Chapter 4: K-Means Clustering

Latreche Sara

Contents

1 Introduction to K-Means Clustering

Clustering is an **unsupervised learning** problem where we aim to group a set of data points into cohesive clusters. Each data point $x^{(i)} \in \mathbb{R}^d$ has no associated label, unlike in supervised learning.

The most commonly used algorithm for clustering is the **K-means algorithm**.

2 Intuition and Motivation

In the machine learning paradigm, we generally follow a three-step process:

- **Observe**: Gather a set of examples — known as the *training data*.
- **Infer**: Learn or estimate the process that generated the data.
- **Predict**: Use the learned model to make predictions on new, previously unseen data — the *test data*.

Machine learning tasks are typically classified into two main categories:

- **Supervised Learning:** Given a set of feature/label pairs $(x^{(i)}, y^{(i)})$, the goal is to learn a function that maps inputs x to outputs y. This function should generalize well to unseen data.
- **Unsupervised Learning:** Given a set of feature vectors $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ with no associated labels, the goal is to uncover hidden structures or patterns in the data. One common task is to group the data into "natural" clusters based on similarity.

K-means Clustering is a fundamental algorithm in unsupervised learning. It aims to partition the dataset into k groups such that the data points in the same group are more similar to each other than to those in other groups. In essence, it discovers underlying groupings within data by minimizing within-cluster variance.

3 K-means as an Optimization Problem

Clustering can be viewed as an optimization problem. The core idea of K-means is to assign each data point to one of K clusters such that the total dissimilarity within clusters is minimized.

Why not simply minimize total variability? That would allow for degenerate solutions like putting every point in its own cluster — which leads to zero variability but no meaningful grouping.

To prevent such trivial solutions, we impose a constraint: the number of clusters K is fixed and smaller than the number of data points.

K-means Optimization Problem

Given:

- Data points $\{x^{(i)}\}_{i=1}^{n} \subset \mathbb{R}^d$
- Number of clusters K

Find:

- Assignments $c^{(i)} \in \{1, \ldots, K\}$
- Centroids $\{\mu_k\}_{k=1}^{K} \in \mathbb{R}^d$

Objective:

$$\min_{\{c^{(i)}\},\{\mu_k\}} \quad J = \sum_{i=1}^{n} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2$$

What is being optimized?

- The cluster assignments $c^{(i)}$, which decide to which centroid each point is closest
- The cluster centroids μ_k, which are updated as the mean of points assigned to them

Numerical Example

Assume we have 4 data points on a line:

$$x^{(1)} = 1, \quad x^{(2)} = 2, \quad x^{(3)} = 8, \quad x^{(4)} = 9$$

and we want to form $K = 2$ clusters.
A good clustering would be:

- Cluster 1: $\{x^{(1)} = 1, x^{(2)} = 2\} \rightarrow \mu_1 = \frac{1+2}{2} = 1.5$
- Cluster 2: $\{x^{(3)} = 8, x^{(4)} = 9\} \rightarrow \mu_2 = \frac{8+9}{2} = 8.5$

Objective value:

$J = (1-1.5)^2+(2-1.5)^2+(8-8.5)^2+(9-8.5)^2 = 0.25+0.25+0.25+0.25 = 1$

Comprehension Questions

- What happens to the objective value if we increase the number of clusters?
- Why must we impose constraints on cluster assignments?
- How many clusters would be ideal for the data points: $\{1, 2, 3, 10, 11, 12\}$?
- Try recomputing the objective value if we grouped $\{1, 2, 10\}$ and $\{3, 11, 12\}$.

4 The K-Means Algorithm

The goal is to partition the dataset into k clusters such that the within-cluster variance is minimized. The algorithm proceeds as follows:

K-Means Clustering Algorithm

1: Initialize k cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^d$ randomly.
2: **repeat**
3: **for** each data point $x^{(i)}$ **do**
4: Assign it to the closest centroid:

$$c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2$$

5: **end for**
6: **for** each cluster $j = 1, \ldots, k$ **do**
7: Update centroid μ_j to mean of assigned points:

$$\mu_j := \frac{\sum_{i=1}^{n} \mathbf{1}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{c^{(i)} = j\}}$$

8: **end for**
9: **until** convergence (no change in assignments or centroids)

Objective Function (Distortion Function)

The K-means algorithm minimizes the following cost function:

$$J(c, \mu) = \sum_{i=1}^{n} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2$$

This is called the **distortion function**, and it measures the total squared distance from each point to its assigned cluster centroid.

Convergence

Each iteration of the K-means algorithm is guaranteed to decrease the distortion function J, and hence the algorithm is guaranteed to converge to a local minimum. However, since J is non-convex, convergence to the global minimum is not guaranteed.

