CHAPTER 8

k-Means

> "k-Means is a clustering algorithm that partitions data into k clusters by minimizing the sum of squared distances between data points and their respective cluster centroids."

8.1 Introduction

Clustering is a fundamental task in unsupervised learning that aims to partition data into meaningful groups. Among the various clustering algorithms, k-Means has remained popular due to its simplicity and efficiency. Introduced by MacQueen in 1967 [1], k-Means has been extensively studied and applied in fields ranging from image segmentation to customer segmentation. Our discussion is supported by seminal works in the field [1, 2].

8.2 Intuitive Explanation (Without Math)

8.2.1 Intuition

k-Means partitions a dataset into k clusters by iteratively assigning each data point to the nearest cluster center and then updating the cluster centers as the mean of the points assigned to them.

8.2.2 Step-by-Step Example

1. **Initialization:** Choose k initial centroids. For example, in a 2D dataset, these might be randomly selected points.
2. **Assignment:** Each data point is assigned to the cluster corresponding to the closest centroid. Imagine a dataset containing two well-separated groups; points are "pulled" toward the nearest centroid.

3. **Update:** After assignment, new centroids are computed as the mean of all points in each cluster.

4. **Iteration:** Steps 2 and 3 are repeated until the centroids stabilize (i.e., further iterations yield negligible changes in centroid positions).

8.2.3 Application Examples

Imagine a set of customer data where each customer is represented by features such as age and annual income. k-Means can automatically group these customers into clusters that share similar characteristics. For instance:

- **Image Segmentation:** Grouping pixels based on color intensity.
- **Customer Segmentation:** Dividing customers into distinct groups for targeted marketing.

8.2.4 Graphical Illustration

Figure 8.1 provides a visual representation where data points are partitioned into different clusters with distinct colors, and the cluster centers are marked by larger symbols. This illustration was generated by the Matlab code that is provided in the following section.

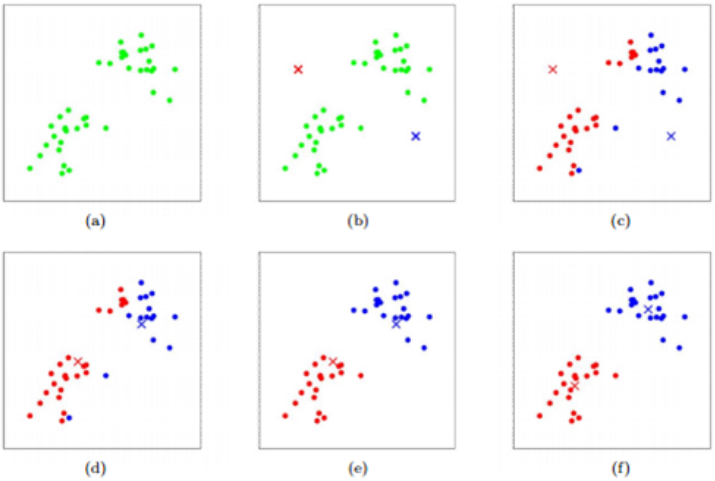


Figure 8.1: An illustration of k-Means clustering steps: a) input data, b) class initialization (red and blue "x"), c) initial classification of clusters, d) & e) iterations to reduce error in classification, f) final classification.

8.3 Tutorial: Implementation From Scratch

In this section, we provide a practical step-by-step tutorial for implementing a simple k-Means algorithm. First, we present pseudo-code and then a MATLAB implementation from scratch with graphical illustrations. Complete – from-scratch – Matlab, Python, and R implementations are available on the companion website.[1].

8.3.1 Pseudo-Code

Algorithm 1 is the pseudo-code implementation of the steps and equations defined in Section 8.4.

