Mixtures and Clustering Algorithms [1]

Jasmine. Hao[1]

[1]University of Hong Kong

ECON 6083: Machine Learning

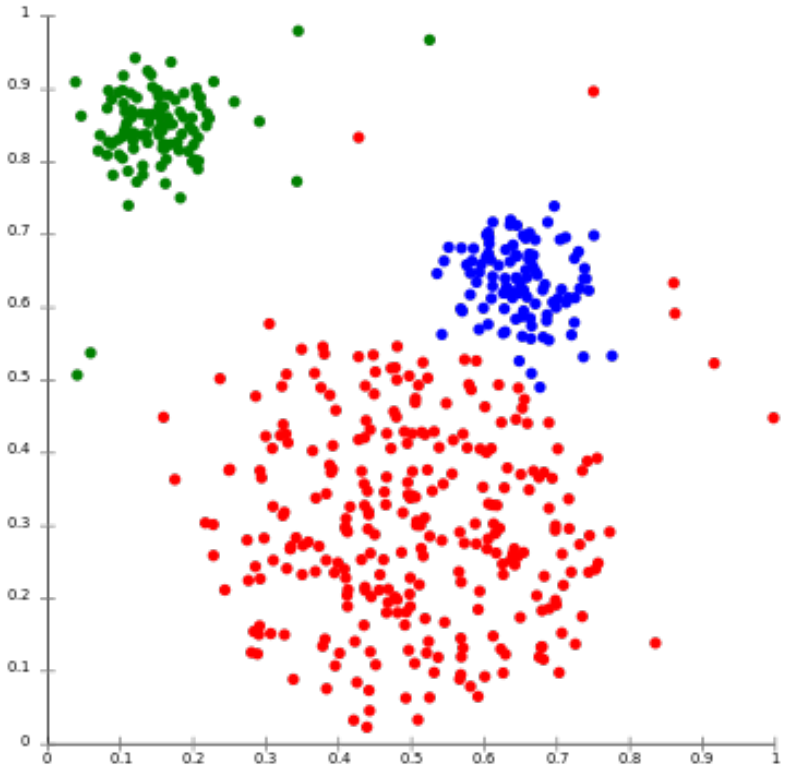
[1]This section is based on [Compiani and Kitamura, 2016], Sections 1-3; [Judd et al., 2010], Sections 1-3.

Outline

1. Compiani & Kitamura (2016): Mixtures
2. The k-means clustering
3. Judd, Maliar & Maliar (2010): Clustering Algorithms
   - Overview
   - Hierarchical Algorithm
   - K-means Algorithm

Stereotypical Clustering

Note: Points are samples plotted in feature space, e.g., 10,000-dimensional space for 100x100 images.

