DLI Accelerated Data Science Teaching Kt

Lecture 15.2 - KMeans and Hierarchical Clustering

K-means Clustering

K-means is a partitional clustering algorithm

Let the set of data points (or instances) D be $\{x_1, x_2, \ldots, , \ldots\ x_i, x_n\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.

The k-means algorithm partitions the given data into k clusters.

- Each cluster has a cluster center, called centroid.
- k is specified by the user

Distance Functions

Key to clustering. “similarity” and “dissimilarity” can also commonly used terms.

There are numerous distance functions for

- Different types of data
- Different specific applications

Most commonly used functions are

- Euclidean distance and

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ir} - x_{jr})^2}$$

- Manhattan (city block) distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + ... + | x_{ir} - x_{jr} |$$

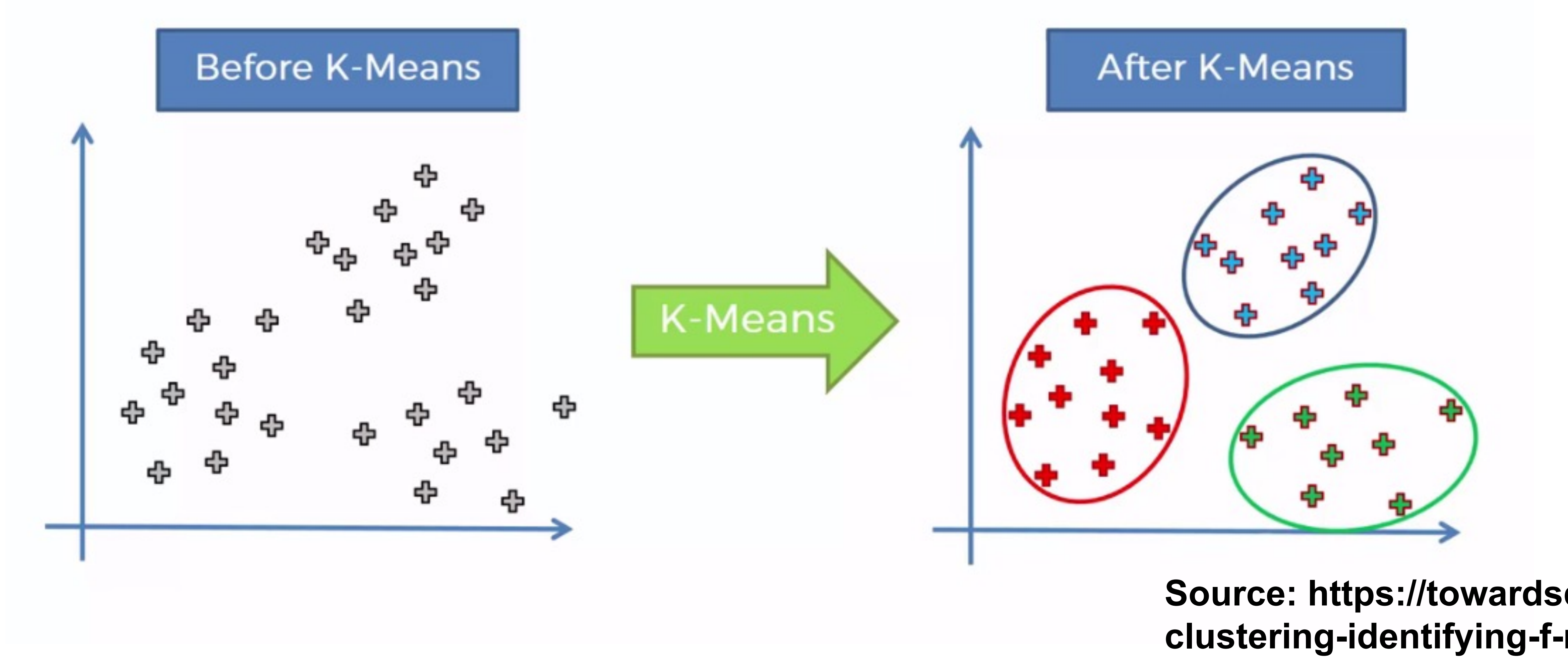
They are special cases of Minkowski distance. h is positive integer

$$dist(\mathbf{x}_i, \mathbf{x}_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + ... + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

Algorithm

Given k, the k-means algorithm works as follows:

1. Randomly choose k data points (seeds) to be the initial centroids, cluster centers
2. Assign each data point to the closest centroid
3. Re-compute the centroids using the current cluster memberships.
4. If a convergence criterion is not met, go to 2).



