K-means Clustering Algorithm

HIMANSHU KESARVANI

06/10/2024

1 Introduction

K-means is a popular unsupervised learning algorithm used for clustering, where the aim is to partition a set of n data points into k clusters. Each data point belongs to the cluster with the nearest mean, serving as the cluster's centroid. K-means minimizes the within-cluster sum of squares (WCSS), which is also known as the inertia or variance.

2 Mathematical Formulation

Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of n data points, where each data point x_i is a d-dimensional vector: $x_i \in \mathbb{R}^d$. The goal of K-means is to divide X into k clusters, denoted as $C_1, C_2, \dots, C_k$, such that the following objective function is minimized:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Here, μ_i represents the centroid of cluster C_i, and $\|x - \mu_i\|^2$ is the squared Euclidean distance between a data point x and the centroid μ_i.

The K-means algorithm iteratively updates the centroids and assigns data points to clusters based on the following steps.

3 Steps of the K-means Algorithm

3.1 Step 1: Initialization

Randomly initialize k centroids $\mu_1, \mu_2, \dots, \mu_k$. These centroids can be initialized randomly from the data points or by using methods such as the K-means++ algorithm, which improves the initialization to avoid poor clustering.

3.2 Step 2: Assign Data Points to Clusters

For each data point x_j, assign it to the cluster with the nearest centroid. Mathematically, data point x_j is assigned to cluster C_i if:

$$C_i = \{x_j : \|x_j - \mu_i\|^2 \leq \|x_j - \mu_l\|^2 \text{ for all } l = 1, \ldots, k\}$$

This step minimizes the distance between each data point and its closest centroid.

3.3 Step 3: Update Centroids

After assigning all data points to clusters, recompute the centroid of each cluster as the mean of the points in that cluster:

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

where $|C_i|$ is the number of points in cluster C_i.

3.4 Step 4: Convergence Check

Repeat Steps 2 and 3 until convergence. Convergence occurs when the centroids no longer change significantly or when the assignments of data points to clusters remain unchanged. Formally, you can stop the algorithm when:

$$\sum_{i=1}^{k} \|\mu_i^{(t)} - \mu_i^{(t-1)}\|^2 < \epsilon$$

where ϵ is a small threshold and $\mu_i^{(t)}$ and $\mu_i^{(t-1)}$ represent the centroids at iteration t and $t-1$, respectively.

4 Complexity and Limitations

The time complexity of the K-means algorithm is $O(n \cdot k \cdot d \cdot t)$, where:

- n is the number of data points,
- k is the number of clusters,
- d is the dimensionality of each data point,
- t is the number of iterations until convergence.

4.1 Limitations

- K-means is sensitive to the initial choice of centroids. Poor initialization can lead to suboptimal clustering.
- The algorithm assumes clusters to be spherical and equally sized, which might not be the case for real-world data.
- K-means requires the number of clusters, k, to be predefined.
- The algorithm converges to a local minimum, which might not be the global minimum.

5 Conclusion

The K-means algorithm is a simple yet powerful clustering technique that groups data into k clusters based on the nearest centroid. Although it has limitations such as sensitivity to initialization and the assumption of spherical clusters, it remains widely used due to its efficiency and ease of implementation. Proper initialization techniques, such as K-means++, can mitigate some of the algorithm's weaknesses.

K-means Number of Cluster Finding Method

The determination of the optimal number of clusters in **K-Means** is a challenging problem because the algorithm requires the user to specify (k), the number of clusters, in advance. There are various methods to estimate (k) based on mathematical and statistical principles. Here are some of the most common techniques used:

Elbow Method

The **elbow method** is one of the most widely used techniques for determining the optimal number of clusters.

Steps:

- Run K-Means clustering for a range of (k) values (e.g., ($k = 1$) to ($k = 10$)).
- For each (k), calculate the **Within-Cluster Sum of Squares (WCSS)** or **inertia**, which measures the compactness of the clusters. This is the sum of squared distances between each data point and its assigned cluster centroid.

The formula for WCSS for a single cluster (j) is:

$$\text{WCSS}_j = \sum_{i \in C_j} ||x_i - \mu_j||^2$$

Where: - (x_i) is a data point in cluster (C_j), - (μ_j) is the centroid of cluster (C_j), - ($||x_i - \mu_j||^2$) is the Euclidean distance between the data point and the cluster centroid.

