# 40.016 The Analytics Edge

## Recommendation systems (Part 1)

Stefano Galelli

SUTD

Term 6, 2020

# Outline

- Recommendation systems
- Netflix prize
- Clustering
- Hierarchical clustering
- K-means clustering

# Recommendation systems

- Personalize the user experience for online applications
- Leverage data on items and customers (e.g., likes, purchase history)
- A key challenge: they must be fast and accurate
- Common underlying analytics: clustering, collaborative and content filtering

# Netflix prize

**Cinematch**

- Launched in 2000
- Accurate within half a star 75% of the time
- Need for a more advanced recommendation system

# Netflix prize

**Key aspects of the competition**

- 1 million USD prize for improving Cinematch by more than 10% on the test dataset
- Training dataset: 100,480,507 ratings that 480,189 users gave to $\sim$ 18,000 movies
- Test dataset: 2,817,131 data points, used for the public and private leaderboard
- Test predictions were scored against the true ratings (performance measured with the RMSE)

# Netflix prize

**Timeline**

- October 2006: the competition was launched
- November 2007: *BellKor* wins the 50,000 USD progress prize with a 8.43% improvement over Cinematch
- June 2009: Team *BellKor Pragmatic Chaos* achieves 10.05% improvement
- September 2009: Team *BellKor Pragmatic Chaos* officially wins the competition

A few words from the winner: *From the Labs: Winning the Netflix Prize*

# Clustering

**Goal:** to partition observations into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other.

**Note:** This is an *unsupervised problem*, because we are trying to discover clusters (or, in general, a structure) within a dataset (and not to predict the outcome of a given variable).

# Clustering

**Difference w.r.t. PCA**

They both try to simplify the data, but:

- PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance
- Clustering looks to find homogeneous subgroups among the observations