Source: https://towardsdatascience.com/k-means-clustering-identifying-f-r-i-e-n-d-s-in-the-world-of-strangers-695537505d

Stopping/Convergence Criterion

No (or minimum) re-assignments of data points to different clusters,

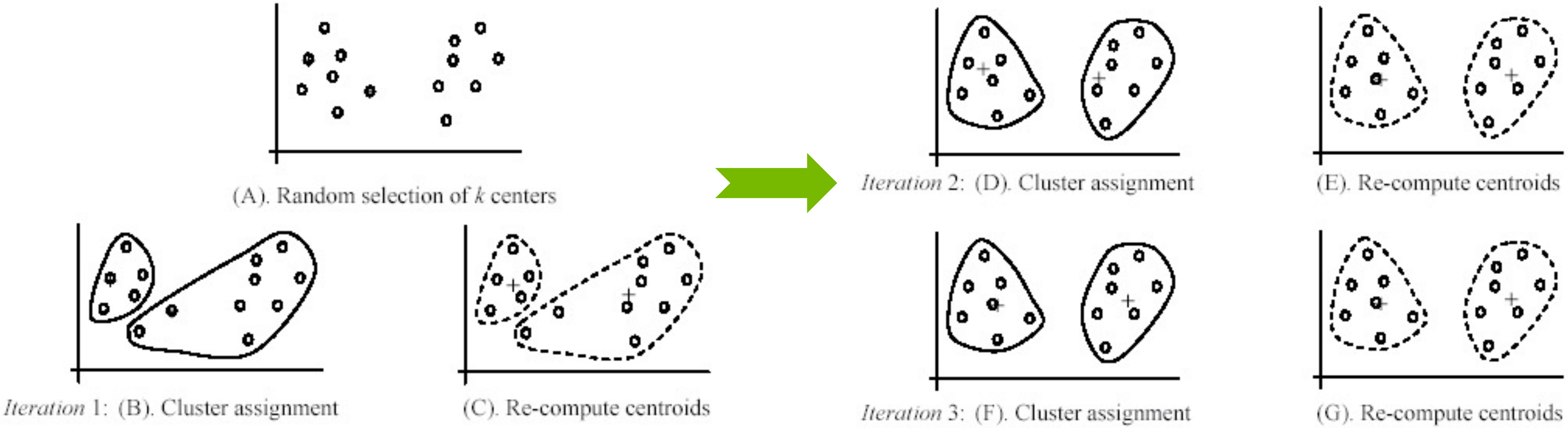
No (or minimum) change of centroids, or

Minimum decrease in the sum of squared error (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

- C_i is the j^{th} cluster, m_j is the centroid of cluster C_j (the mean vector of all the data points in C_j), and dist(x, m_j) is the distance between data point x and centroid m_j.

An Example of KMeans Clustering



Strengths and Weakness of KMeans

Strength

Simple: easy to understand and to implement

Efficient: Time complexity: O(tkn),

- where n is the number of data points,
- k is the number of clusters, and
- t is the number of iterations.

