K-Means Clustering from Scratch

Bhanu Prasanna Koppolu

Table of contents

Source Code

Full implementation available at ML_from_scratch/K-Means

K-Means Clustering

K-Means is a clustering algorithm which comes under **Unsupervised** rather than Supervised. We only have the Data X which has m data points where each data point is represented as x_i.

Objective

The main objective of K-Means is to form K clusters where the points inside have the lowest Cost J.

The cost is nothing but the **sum of Squared Euclidean Distance** of all the data points in a cluster with respect to the center of the cluster.

The Algorithm

1. First we randomly take K points from the Data X to be our Center points for our K clusters
2. **Assignment operation**: assign each data point x_i to its respective k-th cluster based on the calculated distance
3. After assignment, compute the mean of all points inside the cluster to find the **new center** for cluster k
4. Calculate cost J. If it converges and is less than our threshold, the model has successfully been fit
5. Steps 1-4 are repeated for a certain number of iterations or until convergence

Mathematical Formulation

We have: - Data points x_i where $i = 1$ to m - Cluster centers μ_k where $k = 1$ to K - r_{ik} - indicator function that tells if the i-th data point is in k-th cluster (1 if yes, 0 if no)

Squared Euclidean Distance:

$$(x_i - \mu_k)^2$$

Cost Function:

$$J(\{r_{ik}\}, \{\mu_k\}) = \sum_{i=1}^{m} \sum_{k=1}^{K} r_{ik}(x_i - \mu_k)^2$$

Two Main Steps

Assignment Step (Given Centers μ_k, Find best assignments r_{ik})

We have to choose r_{ik} for each i: - $r_{ik} \in \{0, 1\}$ - $\sum_{k=1}^{K} r_{ik} = 1$

For each point x_i:

1. Compute distances to each centroid: $(x_i - \mu_k)^2$ for all k
2. Assign the point to the closest centroid:

$$r_{ik} = 1 \text{ if } k = \arg\min_j (x_i - \mu_j)^2, \quad r_{ij} = 0 \text{ for } j \neq k$$

Update Step

For each cluster k, recompute centroid as the mean of its assigned points:

$$\mu_k = \frac{\sum_{i=1}^{m} r_{ik} \cdot x_i}{\sum_{i=1}^{m} r_{ik}} = \frac{1}{n_k} \sum_{\substack{i=1 \\ r_{ik}=1}}^{m} x_i$$

Convergence

Stop when: - Assignments no longer change - Centroids move less than a certain tiny threshold - After some max iterations

The algorithm converges to a **local minimum** of J.

Implementation

```python
class kmeans:
    """K-Means clustering algorithm.

    Partitions data into K clusters by iteratively:
    1. Assigning points to nearest centroid
    2. Updating centroids to cluster means
    """

    def __init__(self,
                 k=3,                # Number of clusters
                 iterations=1000,  # Max iterations
                 threshold=1e-3):  # Convergence threshold for cost change
        self.k = k
        self.iterations = iterations
        self.threshold = threshold

    def fit(self, X_train):
        """Fit K-Means to training data.

        Algorithm:
        1. Initialize k centroids randomly from data points
        2. For each iteration:
           - Compute distances from all points to all centroids
           - Assign each point to closest centroid
           - Update centroid as mean of assigned points
           - Handle empty clusters by reinitializing randomly
           - Check convergence (cost change < threshold)

        Returns:
            k_points: Final cluster centroids (k, n_features)
        """
        ...

    def predict(self, X_test):
        """Predict cluster labels for new data.

        Assigns each point to the nearest centroid based on
        squared Euclidean distance.

        Returns:
            cluster_labels: Cluster assignment for each point
        """
```

```
...
```