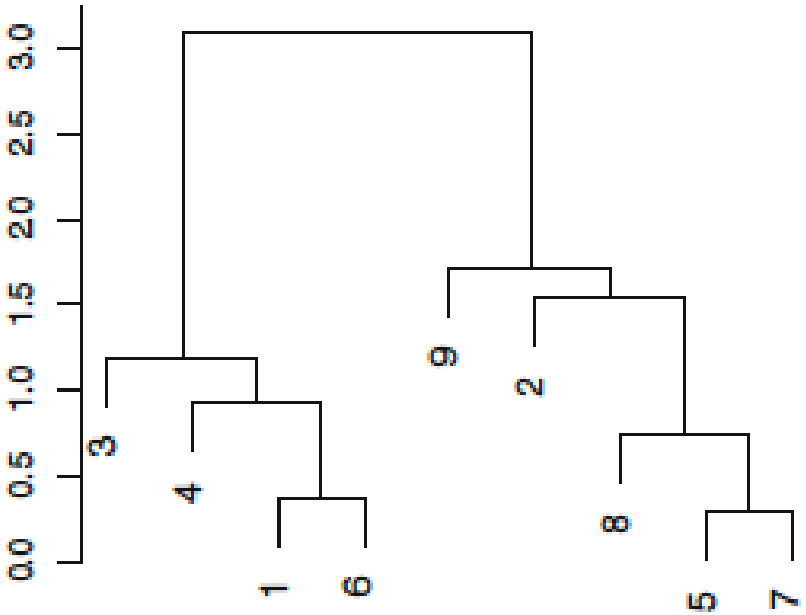
# Clustering

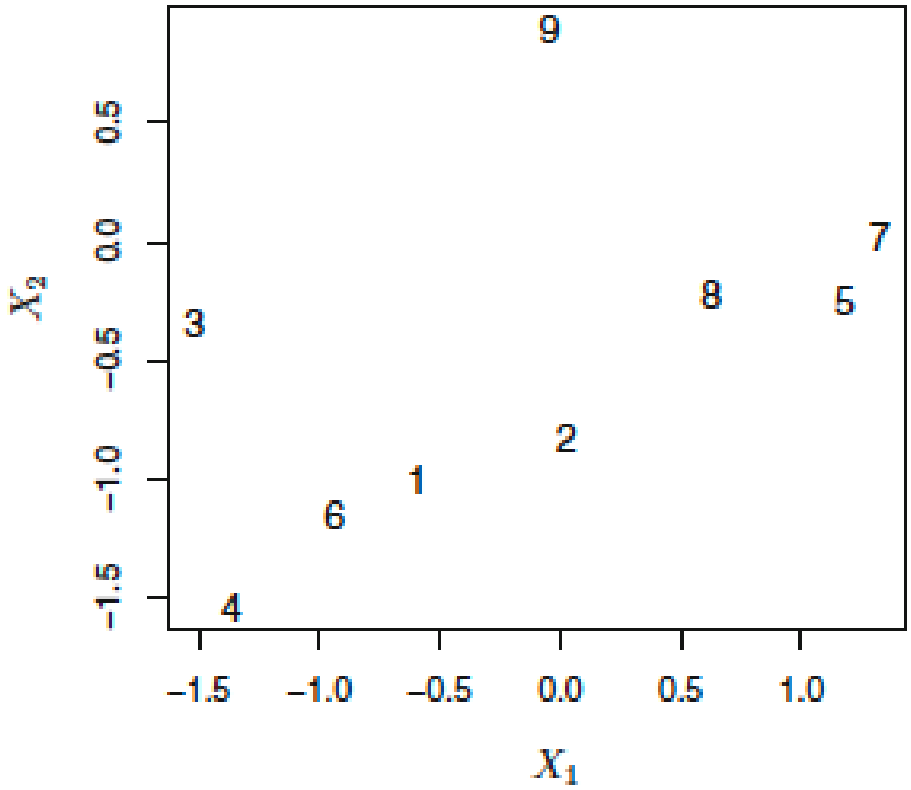
**Common methods**

- *Hierarchical clustering*: we do not know in advance how many clusters we want (bottom-up, or agglomerative)
- *K-means clustering*: we partition the observations into a pre-specified number of clusters (top-down)

# Hierarchical clustering

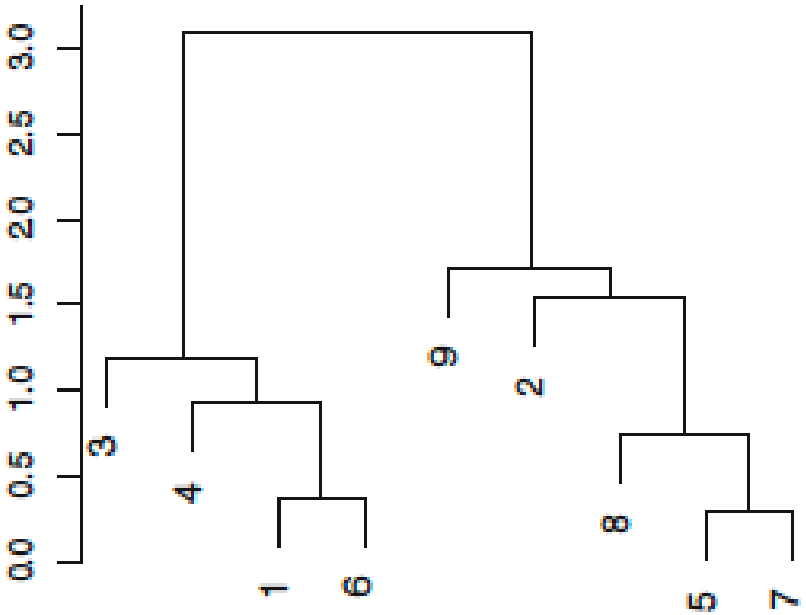
**A motivating example (from James et al. (2014)):**