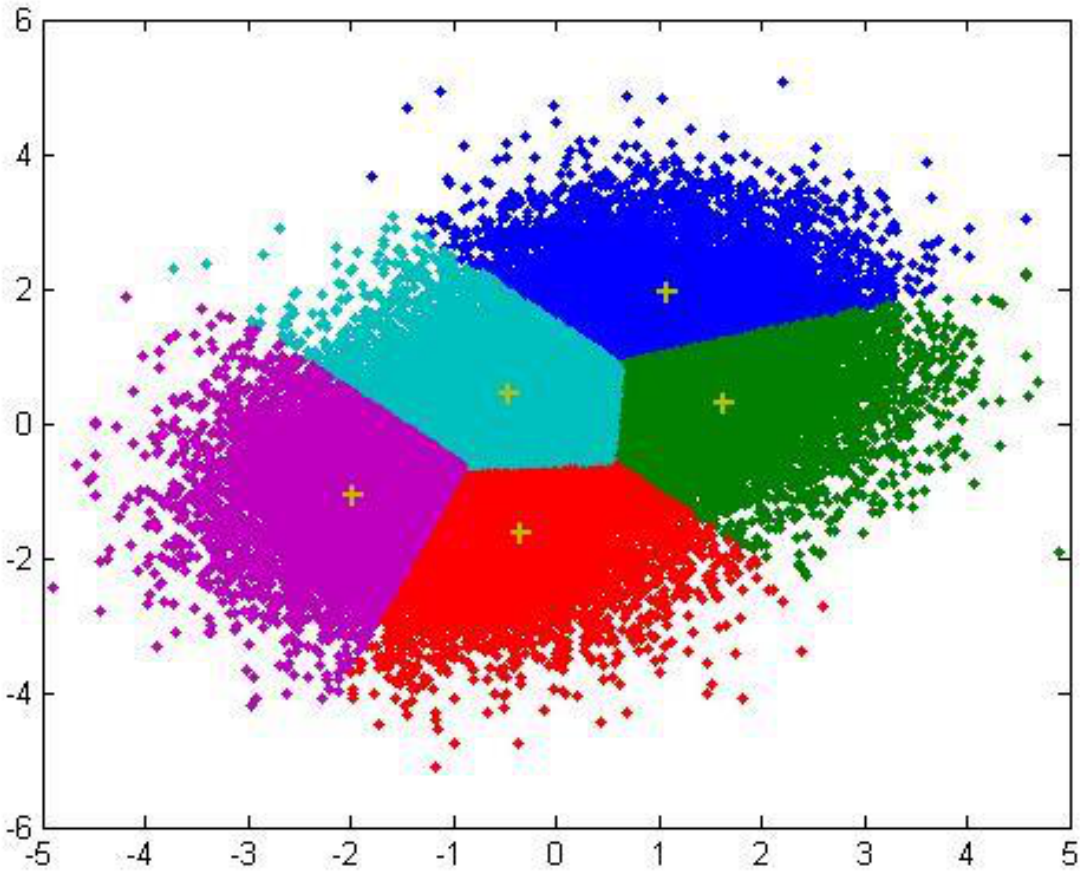


K-means Clustering

- The standard k-means algorithm is based on **Euclidean distance**.
- The cluster quality measure is an **intra-cluster measure only**, equivalent to the sum of item-to-centroid kernels.
- A simple greedy algorithm locally optimizes this measure (usually called Lloyd's algorithm):
  - **Find the closest cluster center** for each item and assign it to that cluster.
  - **Recompute the cluster centroid** as the mean of items for the newly-assigned items in the cluster.

