K-Means Clustering

ANANDHU S

Abstract

Clustering is an automatic learning technique aimed at grouping a set of objects into subsets or clusters. The goal is to create clusters that are coherent internally, but substantially different from each other. In plain words, objects in the same cluster should be as similar as possible, whereas objects in one cluster should be as dissimilar as possible from objects in the other clusters.

K-means clustering is a widely used unsupervised machine learning algorithm that partitions a dataset into a predefined number of clusters, k, based on similarity. The algorithm operates by initializing k centroids and assigning each data point to the nearest centroid, forming clusters. It iteratively recalculates the centroids by minimizing the within-cluster variance, thereby optimizing the cluster configurations. K-means is computationally efficient and suitable for large datasets but may face challenges such as sensitivity to the choice of k, initial centroid placement, and susceptibility to noise and outliers. Despite these limitations, it remains an effective tool for various applications, including market segmentation, image compression, and anomaly detection.

K-Means

K-means is the most important flat clustering algorithm. The objective function of K- means is to minimize the average squared distance of objects from their cluster centers, where a cluster center is defined as the mean or centroid μ of the objects in a cluster C:

$$\vec{\mu}(\mathrm{C}) = \frac{1}{|C|}\sum_{\vec{x}\in C}\vec{x}$$

The ideal cluster in K-means is a sphere with the centroid as its center of gravity. Ideally, the clusters should not overlap. A measure of how well the centroids represent the members of their clusters is the Residual Sum of Squares (RSS), the squared distance of each vector from its centroid summed over all vectors

$$\mathrm{RSS_i} = \sum_{\vec{x}\in C_i} || \vec{x} - \vec{\mu}(C_i) ||^2$$

$$\mathrm{RSS} = \sum_{i=1}^{K} RSS_i$$

K-means can start with selecting as initial clusters centers K randomly chosen objects, namely the seeds. It then moves the cluster centers around in space in order to minimize RSS. This is done iteratively by repeating two steps until a stopping criterion is met

1. reassigning objects to the cluster with closest centroid
2. recomputing each centroid based on the current members of its cluster.

We can use one of the following termination conditions as stopping criterion

- A fixed number of iterations I has been completed.
- Centroids μ_i do not change between iterations.
- Terminate when RSS falls below a pre-estabilished threshold.

Algorithm for K-Means

```
procedure KMEANS(X,K)
    {s1, s2, · · · , sk}   SelectRandomSeeds(K,X)
    for i ←1,K do
        $\mu(Ci) \leftarrow si$
    end for
    repeat
        $\min_{k\sim x_n - -\sim\mu(C_k)k} C_k = C_k [ \{\sim x_n\}$
        for all $C_k$ do
            $\mu(C_k) = 1$
        end for
    until stopping criterion is met
end procedure
```

Merits of K-Means Clustering:

1. **Simplicity and Efficiency**: K-means is easy to implement and computationally efficient, especially for large datasets. Its time complexity is linear with respect to the number of data points and clusters.
2. **Scalability**: It can handle large datasets efficiently, making it suitable for big data applications.
3. **Quick Convergence**: In most cases, K-means converges quickly, often within a few iterations, especially if centroids are initialized well.
4. **Interpretability**: The clusters formed are easy to interpret, as they are defined by the centroids and the distance metric used (typically Euclidean distance).
5. **Use Cases**: It works well for spherical and well-separated clusters, and it's commonly applied in tasks like market segmentation, document clustering, and image compression.

Demerits of K-Means Clustering:

1. **Choice of K**: The number of clusters (k) must be predefined, and it is not always clear how many clusters are appropriate for a given dataset.
2. **Sensitivity to Initial Centroids**: The initial placement of centroids can significantly affect the final results. Poor initialization can lead to suboptimal clusters.
3. **Assumes Spherical Clusters**: K-means works best when clusters are spherical or roughly of equal size. It struggles with clusters that have non-circular shapes or varying densities.
4. **Sensitive to Outliers**: K-means is sensitive to noise and outliers, as they can distort the cluster formation by pulling centroids towards them.
5. **Convergence to Local Minima**: The algorithm can converge to a local minimum rather than the global optimum, depending on the initial centroid selection.

6. **Distance Metric Dependency**: The performance is heavily dependent on the choice of the distance metric (typically Euclidean distance), which may not work well with non-linearly separable data.