Since both k and t are small, k-means is considered a linear algorithm

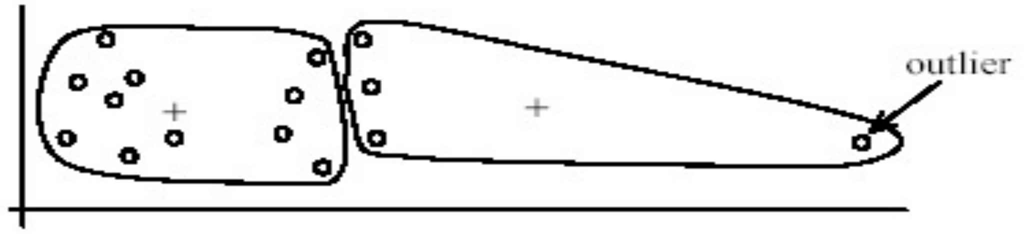
The most popular clustering algorithm

Weakness

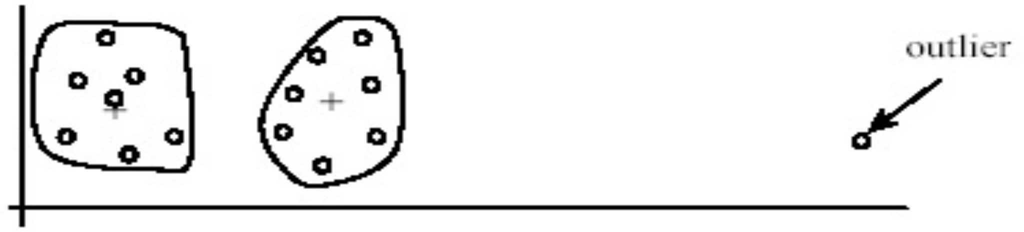
The user needs to specify k.

The algorithm is sensitive to outliers

- Outliers are data points that are very far away from other data points.
- Outliers could be errors in the data recording or some special data points with very different values.



(A): Undesirable clusters



(B): Ideal clusters

Strengths and Weakness of KMeans

Weakness

Processing outliers

Remove some data points in the clustering process that are much further away from the centroids than other data points.

- To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

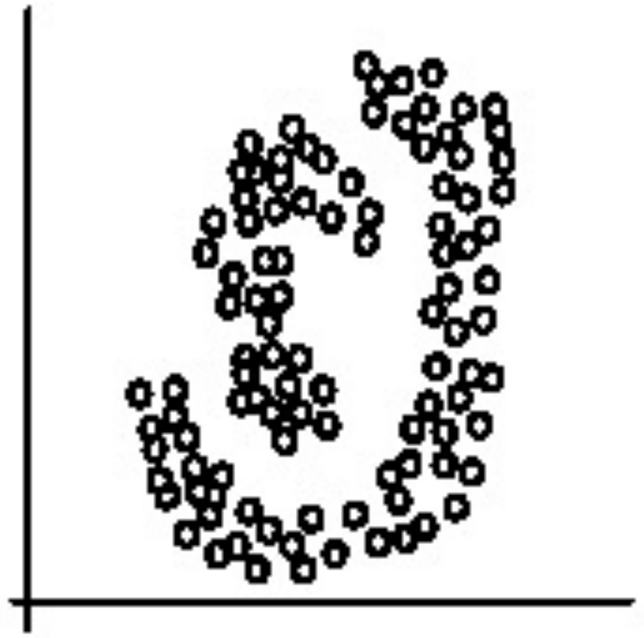
Perform random sampling.

- Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
- Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

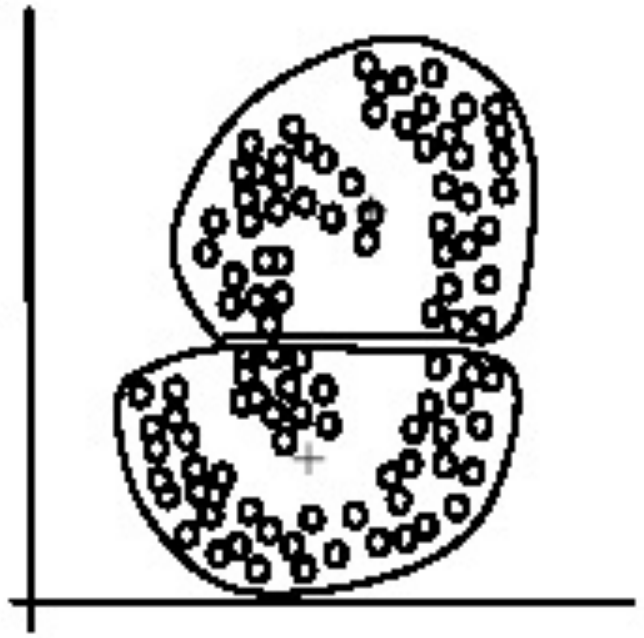
Strengths and Weakness of KMeans

Weakness

The k-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

KMeans Summary

Despite weaknesses, k-means is still the most popular algorithm due to its simplicity, efficiency.

