Exam Review, K-means Clustering

Rebecca C. Steorts, Duke University

STA 325

k-means clustering

K-means clustering is a popular unsupervised machine learning algorithm used to partition a data set into K distinct, non-overlapping clusters.

The goal is to group data points such that the data points within each cluster are as similar as possible, while the clusters themselves are as distinct as possible.

k-means algorithm

1. Initialize.

- Choose the number of clusters, K.
- Then, randomly select K initial cluster centroids (the center points of the clusters).

2. Assign clusters.

- For each data point, assign it to the nearest centroid based on a distance metric, such as the Euclidean distance.

3. Update centroids.

- After assigning the data points to clusters, re-calculate the centroids by taking the mean of the data points in each cluster.

4. Repeat: Repeat steps 2 and 3 until convergence.

Strengths and Weaknesses

Strengths:

- Simple and fast.
- Works well when clusters are roughly spherical and well-separated.

Weaknesses

- Requires the number of clusters to be pre-defined.
- Sensitive to the initial choice of centroids (can converge to local minima).
- Struggles with clusters of different shapes.

Notation and Within-Cluster Sum of Squares

Assume

- K is the number of clusters,
- C_k is the set of points assigned to cluster k,
- μ_k is the centroid of cluster k,
- x_i is a data point assigned to cluster k.

The algorithm minimizes the within-cluster sum of squares (WCSS), which is the sum of the squared Euclidean distances between each data point and its corresponding centroid:

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2.$$

Algorithm

1. Initialize. Choose K initial centroids. Ex: Randomly.
2. Assignment. For each data point x_i, assign it to the nearest centroid.

   This assignment is based on the distance to the centroids, typically using the Euclidean distance:

   $$c_i = \arg\min_k \|x_i - \mu_k\|,$$

   where c_i is the index of the centroid closest to point x_i, and μ_k is the centroid of cluster k.

Algorithm (continued)

3. Update. After all data points are assigned to clusters, update the centroids. The new centroid for cluster k is the mean of all data points assigned to it:

   $$\mu_k = \frac{1}{N_k} \sum_{x_i \in C_k} x_i$$

   where N_k is the number of points in cluster k, and C_k is the set of points assigned to cluster k.
4. Repeat. Iterate between 2 and 3 until convergence.