Algorithm 1 k-Means Clustering Algorithm

Input: Dataset $\mathbf{X}$, number of clusters k, maximum iterations T
Initialize: Randomly choose k points as initial centroids $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$
for iteration = 1 to T **do**
    for each data point $\mathbf{x} \in \mathbf{X}$ **do**
        Assign $\mathbf{x}$ to the cluster with the nearest centroid:

$$\text{Cluster}(\mathbf{x}) = \arg\min_{1 \leq i \leq k} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

    end for
    for each cluster $\mathbf{C}_i$, $i = 1, \ldots, k$ **do**
        Update the centroid:

$$\boldsymbol{\mu}_i = \frac{1}{|\mathbf{C}_i|} \sum_{\mathbf{x} \in \mathbf{C}_i} \mathbf{x}$$

    end for
    Check for convergence (if centroids do not change, break)
end for
Output: Clusters $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k\}$ and centroids $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$

8.3.2 Code Implementation

The code below is a simple k-Means algorithm that is a direct implementation of the steps outlined in Algorithm 1. The code for this section is extensive so we will only cover the key elements. For brevity, we use the "$\cdots$" to denote the omission of some code.

[1] `https://pfroysdon.github.io/publications/`

In this example, the code comments provide the necessary information for the `kMeans()` function.

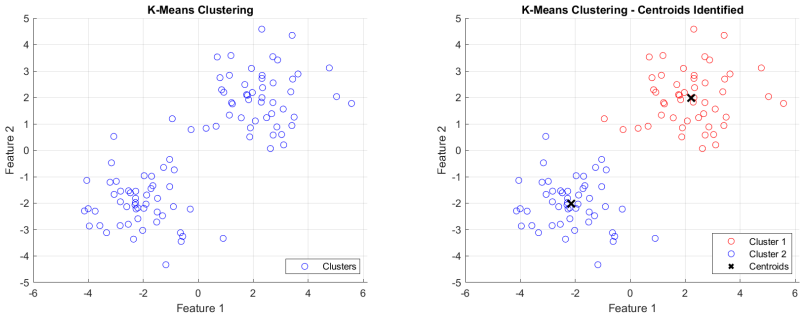
```
function [idx, centroids] = kMeans(X, k, maxIter)
% Randomly initialize centroids by selecting k unique
% data points
rng(1); % For reproducibility
[m, ~] = size(X);
initIdx = randperm(m, k);
centroids = X(initIdx, :);
idx = zeros(m, 1);
for iter = 1:maxIter

  % Assignment Step: assign each point to the nearest
  % centroid
  for i = 1:m
    distances = sum((centroids - X(i,:)).^2, 2);
    [~, idx(i)] = min(distances);
  end

  % Update Step: recompute centroids
  newCentroids = centroids;
  for j = 1:k
    clusterPoints = X(idx == j, :);
    if ~isempty(clusterPoints)
      newCentroids(j, :) = mean(clusterPoints, 1);
    end
  end

  % Check for convergence
  if norm(newCentroids - centroids, 'fro') < 1e-6
    break;
  end
centroids = newCentroids;
end

end
```

The final results are shown in Figures 8.2a and 8.2b, and the complete code (in Matlab, Python and R) is available on the companion website.

8.4 Mathematical Derivations & Proofs

8.4.1 Introduction

k-Means is an unsupervised clustering algorithm that partitions a set of data points into K clusters. The algorithm seeks to minimize the within-cluster sum-of-squares error by iteratively assigning each data point to the nearest cluster center and then updating each cluster center to be the mean of the points assigned to it. In this section, we provide the mathematical



(a) (b)

Figure 8.2: An illustration of k-means clustering: (a) data points are grouped into clusters, (b) after several iterations their centroids are identified with large "X" markers.

formulation of k-Means, derive the update steps, and explain the algorithm in detail. All variables are declared with their dimensions and properties.

8.4.2 Data and Notation

Assume we are given a dataset of n samples:

$$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\},$$

where:

- $\mathbf{x}_i \in \mathbb{R}^d$ is the ith data point (a column vector of dimension $d \times 1$).
- n is the number of data points.
- d is the number of features (dimensions) of each data point.