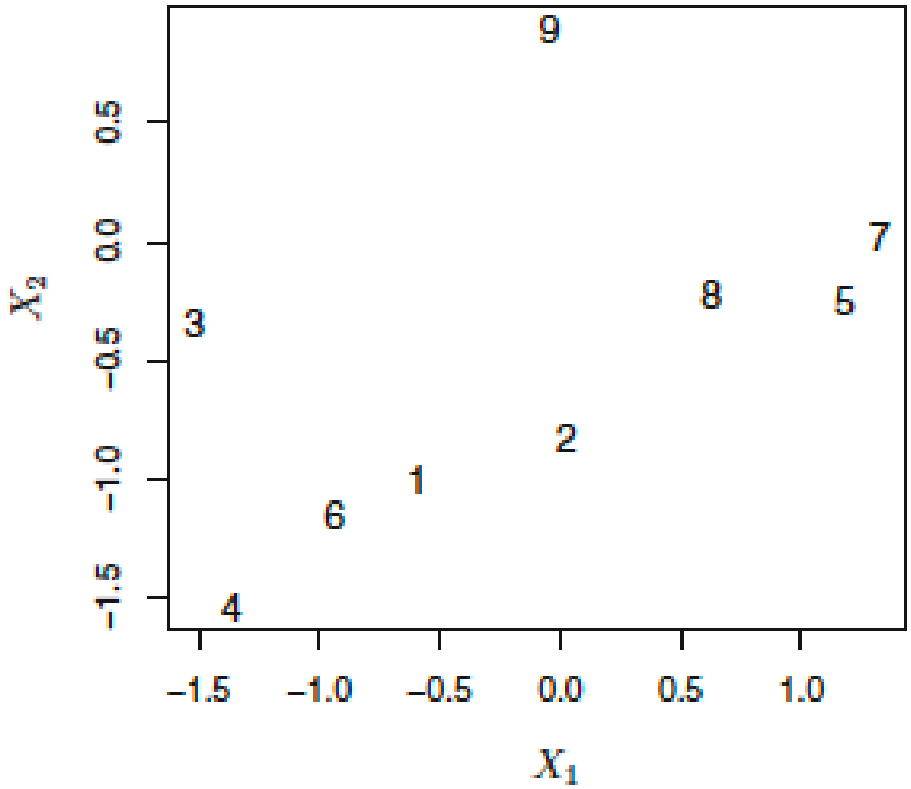


# Hierarchical clustering

**A motivating example (from James et al. (2014)):**



# Hierarchical clustering

**Algorithm:**

1. Begin with $n$ observations and a measure (such as Euclidean distance) of all the $n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
2. For $i = n, n-1, \ldots, 2$:
   - Examine all pairwise inter-cluster dissimilarities among the $i$ clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters.
   - Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters

# Hierarchical clustering

How do we define the dissimilarity between two clusters when we have multiple obs. in a cluster? Common types of **linkage**:

- *Complete* (maximal intercluster dissimilarity)
- *Average* (minimal intercluster dissimilarity)
- *Single* (mean intercluster dissimilarity)
- *Centroid* (dissimilarity between centroids)

# K-means clustering

**Notation:** Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$ (each observation belongs to at least one of the $K$ clusters)
2. $C_k \cap C_{k'} = \{\}$ for all $k \neq k'$ (clusters are non-overlapping)

# K-means clustering

**Idea:** a good clustering is one for which the *within-cluster variation* is as small as possible. That means we want to solve the following problem:

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

We quantify the within-cluster variation with the *squared Euclidean distance*, that is:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

# K-means clustering

**Idea (cont'd):** Putting the previous equations together, we get

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

This is the optimization problem we want to solve. The K-means algorithm uses an heuristic search process to solve it.