No clear evidence that any other clustering algorithm performs better in general

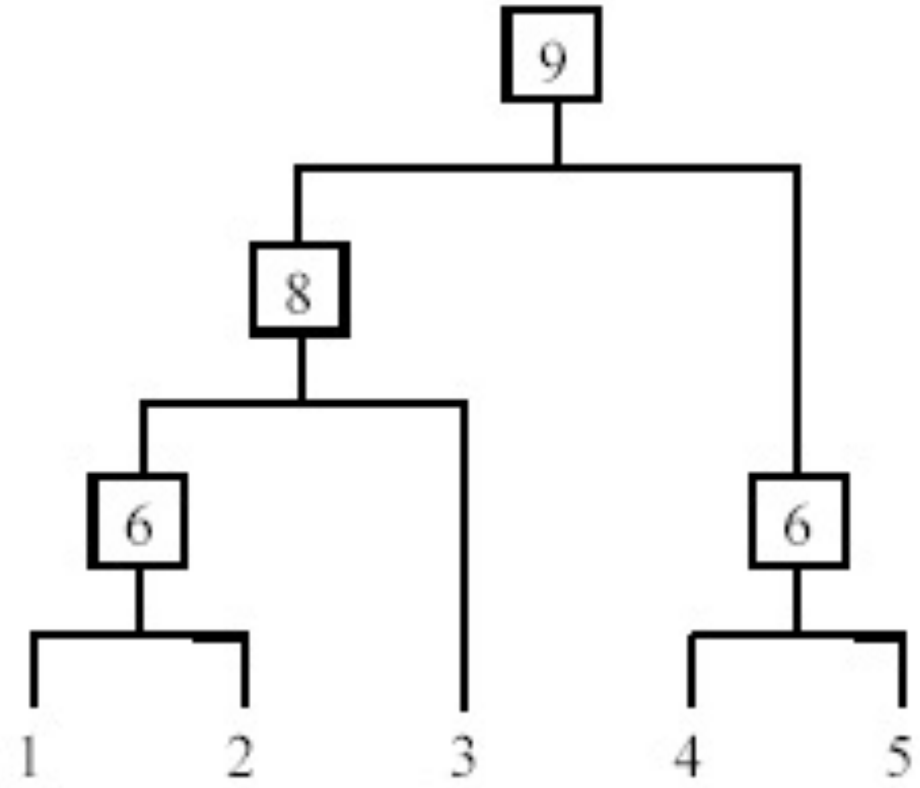
- although they may be more suitable for some specific types of data or applications.

Comparing different clustering algorithms is a difficult task.

- No one knows the correct clusters!

Hierarchical Clustering

Produce a nested sequence of clusters, a tree, also called Dendrogram.



Types of Hierarchical Clustering

Agglomerative (bottom up) clustering: It builds the tree from the bottom level, and

- Merges the most similar (or nearest) pair of clusters
- Stops when all the data points are merged into a single cluster (i.e., the root cluster).

Divisive (top down) clustering: It starts with all data points in one cluster, the root.