We wish to partition these n points into K clusters. The following variables are introduced:

- $\mathbf{c}_k \in \mathbb{R}^d$ for $k = 1, 2, \ldots, K$ denotes the center (centroid) of cluster k.
- K is the number of clusters.
- r_{ik} is a binary indicator variable defined as:

$$r_{ik} = \begin{cases} 1, & \text{if data point } \mathbf{x}_i \text{ is assigned to cluster } k, \\ 0, & \text{otherwise.} \end{cases}$$

- For each data point $\mathbf{x}_i$, we have the constraint:

$$\sum_{k=1}^{K} r_{ik} = 1, \quad \text{for } i = 1, 2, \ldots, n.$$

8.4.3 k-Means Objective Function

The goal of k-Means is to minimize the total within-cluster sum-of-squares error. The objective function J is given by:

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \left\| \mathbf{x}_i - \mathbf{c}_k \right\|^2,$$

where:

- $\|\mathbf{x}_i - \mathbf{c}_k\|^2$ denotes the squared Euclidean distance between data point $\mathbf{x}_i$ and cluster center $\mathbf{c}_k$.
- $J \in \mathbb{R}_{\geq 0}$ is a scalar.

The squared Euclidean distance is defined as:

$$\|\mathbf{x}_i - \mathbf{c}_k\|^2 = (\mathbf{x}_i - \mathbf{c}_k)^\top (\mathbf{x}_i - \mathbf{c}_k) = \sum_{j=1}^{d} (x_{ij} - c_{kj})^2,$$

where x_{ij} and c_{kj} denote the jth components of $\mathbf{x}_i$ and $\mathbf{c}_k$ respectively.

8.4.4 Derivation of the k-Means Algorithm Steps

k-Means is typically solved via an iterative algorithm (often called Lloyd's algorithm) that alternates between two steps:

1. **Assignment Step:** Given the current cluster centers, assign each data point to its nearest center.
2. **Update Step:** Given the assignments, update each cluster center as the mean of the points assigned to that cluster.

8.4.4.1 Assignment Step

For each data point $\mathbf{x}_i$, determine the index k that minimizes the squared Euclidean distance:

$$r_{ik} = \begin{cases} 1, & \text{if } k = \arg\min_{j \in \{1,\ldots,K\}} \|\mathbf{x}_i - \mathbf{c}_j\|^2, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, the optimal assignment for $\mathbf{x}_i$ is:

$$\hat{k}_i = \arg\min_k \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$

This step minimizes the objective function J with respect to the indicator variables r_{ik} while keeping the cluster centers fixed.

8.4.4.2 Update Step

For a fixed assignment $\{r_{ik}\}$, we wish to update the cluster centers $\mathbf{c}_k$ to minimize the objective J. For each cluster k, the portion of the objective function corresponding to $\mathbf{c}_k$ is:

$$J_k = \sum_{i=1}^{n} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$

We now derive the optimal $\mathbf{c}_k$ that minimizes J_k.

To do so, we differentiate J_k with respect to $\mathbf{c}_k$ and set the derivative equal to zero. Note that

$$\|\mathbf{x}_i - \mathbf{c}_k\|^2 = (\mathbf{x}_i - \mathbf{c}_k)^\top (\mathbf{x}_i - \mathbf{c}_k).$$

Differentiate with respect to $\mathbf{c}_k$:

$$\frac{\partial}{\partial \mathbf{c}_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2 = -2(\mathbf{x}_i - \mathbf{c}_k).$$

Thus,

$$\frac{\partial J_k}{\partial \mathbf{c}_k} = \sum_{i=1}^{n} r_{ik} \, [-2(\mathbf{x}_i - \mathbf{c}_k)] = -2 \sum_{i=1}^{n} r_{ik} (\mathbf{x}_i - \mathbf{c}_k).$$