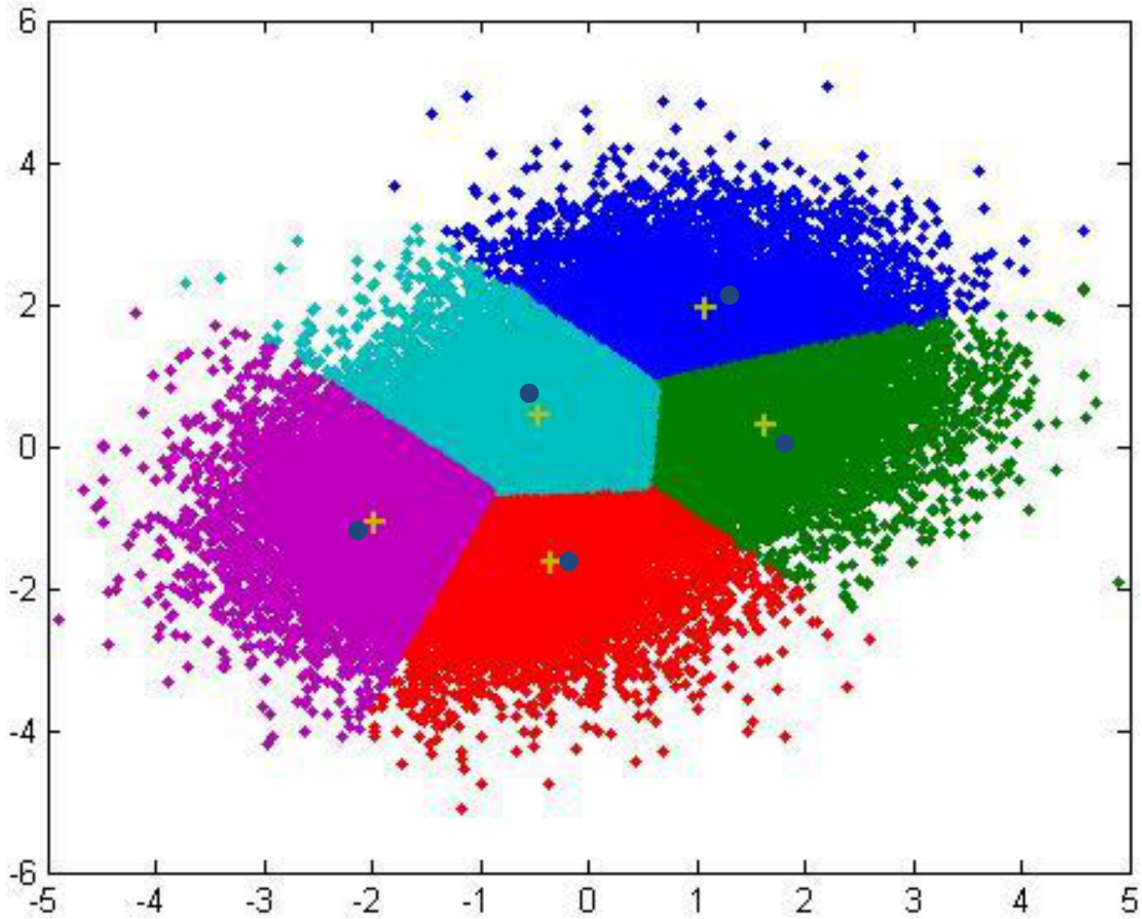
K-means clustering

Assign points to closest center



K-means clustering

Recompute centers (old = cross, new = dot)



K-means Clustering

Iterate:

- For fixed number of iterations
- Until no change in assignments
- Until small change in quality

The k-means clustering algorithm

In the clustering problem, we are given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$, and want to group the data into a few cohesive "clusters." Here, $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ are given. So, this is an unsupervised learning problem.
The k-means clustering algorithm is as follows:

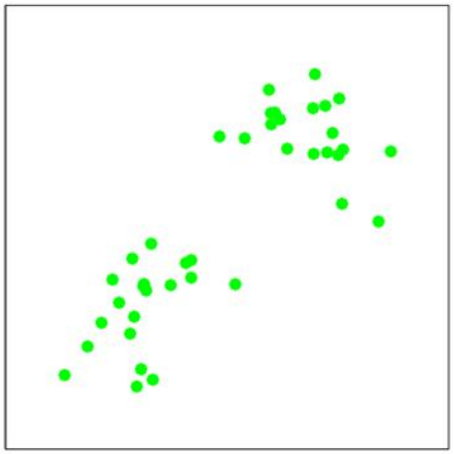
1. Initialize cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence:
   1. For every i, set

$$\text{For each } j, \text{ set}$$
$$c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2$$

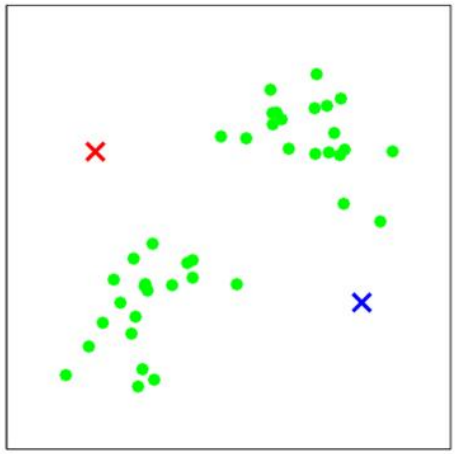
$$\mu_j := \frac{1}{\sum_{i=1}^{m} \mathbb{1}\{c^{(i)}=j\}} \sum_{i=1}^{m} \mathbb{1}\{c^{(i)} = j\} x^{(i)}$$

