Alpaydin's Book: Ex. 7.10.7 Clustering

Machine Learning 2024-25 Course Activity

Furno Francesco - francesco.furno@studenti.unipd.it - 2139507

December 10, 2024

What are the similarities and differences between average-link clustering and k-means?

Average-link clustering, an hierarchical clustering method, and *k-means*, a partitional clustering method, are two techniques used in Unsupervised Learning to cluster data.

Similarities

Unsupervised Learning Methods

Average-link clustering and *k-means* are both Unsupervised Learning methods: they do not need labeled or classified data to perform clustering.

Clustering

Both *Average-link clustering* and *k-means* aim to group similar instances in the same cluster. An element of a cluster should have similar features to the elements of the same group and different features to the elements of the others group.

The use of distance

These two clustering methods use distance metrics to evaluate similarities and differences of data points and clusters. *Average-link clustering* measures the mean similarity between pairs of examples across clusters; *K-means* minimizes the average distance between examples and the *centroids*, the "center" of the cluster.

Differences

Type of Clustering

Average-link clustering is an hierarchical clustering algorithm. Hierarchical clustering uses two approaches:

- Agglomerative: at the beginning each instance is considered as a single cluster; then similar instances are merged in bigger clusters. The *Average-link clustering* algorithm uses this particular approach.
- Divisive: at the beginning all the instances are considered as one big cluster; then data is split into smaller clusters.

K-means, on the other hand, belongs to the family of partitional clustering algorithm, which divide the dataset into random partitions initially. The algorithm aims to minimize an objective function and iteratively adjusts partitions to optimize the clustering result.

Number of clusters
In the *Average-link clustering* algorithm, the number of clusters is not pre-specified. It generates a *dendrogram*, a tree diagram, where the number of clusters can be easily determined on a specifical level of cut of the tree.

To use *k-means* the number of clusters must be specified beforehand: this algorithm, in fact, finds a k clusters partition that optimizes a certain criterion given a set of instances and a k number.

Initialization & sensibility
The *Average-link clustering* does not require any initialization: it works directly on data, considering the full hierarchy during clustering,and it is very less sensitive to initialization issues.

The *k-means* algorithm requires the initialization of k *centroids*, which is usually done with random values. Different initializations can lead to different clustering results due to the algorithm's sensitivity to the starting positions of the centroids.

Output
The *Average-link clustering* algorithm generates a tree-like structure, a *dendrogram*, which shows nested clusters at various levels.

The *k-means* algorithm produces a flat partition of the dataset into k clusters.

Clusters shape
Average-link clustering can detect clusters of various shapes because it considers the relationships between all data points during the clustering process.

K-means assumes that clusters are spherical and equally sized: for this reason it is less effective for datasets with irregularly shaped clusters.

Computational cost
The *Average-link clustering* algorithm is computationally more expensive than *k-means* because it requires iterative calculations of distances between all points during the merging process. This makes it less suitable for large datasets.

K-means, on the other hand, is designed to be more efficient as it works with a fixed number of clusters and updates the positions of centroids at each iteration. This efficiency makes it preferable for larger datasets, but it may require more iterations to converge an optimal solution.

When to use them
The *Average-link clustering* algorithm should be used when the hierarchy is important or when the shape or the number of clusters is unknown. Due to its high computational costs, it is suitable for small datasets.

The *k-means* is suitable when the number of clusters is known or can be estimated. It is the right choice if clusters have a spherical shape and the dataset has low noise.