Setting the derivative to zero for optimality:

$$-2 \sum_{i=1}^{n} r_{ik} (\mathbf{x}_i - \mathbf{c}_k) = \mathbf{0}.$$

This implies:

$$\sum_{i=1}^{n} r_{ik} \mathbf{x}_i = \sum_{i=1}^{n} r_{ik} \mathbf{c}_k.$$

Since $\mathbf{c}_k$ is constant with respect to i, we have:

$$\sum_{i=1}^{n} r_{ik} \mathbf{x}_i = \mathbf{c}_k \sum_{i=1}^{n} r_{ik}.$$

Let

$$n_k = \sum_{i=1}^{n} r_{ik},$$

which is the number of points assigned to cluster k. Then the optimal update is:

$$\mathbf{c}_k = \frac{1}{n_k} \sum_{i=1}^{n} r_{ik}\, \mathbf{x}_i.$$

Dimensions:

- Each $\mathbf{c}_k \in \mathbb{R}^d$.
- Each $\mathbf{x}_i \in \mathbb{R}^d$.
- n_k is a scalar (nonnegative integer) representing the number of points in cluster k.

8.4.5 Summary of the k-Means Algorithm

The k-Means algorithm proceeds iteratively as follows:

1. **Initialization:** Choose initial cluster centers $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K \in \mathbb{R}^d$ (e.g., randomly select K data points).
2. **Repeat Until Convergence:**
   (a) **Assignment Step:** For each data point $\mathbf{x}_i$ (for $i = 1, \ldots, n$), assign it to the cluster whose center is closest:

   $$\hat{k}_i = \arg \min_{k \in \{1,\ldots,K\}} \|\mathbf{x}_i - \mathbf{c}_k\|^2.$$

   Equivalently, set

   $$r_{ik} = \begin{cases} 1, & \text{if } k = \hat{k}_i, \\ 0, & \text{otherwise.} \end{cases}$$

   (b) **Update Step:** For each cluster $k \in \{1, \ldots, K\}$, update the cluster center as the mean of all points assigned to it:

   $$\mathbf{c}_k = \frac{1}{n_k} \sum_{i=1}^{n} r_{ik}\, \mathbf{x}_i,$$

   where $n_k = \sum_{i=1}^{n} r_{ik}$.
3. **Convergence Check:** The algorithm stops when the assignments no longer change (or when the decrease in J is below a predefined threshold).

8.4.6 Summary of Variables and Their Dimensions

- $\mathbf{x}_i \in \mathbb{R}^d$: ith data point; column vector with $d \times 1$ dimensions.
- n: Number of data points.
- d: Number of features.
- K: Number of clusters.
- $\mathbf{c}_k \in \mathbb{R}^d$: Cluster center for cluster k (for $k = 1, 2, \ldots, K$).
- $r_{ik} \in \{0, 1\}$: Binary indicator; $r_{ik} = 1$ if $\mathbf{x}_i$ is assigned to cluster k and 0 otherwise.
- $n_k = \sum_{i=1}^{n} r_{ik}$: Number of data points assigned to cluster k; a scalar.
- $J \in \mathbb{R}_{\geq 0}$: Objective function value (sum of squared errors).

8.4.7 Final Thoughts

We have derived the k-Means clustering algorithm from first principles. Starting from a dataset $\mathcal{D}$ represented by a design matrix of n points in $\mathbb{R}^d$, we defined the k-Means objective function as the within-cluster sum-of-squares. We then derived the two key steps of the algorithm: (1) assigning each data point to the nearest cluster center (minimizing the objective with respect to the assignment variables) and (2) updating each cluster center as the mean of the points assigned to it (by setting the gradient of the objective with respect to the cluster center to zero). This section provides a thorough mathematical and algorithmic foundation for understanding and implementing the k-Means clustering model, with all variables and dimensions clearly specified.

8.5 Advantages, Disadvantages, Alternatives, & Applications

Advantages

- ✓ **Simplicity:** k-Means is easy to understand and implement.
- ✓ **Efficiency:** It is computationally efficient and scales well to large datasets.
- ✓ **Interpretability:** The resulting clusters are easy to interpret, and centroids provide a summary of the data.

