mathbf{x_l}$

$$d^2(\mathbf{x_i}, \mathbf{x_l}) = \sum_{j=1}^{p} (x_{ij} - x_{lj})^2$$

Clustering Algorithms

Clustering

There are many different types of clustering methods

We will concentrate on two of the most commonly used approaches:

- K-Means clustering
- Hierarchical clustering

Common Algorithms

Direct Partitioning

- K-means
- K-medoids

Hierarchical: Bottom-Up

- Single linkage, average linkage,
- Ward

K-Means

How does K-Means work?

We would like to partition that data set into K non-overlapping clusters

$$C_1, C_2, \ldots, C_K$$

For instance, if the i-th observation is in the k-th cluster, then $i \in C_k$.

Preliminary Concepts

Let $C_1, C_2, \ldots, C_K$ denote sets containing the indices of the observations in each cluster.

- $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.
- $C_k \cap C_h = \emptyset$ for all $k \neq h$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

How does K-Means work?

The goal is to have a minimal "within-cluster variation", i.e. the elements within a cluster should be as similar as possible.

One way of achieving this is to minimize the sum of all the pair-wise squared Euclidean distances between the observations in each cluster:

$$min \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,l \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{lj})^2 \right\}$$

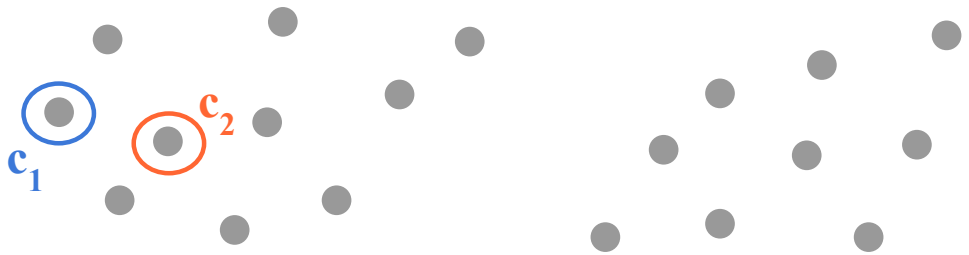
K-Means Algorithm

Initialization: Randomly select K centers $\mathbf{c_k}$

Do for 1:niter

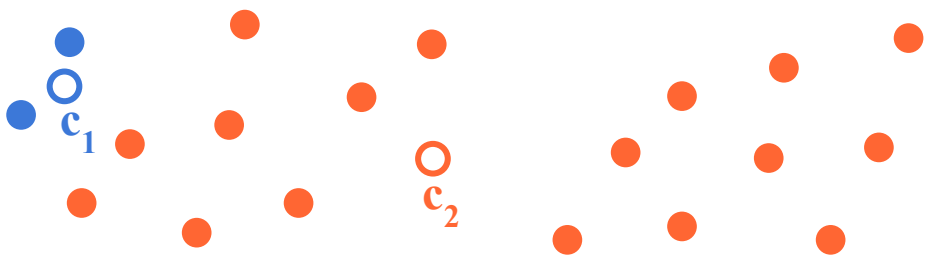
- Assign every individual to the closest center
- Definition of the new partition
- Update centers as the centroids of every class
- Stop when:
  - the old centers and the new centers are sufficiently similar
  - or the maximum number of iterations is reached