Visualization Example

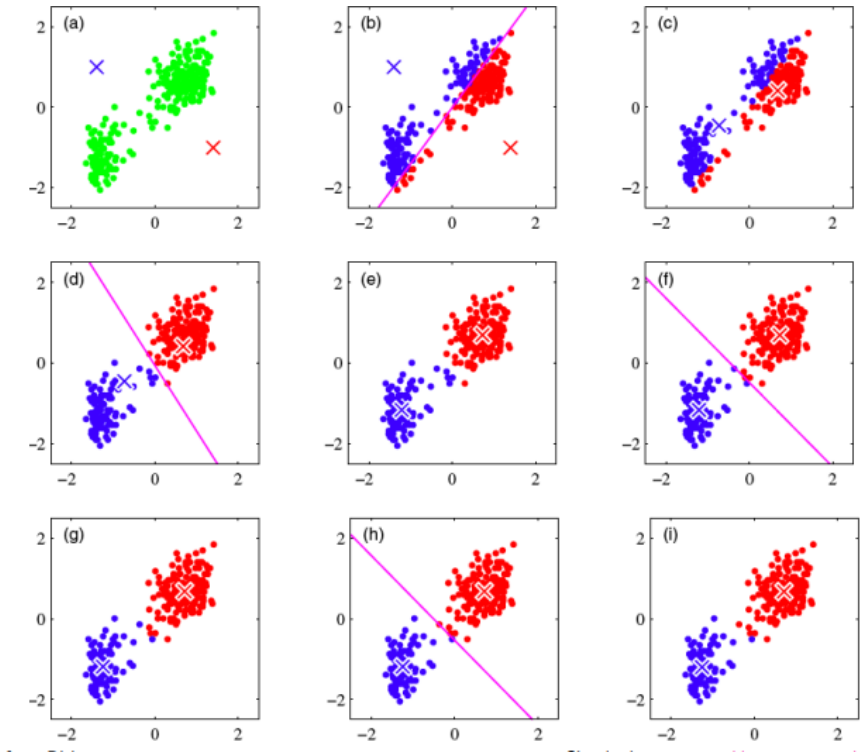

Figure 1: Example of K-Means clustering in 2D: Initial centroids (crosses) and points (dots) evolving over iterations.

5 Choosing the Best Number of Clusters

One of the key challenges in using K-means is selecting an appropriate number of clusters K. While the optimization objective J always decreases as K increases, setting K too high can lead to overfitting, while too small a K leads to underfitting.

The Elbow Method

The **Elbow Method** involves plotting the cost function J (sum of squared distances to cluster centers) against the number of clusters K. As K increases:

- J always decreases.
- The marginal gain (reduction in J) typically decreases.

Look for the **"elbow" point** — where the rate of decrease sharply shifts — as a good trade-off between model complexity and clustering quality.

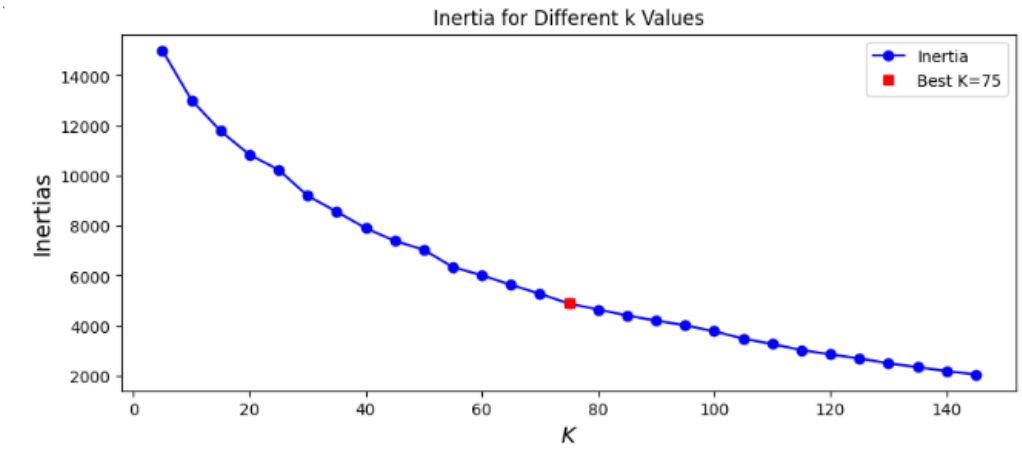

Figure 2: Elbow plot: The optimal K is where the curve bends (forms an "elbow").

Convergence of K-means

K-means alternates between two steps:

- **E-step (Assignment):** Assign each point to the nearest cluster center.

- **M-step (Update):** Update cluster centers as the mean of the assigned points.

Each of these steps reduces the objective J:

- Whenever an **assignment changes**, J decreases.
- Whenever a **cluster center is updated**, J decreases.

Convergence test: The algorithm has converged when no assignments change in the E-step. This ensures convergence to a local minimum.

Local Minima in K-means

The cost function J is non-convex. K-means can converge to poor local minima depending on the initial centroids. Strategies to address this include:

- **Multiple random initializations:** Run K-means multiple times with different starting points.
- **Split-and-merge moves:**
  - Merge two nearby clusters
  - Split a large, spread-out cluster into two
- **Use smarter initializations:** Such as K-means++ to improve starting centroid selection.