Understanding the K-Means Clustering Algorithm

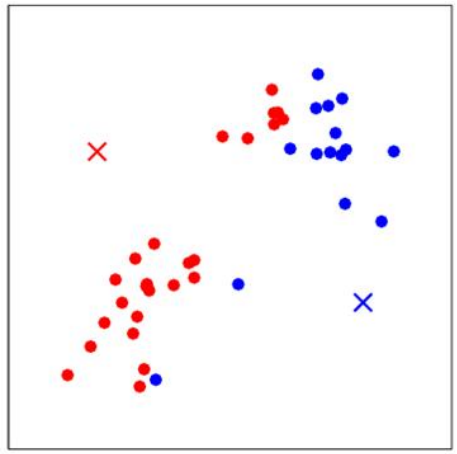
- **Algorithm Parameter**: The variable k signifies the number of clusters to be identified.
- **Cluster Centroids**: The terms μ_j represent our initial estimates for the cluster centers.
- **Initialization**:
  - Step 1: Initialize the cluster centroids by randomly selecting k examples from the training set. These serve as the initial positions of the cluster centroids.
  - Note: Alternative initialization strategies can also be employed.
- **The Inner Loop**: The algorithm then iterates through two main steps:
  1. Assignment Step: Assign each training example $x^{(i)}$ to the nearest cluster centroid μ_j.
  2. Update Step: Update each cluster centroid μ_j to be the mean of all points assigned to it.



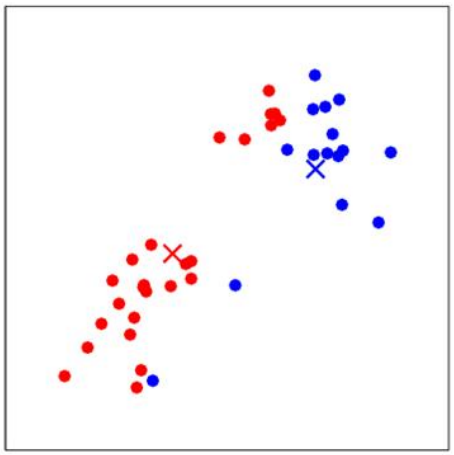
(a) Step 1



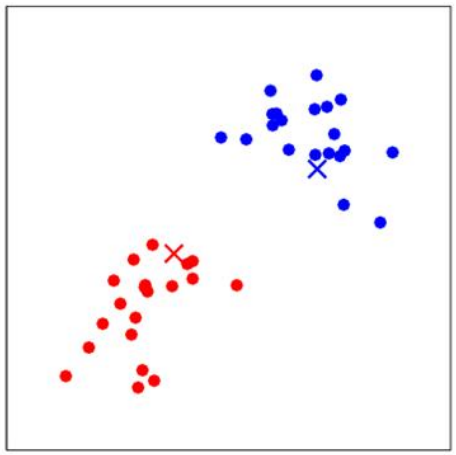
(b) Step 2



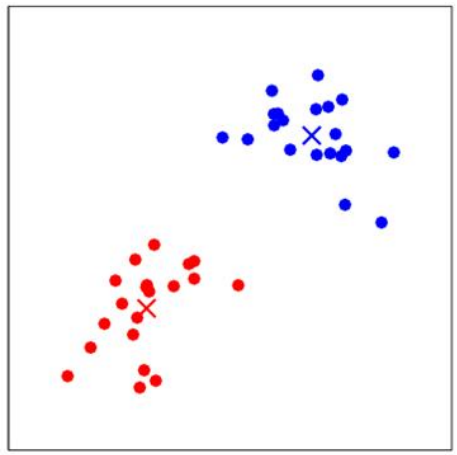
(c) Step 3



(d) Step 4



(e) Step 5



(f) Step 6

Visualization of the K-Means Algorithm

- Training examples are represented as dots, and cluster centroids as crosses.
- **(a) Original dataset**: The starting point of the algorithm.
- **(b) Initial cluster centroids**: Random initial placement of centroids (not necessarily on training examples).
- **(c-f) Two iterations of K-means**:
  1. Assignment Step: Assign each training example to the closest centroid, indicated by color.
  2. Update Step: Move each centroid to the average position of the assigned examples.
- Best viewed in color to observe the "painting" of examples.