Summing over all clusters gives the total WCSS:

$$\text{WCSS}_{\text{total}} = \sum_{j=1}^{k} \text{WCSS}_j$$

- Plot the WCSS against (k) values. As (k) increases, WCSS will decrease because the clusters become smaller and tighter. The idea is to find the point where adding more clusters no longer significantly improves the WCSS. This point is known as the **elbow**, and it indicates the optimal number of clusters.

Mathematical Interpretation:

When (k) increases, the decrease in WCSS becomes less significant. The ”elbow” point is where the rate of decrease sharply changes, indicating the point of diminishing returns.

Silhouette Score

The **silhouette score** measures how similar each point is to its own cluster (cohesion) compared to other clusters (separation).

For each point (i), the silhouette coefficient ($s(i)$) is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where: - ($a(i)$) is the average distance between point (i) and all other points in its own cluster. - ($b(i)$) is the average distance between point (i) and points in the nearest neighboring cluster.

- ($s(i)$) ranges from -1 to 1. A value close to 1 means the point is well clustered, while a value near -1 indicates that the point is misclassified.

To find the optimal (k), compute the silhouette score for different values of (k) and choose the (k) that maximizes the average silhouette score.

Mathematical Interpretation:

The silhouette score quantifies the separation between clusters. A higher silhouette score suggests that the clusters are well separated and that the points are well matched to their own cluster.

Gap Statistic Method

The **Gap Statistic** compares the WCSS of the K-Means solution with the expected WCSS under a null reference distribution (randomly distributed data points). This helps in understanding how much better the clustering is compared to random noise.

Steps:

- For each (k), compute the WCSS for the clustering solution (WCSS_k). - Generate multiple random datasets (following a uniform distribution over the data's bounding box) and compute the WCSS for these random datasets. - The gap statistic is then defined as:

$$\text{Gap}(k) = \frac{1}{B} \sum_{b=1}^{B} \log(\text{WCSS}^b) - \log(\text{WCSS}_k)$$

Where (B) is the number of bootstrapped random datasets, (WCSS^b) is the WCSS of the random dataset (b), and (WCSS_k) is the WCSS for the real data with (k) clusters.

- Choose the smallest (k) such that:

$$\text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$$

Where (s_{k+1}) is the standard deviation of the bootstrapped WCSS estimates.

Mathematical Interpretation:

The gap statistic measures how much better the K-Means clustering performs compared to random clustering. The optimal (k) is where the gap statistic is maximized.

Davies-Bouldin Index

The **Davies-Bouldin Index** (DBI) measures the average similarity ratio of each cluster with its most similar cluster. A lower DBI indicates better clustering.

The formula for the DBI for cluster (i) is:

$$\text{DBI} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{d(\mu_i,\mu_j)}\right)$$

Where: - (σ_i) is the average distance between points in cluster (i) and the centroid (μ_i), - (μ_i) and (μ_j) are the centroids of clusters (i) and (j), - $(d(\mu_i,\mu_j))$ is the distance between the centroids of clusters (i) and (j).

- A lower DBI indicates that clusters are compact and well-separated. You can compute the DBI for various values of (k) and choose the one with the lowest index.

BIC (Bayesian Information Criterion) or AIC (Akaike Information Criterion)

These are model selection criteria typically used in mixture models, but they can also be applied to K-Means to estimate the optimal (k).

Bayesian Information Criterion (BIC):

BIC penalizes the likelihood function based on the number of parameters in the model. It is defined as:

$$\text{BIC}(k) = \log(n)\cdot k - 2\cdot\log(L)$$

Where: - (n) is the number of data points, - (k) is the number of clusters, - (L) is the likelihood of the data given the clustering.

5.1 Akaike Information Criterion (AIC):

AIC is another information criterion that adds a penalty for the number of clusters (k):

$$\text{AIC}(k) = 2\cdot k - 2\cdot\log(L)$$

For both BIC and AIC, lower values indicate a better balance between goodness of fit and model complexity.

Conclusion:

- **Elbow Method** is a simple and intuitive method, especially when plotting WCSS vs. (k).
- **Silhouette Score** provides a measure of how well-separated clusters are.
- **Gap Statistic** compares clustering performance with random data.
- **Davies-Bouldin Index** minimizes inter-cluster similarity, indicating well-separated clusters.
- **BIC/AIC** help in balancing model fit and complexity.

Each method has its strengths and weaknesses, and in practice, it's often useful to try multiple methods to validate the choice of (k).