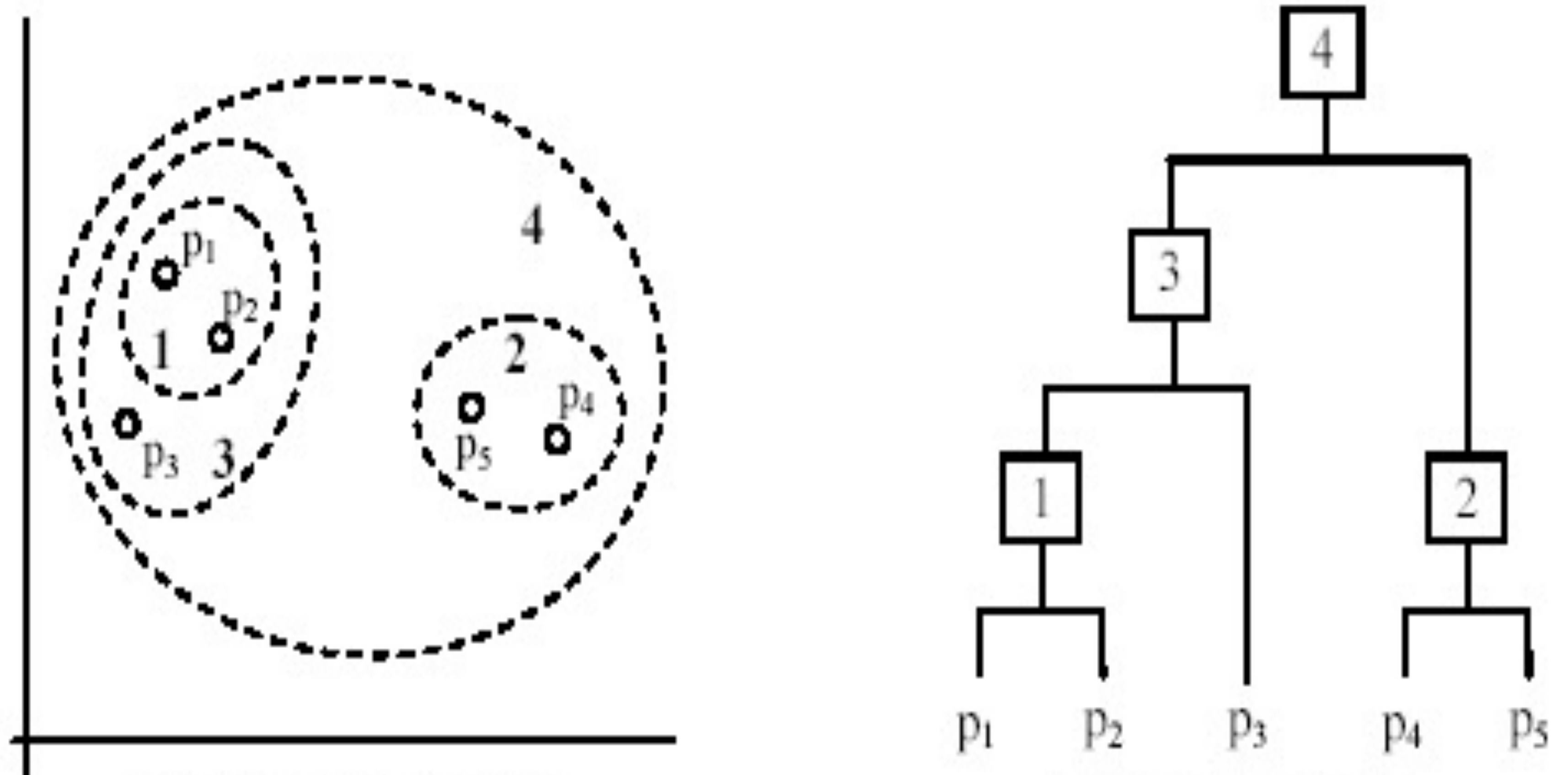
- Splits the root into a set of child clusters. Each child cluster is recursively divided further
- Stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

Agglomerative Clustering

It is more popular than divisive methods.

The basic ideas are below:

1. At the beginning, each data point forms a cluster (also called a node).
2. Merge nodes/clusters that have the least distance.
3. Go on merging
4. Eventually all nodes belong to one cluster



(A). Nested clusters (B) Dendrogram

Hard to Evaluate Clustering Performance

The quality of a clustering is very hard to evaluate since we do not know the correct clusters

Some methods are used:

- User inspection
- Study centroids
- For text documents, one can read some documents in clusters.

Summary

Clustering (Unsupervised Learning) has along history and is still active.

- There are a huge number of clustering algorithms
- More are still coming every year.

Clustering is hard to evaluate, but very useful in practice.

Clustering is highly application dependent and to some extent subjective.

DLI Accelerated Data Science Teaching Kit

Thank You