Convergence Guarantee of the K-Means Algorithm

- The K-means algorithm is guaranteed to converge, minimizing the distortion function $J(c, \mu)$.
- **Distortion Function** $J(c, \mu)$:
  - Measures the sum of squared distances between each training example and its assigned centroid.
  - $J(c, \mu) = \sum_{i=1}^{m} \|x^{(i)} - \mu_{c(i)}\|^2$, where m is the number of training examples.
- **Minimization of J**:
  1. With respect to c: Assignment of examples to centroids.
  2. With respect to μ: Update of centroid positions.
- The value of J monotonically decreases, leading to convergence of both cluster assignments and centroids.

Behavior and Optimization of K-Means Clustering

- Oscillation in K-Means:
  - K-means may oscillate between clusterings with equal distortion J, but this is rare in practice.
- Non-Convex Nature of Distortion Function J:
  - J is non-convex, hence coordinate descent does not guarantee global minimum convergence.
  - K-means may converge to local optima, affecting the quality of clustering.
- Practical Performance:
  - Despite non-convexity, K-means often performs well and finds good clusterings.
- Avoiding Bad Local Minima:
  - Run K-means multiple times with different random initial centroids.
  - Choose the clustering with the lowest distortion $J(c, \mu)$ from the multiple runs.

K-means properties

- It's a greedy algorithm with random setup – **solution isn't optimal** and varies significantly with different initial points.
- Very simple convergence proofs.
- **Performance is O(nk) per iteration**, not bad and can be heuristically improved.
  - n = total features in the dataset, k = number clusters
- Many generalizations, e.g.
  - Fixed-size clusters
  - Simple generalization to m-best soft clustering
- As a "local" clustering method, it works well for data condensation/compression.

Choosing clustering dimension

- AIC or Akaike Information Criterion:

$$\textbf{AIC:} \qquad K = \arg\min_{K}[-2L(K) + 2q(K)]$$

- K = dimension, $L(K)$ is the likelihood (could be RSS) and $q(K)$ is a measure of model complexity (cluster description complexity).
- AIC favors more compact (fewer clusters) clusterings.
- For sparse data, AIC will incorporate the number of non-zeros in the cluster spec. Lower is better.

Outline

1. Compiani & Kitamura (2016): Mixtures
2. The k-means clustering
3. Judd, Maliar & Maliar (2010): Clustering Algorithms
   - Overview
   - Hierarchical Algorithm
   - K-means Algorithm

Motivation

Judd, K. L., Maliar, L., & Maliar, S. (2010). *A cluster-grid projection method: solving problems with high dimensionality* (No. w15965). National Bureau of Economic Research.

Dynamic economic models with heterogeneous agents can be computationally expensive to solve.

- The cost grows exponentially if an algorithm relies on a tensor-product rule for grid construction or numerical integration.

This paper presents a simple **projection method**.

- The method operates on the **ergodic set realized in equilibrium**, avoiding costs associated with unsampled areas of the state space.
- The method utilizes **clustering algorithms** to construct a grid of points representing the ergodic set.

Method Overview

1. Guess a decision rule and simulate a time series solution.
2. Distinguish clusters on the simulated series using **clustering algorithms**.
   - Use **hierarchal** and ***K*-means** algorithms.
   - replace a large number of closely-located simulated points with a given small number of uniformly distributed **representative points**.
3. Compute **cluster centers** and use them as a **grid of points for projections**.
   1. Parameterize decision rules using **polynomial functions** for fast and numerically stable approximation.
   2. Evaluate conditional expectations using low-cost **numerical integration** formulas suitable for high-dimensional applications.
   3. Solve for polynomial coefficients using a **fixed-point iteration** procedure.
   4. Use the cluster-grid method itself to initialize the construction of the ergodic set.

Our lecture would only focus on the second step (clustering algorithms).

Literature I

(1) **Cluster-grid method**

- Similar to stochastic simulation methods of Fair & Taylor (1984); Den Haan & Marcet (1990); Rust (1996); Pakes & McGuire (2001); Judd, Maliar & Maliar (2009).

  Unlike the literature, this paper differs from 2 aspects:

- Use a **cluster-grid representation** of the ergodic set, which is more efficient than a set of original closely-located simulated points
- Use **numerical integration methods** that are unrelated to estimated density function and are more accurate than MC integration method.