# K-means clustering

**Algorithm:**

1. Randomly assign a number, from 1 to K, to each of the observations (initial cluster assignment)
2. Iterate until the cluster assignment does not change:
   - Assignment: For each of the K clusters, compute the cluster *centroid* (for the $k$-th cluster, the centroid is the vector of the $p$ feature means for the observations in the $k$-th cluster)
   - Update: Assign each observation to the cluster whose centroid is closest

# K-means clustering

## Illustration of the search process for a toy case

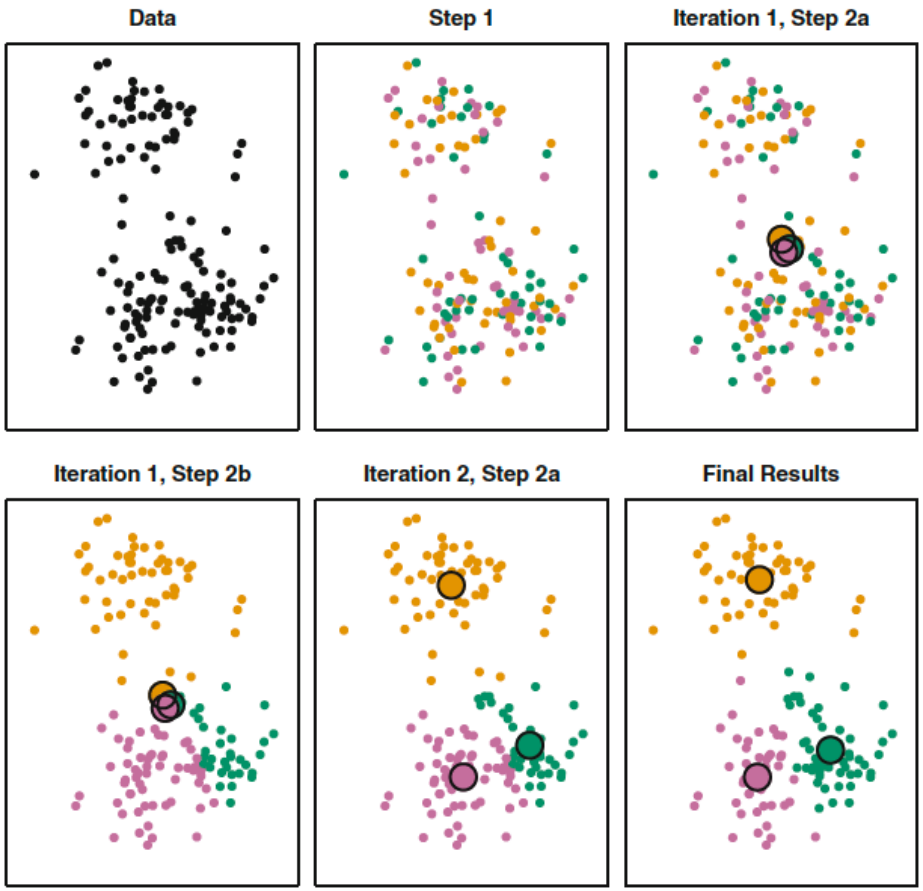



Figure 1: Source: James et al. (2014)

# K-means clustering

**A couple of considerations:**

- The results will depend on the initialization
- So, it is good practice to run the algorithm maultiple times (with different initialization)
- We will the pick the configuration that minimizes the value of the objective function

# Final remarks

Some important decisions we generally have to make:

- Should the observations or features first be standardized in some way?
- For hierarchical clustering:
  - What type of linkage should be used?
  - Where should we cut the dendrogram in order to obtain clusters?
  - (What dissimilarity measure should be used?)
- For K-means clustering:
  - How many clusters should we use?

# References

**Clustering and Recommendation systems**

- Teaching notes.
- James et al. (2014) *An Introduction to Statistical Learning with Applications in R*, Springer, 2014. Chapter 10.3.

**Netflix**

- Koren Y. "The Bellkor solution to the Netflix grand prize." Netflix prize documentation 81.2009 (2009): 1-10.
- *From the Labs: Winning the Netflix Prize*
- *Why Netflix's Algorithm Is So Binge-Worthy*
- *Why your Netflix thumbnails don't look like mine*