K-Means



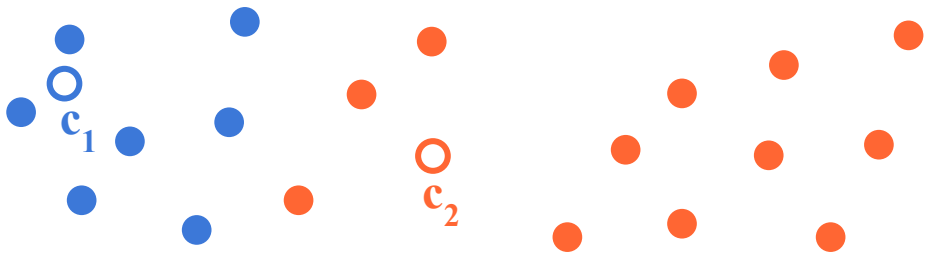
Start with random centroids

Assign observations to closest centroid

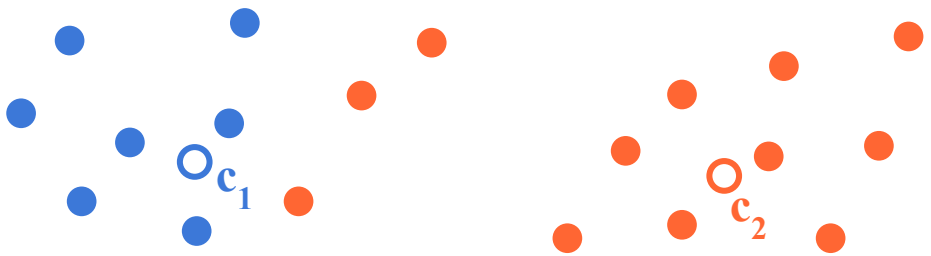


Recalculate centroids

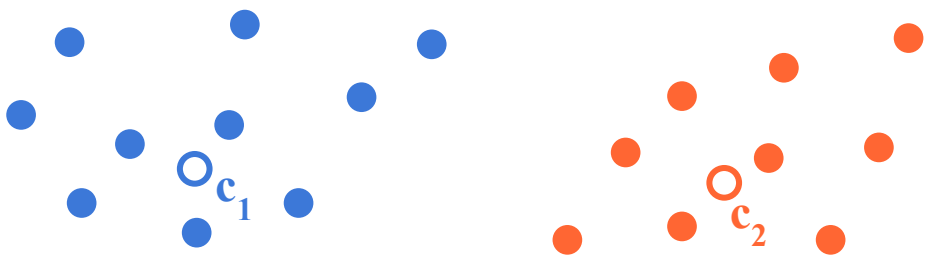
K-Means



Reassign observations to closest centroid



Recalculate centroids



Reassign observations to closest centroid

What does K-Means optimize?

Assume that the n observations have masses m_i

Let $d^2(\mathbf{x_i}, \mathbf{c_k^s})$ be the squared distance between observation i and the centroid of the group k, at step s.

We focus on the following criterion:

$$v(s) = \sum_{k=1}^{K} \left\{ \sum_{i \in C_k^s} m_i d^2(\mathbf{x_i}, \mathbf{c_k^s}) \right\}$$

What does K-Means optimize?

At the s-th step, the group C_k^s is formed of those observations that are closest to the centroid $\mathbf{c_k^s}$

The within-gropus variance at step s is given by:

$$V(s) = \sum_{k=1}^{K} \left\{ \sum_{i \in C_k^s} m_i d^2(\mathbf{x_i}, \mathbf{c_k^{s+1}}) \right\}$$

where $\mathbf{c_k^{s+1}}$ is the centroid of class C_k

What does K-Means optimize?

It can be shown that:

$$v(s) \geq V(s) + v(s+1)$$

- The objective criterion decreases (i.e. non-decreasing function).
- This guarantees convergence of the K-Means algorithm.
- It usually converges fast.

Local Optima

The K-means algorithm can get stuck in "local optima" and not find the best partition.

Hence, it is important to run the algorithm multiple times with random starting points to find a good solution.

K-Means: local optima

K-Means Algorithm

- Choose the seeds "wisely"
- Extensions: K-medoids, Kohonen maps, ...

Fast K-Means Algorithm

Quick and dirty

- Randomly select $\mathbf{c_k}$ centers
- Assign the first individual to the closest center
- Update the new center of the affected class
- Assign the second individual to its closes center
- Update the new center of the affected class
- ...

References

- **Modern Multivariate Statistical Techniques** by Julian Izenman (2008). *Chapter 12: Cluster Analysis*. Springer.
- **Data Mining and Statistics for Decision Making** by Stephane Tuffery (2011). *Chapter 9: Cluster Analysis*. Wiley.

References (French literature)

- **Probabilites, analyse des donnees et statistique** by Gilbert Saporta (2006). *Chapter 11: Methodes de classification*. Editions Technip, Paris.
- **Statistique Exploratoire Multidimensionnelle** by Lebart et al (2004). *Chapter 2, section 2.1: Agregation autour des centres mobiles*. Dunod, Paris.
- **Approche pragmatique de la classification** by Nakache and Confais (2005). *Chapter 4: Classification par partition*. Editions Technip, Paris.
- **Statistique: Methodes pour decrire, expliquer et prevoir** by Michel Tenenhaus (2008). *Chapter 9: Analyses des proximites, des preferences et typologie*. Editions Technip, Paris.