Literature II

(2) **Algorithm**

- Similar to Smolyak's sparse grid method, developed in Krueger & Kubler (2004). Both methods use **nonproduct rules for ameliorating costs** in high-dimensional applications
- Differ: cluster-grid is **endogenous** while Smolyak's grid is exogenous.
- Cluster-grid method typically has an **advantage (disadvantage) in accuracy** over Smolyak's method **inside (outside) the ergodic set** since cluster-grid fits a polynomial directly in the ergodic set.

Literature III

(3) **Projection method**

- Similar to perturbation methods in its ability to solve models with a **large number of agents**.
- Differ: This paper aims to be **accurate on the ergodic set**, whereas solutions produced by perturbation methods are accurate in a neighborhood of steady state, which is much smaller than the ergodic set.

  $\rightarrow$ Accuracy of this paper's global solutions does not decline rapidly away from steady state.

Cluster and Cluster Grid

To operate projection method on ergodic set, first need to construct a grid of points that covers the ergodic set.

- Simplest possible grid is obtained by simulation. Use either all or some of the simulated points as a grid.
- This paper advocates an efficient approach: replace the simulated points with a small number of appropriately chosen representative points.
- This is done by **clustering algorithms**.
- A set of obs are assigned into groups called **clusters**, so that obs belong to a same cluster are more similar than belong to different clusters.
- Obs in each cluster is represented with one point called **cluster grid**, computed as the average of all observations in the given cluster.

Before clustering, we should **measure distance** between observations and **transform variables** into comparable measurement units.

Distance Measure

Clustering algorithms require measuring **distance between observations**.

- Distance between observations i and j is measured in Euclidean (L^2 norm):

$$d_{ij} = \left[\sum_{l=1}^{L} (x_i^l - x_j^l)^2 \right]^{1/2} \tag{1}$$

where x_i^l is the i-th observation (object) on variable $l \in \{1, \cdots, L\}$.

Data Normalization

To transform variables into comparable measurement units, it is sufficient to **normalize variables to zero mean and unit variance**. Use a **principal components** (PCs) transformation.

- Let $X = (\mathbf{x}^1, \cdots, \mathbf{x}^L) \in \mathbb{R}^{T \times L}$ be a matrix of L variables which are normalized to zero mean and unit variance, where $\mathbb{R} \subseteq (-\infty, +\infty)$ and $x^l \in \mathbb{R}^{T \times 1}$.
- Consider the eigenvalue decomposition $X'X = V \Lambda V'$, where
  - $\Lambda \in \mathbb{R}^{L \times L}$: diagonal matrix with $\lambda^1 \geq \cdots \geq \lambda^L \geq 0$ being eigenvalues.
  - $V \in \mathbb{R}^{L \times L}$: orthogonal matrix of eigenvectors.
- Perform the linear transformation of X: $Z \equiv XV$, where $Z \in \mathbb{R}^{T \times L}$.
- Variables $\mathbf{z}^1, \cdots, \mathbf{z}^L$ are called **principal components** of X, and they are orthogonal (uncorrelated), i.e., $(\mathbf{z}^i)'\mathbf{z}^j = 0$ and $(\mathbf{z}^i)'\mathbf{z}^i = \lambda^i$ for any $i \neq j$.

Outline

1. Compiani & Kitamura (2016): Mixtures
2. The k-means clustering
3. Judd, Maliar & Maliar (2010): Clustering Algorithms
   - Overview
   - Hierarchical Algorithm
   - K-means Algorithm

Clustering Algorithms: Hierarchical Algorithm I

Hierarchical algorithm creates a multi-level hierarchy of clusters in the form of a tree.

- Root corresponds to a single cluster that includes all observations.
- leaves correspond to individual observations.
- Given a hierarchical tree, one can **choose the most appropriate level of clustering** for a given application.

Consider an **agglomerative type** of hierarchical algorithm which begins from individual observations (leaves) and agglomerates them into larger clusters.

1. Find the pairwise distances between the objects (observations) in data.
2. Merge the closest two clusters into a single cluster and continue to group the newly created clusters into larger clusters until there is a single cluster that contains all objects.
3. Cut off the obtained hierarchical tree at some point to obtain clusters.