Disadvantages

- ✗ **Sensitivity to Initialization:** The algorithm may converge to a local minimum depending on the initial centroids.
- ✗ **Fixed Number of Clusters:** The number of clusters k must be specified in advance.
- ✗ **Assumption of Spherical Clusters:** k-Means tends to perform poorly on clusters with non-convex shapes or varying densities, e.g., see Figure 8.3.

Alternatives

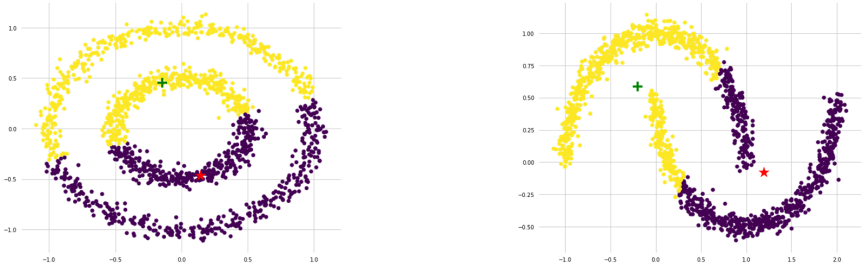
- ↬ **Hierarchical Clustering:** Builds a hierarchy of clusters without requiring a preset number of clusters.
- ↬ **DBSCAN:** A density-based method capable of finding arbitrarily shaped clusters and handling noise; see Figure 8.4.
- ↬ **Gaussian Mixture Models:** Probabilistic clustering that can model clusters with different shapes and sizes.

Real-World Applications & Use Cases

- **Image Segmentation:** Separating different regions within an image based on color or intensity; see Figure 8.5.
- **Market Segmentation:** Grouping customers according to purchasing behavior and demographics.
- **Document Clustering:** Organizing large collections of text data for information retrieval.
- **Anomaly Detection:** Identifying unusual patterns or outliers in datasets, such as fraud detection in finance.
- **Biological Data Analysis:** Grouping gene expression data or identifying patterns in medical imaging.

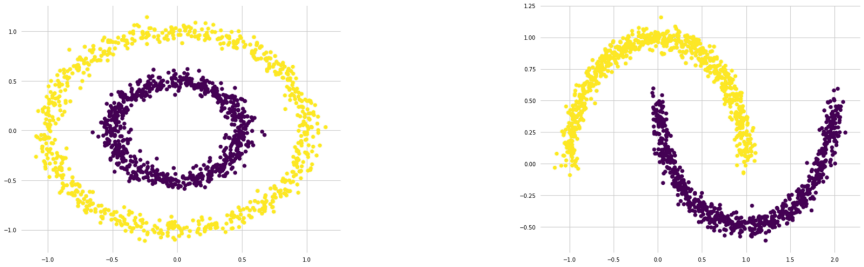
8.6 Conclusion

This chapter presented a comprehensive study of the k-Means algorithm. We began with an intuitive explanation supported by examples and graphical illustrations, followed by a detailed mathematical derivation of its objective function and iterative optimization process. A practical tutorial with



(a) (b)

Figure 8.3: An illustration of k-Means clustering that incorrectly groups points in curvilinear clusters.



(a) (b)

Figure 8.4: An illustration of DBSCAN clustering that correctly groups points in curvilinear clusters.

(a) (b)

Figure 8.5: An illustration of k-Means clustering for image color segmentation: (a) original image, (b) image regenerated with 2 colors.

both pseudo-code and a complete MATLAB implementation was provided to demonstrate how to implement k-Means from scratch and visualize the results. Finally, we discussed the advantages, disadvantages, alternatives, and real-world applications of k-Means.

Despite its sensitivity to initialization and requirement to specify k in advance, k-Means remains a fundamental and widely used clustering technique. For further information, please refer to the seminal works in the field [1, 2].

Bibliography

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297, this seminal work has been cited over 10,000 times.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009, this seminal work has been cited over 10,000 times.