Clustering Algorithms: Hierarchical Algorithm II

To group the newly created clusters into larger clusters, we should measure the distance between clusters.

Suppose i is in cluster A and j is in cluster B, then the distance between clusters A and B can be computed via:

- Single linkage (nearest neighbor) clustering: $d_{AB} = \min_{i \in A, j \in B} d_{ij}$.
- Complete linkage (furthest neighbor) clustering: $d_{AB} = \max_{i \in A, j \in B} d_{ij}$.
- Group average linkage clustering: $d_{AB} = \frac{1}{h_A h_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$, where h_K is number of objects in cluster K.
- Ward's linkage clustering: uses the increase in the sum of squared error (SSE), or variance: $d_{AB} = SSE_{AB} - (SSE_A + SSE_B)$, where $SSE_A \equiv \sum_{i \in A} \sum_{l=1}^{L} \left(x_i^l - \frac{1}{h_A} \sum_{i \in A} x_i^l \right)^2$.

If clusters are well-defined, then all the above linkage clustering algorithms produce similar cluster trees.

Clustering Algorithms: Hierarchical Algorithm III

The four types of clustering have different drawbacks:

- Single linkage clustering: suffer from **chaining**, meaning well separated clusters are joined together if there are outliers in between them.
- Complete linkage clustering: find clusters in which **obs are much closer to some obs of other clusters** than to some obs of their own cluster.
- Group average linkage clustering: results are **not invariant to monotone transformations** to d_{ij}.
- Ward's linkage clustering: deliver a spherical structure where such a **structure does not exist**.

Drawback about agglomerative hierarchical algorithm:

- Relatively **expensive in computational time and memory**.
- Decision about combining two clusters is final and **cannot be undone** on a later stage.

Outline

1. Compiani & Kitamura (2016): Mixtures
2. The k-means clustering
3. Judd, Maliar & Maliar (2010): Clustering Algorithms
   - Overview
   - Hierarchical Algorithm
   - K-means Algorithm

Clustering Algorithms: K-means Algorithm I

K-means clustering algorithm obtains a single partition of data instead of a cluster tree generated by a hierarchical algorithm.

Idea of K-means clustering: with K random clusters, moves objects between those clusters with the goal to **minimize variability within clusters** and to **maximize variability between clusters**. The algorithm proceeds as follows

1. Choose the number of clusters K.
2. Generate randomly K clusters and determine the centers of the clusters.
3. Given a current set of centers, assign each observation to the nearest cluster center and re-compute the centers of the new clusters.
4. Iterate on the last step until convergence.

Clustering Algorithms: K-means Algorithm II

A frequently used criterion function in the K-means algorithm is the SSE function,

$$SSE = \sum_{k=1}^{K} \sum_{i \in k} \sum_{l=1}^{L} \left(x_i^l - \bar{x}^{l,k} \right)^2, \quad \text{where}$$

- x_i^l is the l-th coordinate of a point in the k-th cluster.
- $\bar{x}^{l,k} \equiv \frac{1}{h_k} \sum_{i \in k} x_i^l$ is the l-th coordinate of a center of the k-th cluster.

Drawback about K-means algorithm:

- It can give different results with each run. This is because the K-means algorithm is sensitive to initial random assignments of obs into clusters.

Takeaway: Hierarchical Algorithm vs K-means Algorithm

For hierarchical algorithm:

- Different linkage clustering algorithms have their advantange and disadvantage.
- If clusters are well-defined, then different linkage clustering algorithms produce similar cluster trees.
- It is expensive in computational time and memory.

For K-means algorithm

- Relatively efficient and computationally cheap, making it suitable for large data sets.
- It can give different results with each run since it is sensitive to initial random assignments of observations into clusters.

References I

Compiani, G. and Kitamura, Y. (2016).
Using mixtures in econometric models: a brief review and some new results.
The Econometrics Journal, 19(3):C95–C127.

Judd, K. L., Maliar, L., and Maliar, S. (2010).
A cluster-grid projection method: Solving problems with high dimensionality.
NBER Working Paper, (w15965).