6 Silhouette Score for Cluster Evaluation

The Silhouette Score is a popular metric to evaluate clustering quality. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

For a given point $x^{(i)}$, define:

- $a(i)$: average distance of $x^{(i)}$ to all other points in the same cluster
- $b(i)$: lowest average distance of $x^{(i)}$ to all points in any other cluster (i.e., nearest cluster)

The silhouette score $s(i)$ is then:

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

- $s(i) \approx 1$: well-clustered
- $s(i) \approx 0$: on the boundary
- $s(i) < 0$: possibly misclassified

The average silhouette score over all points helps to determine the optimal number of clusters.

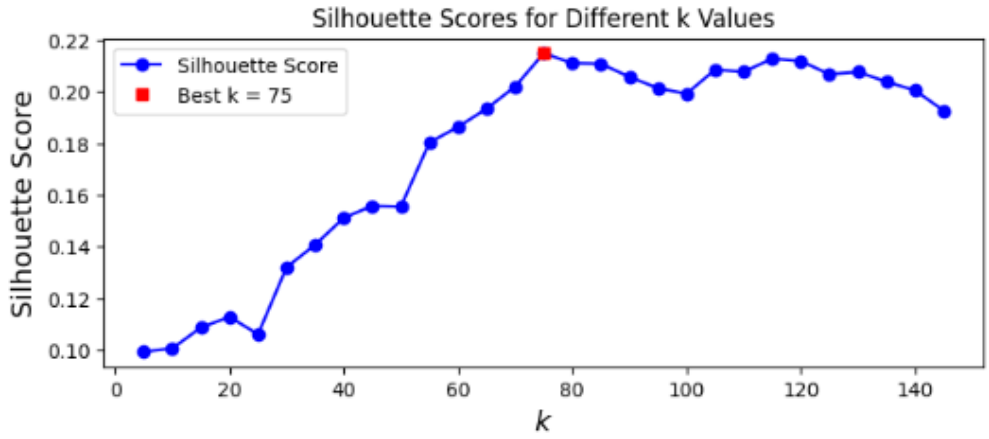

Figure 3: Silhouette scores for different values of K. The red point marks the best K.

Numerical Example: Computing Silhouette Score by Hand

Assume we have the following 3 points assigned to two clusters:

- $x^{(1)} = 1$, $x^{(2)} = 2$ in Cluster A
- $x^{(3)} = 8$ in Cluster B

For $x^{(1)}$:

- $a(1) =$ distance to $x^{(2)} = |1 - 2| = 1$
- $b(1) =$ distance to nearest cluster (only B): $|1 - 8| = 7$
- Silhouette:

$$s(1) = \frac{7 - 1}{\max(7, 1)} = \frac{6}{7} \approx 0.857$$

Repeat similarly for others, then take the average:

$$\text{Avg silhouette} = \frac{s(1) + s(2) + s(3)}{3}$$

7 Tips for Practical Use

- Run the algorithm multiple times with different random initializations and choose the best result.
- Use the Elbow Method to choose the number of clusters k.
- Normalize your data before clustering.

7.1 Two Popular Clustering Methods

Clustering is a fundamental task in unsupervised learning. Two widely used clustering methods are:

- **K-means clustering**

- **Hierarchical clustering**

Hierarchical Clustering

Hierarchical clustering builds a hierarchy of nested clusters either in a *bottom-up* (agglomerative) or *top-down* (divisive) fashion. The agglomerative approach, which we focus on here, follows this procedure:

1. Start by assigning each item to its own cluster. If there are N items, you begin with N clusters.
2. Find the pair of clusters that are closest together, and merge them into a single cluster.
3. Repeat step 2 until all items are merged into one single cluster containing all N points.

What does distance mean? The similarity between clusters depends on the *linkage metric*:

- **Single-linkage:** Distance between two clusters is defined as the shortest distance between any two members of the two clusters.
- **Complete-linkage:** Distance is defined as the greatest distance between any two members.
- **Average-linkage:** Distance is the average of all pairwise distances between elements in the two clusters.

Example: Hierarchical Clustering of US Cities

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

Starting configuration:

$$\{\text{BOS}\}, \{\text{NY}\}, \{\text{CHI}\}, \{\text{DEN}\}, \{\text{SF}\}, \{\text{SEA}\}$$

Merge steps (using a linkage strategy):

$$\{\text{BOS, NY}\}, \{\text{CHI}\}, \{\text{DEN}\}, \{\text{SF}\}, \{\text{SEA}\}$$
$$\{\text{BOS, NY, CHI}\}, \{\text{DEN}\}, \{\text{SF}\}, \{\text{SEA}\}$$
$$\{\text{BOS, NY, CHI}\}, \{\text{DEN}\}, \{\text{SF, SEA}\}$$
$$\{\text{BOS, NY, CHI, DEN}\}, \{\text{SF, SEA}\}$$
$$\{\text{BOS, NY, CHI, DEN, SF, SEA}\}$$

The result can be visualized using a **dendrogram**, where the vertical axis represents inter-cluster distance at each merge step.

Comparison to K-means:

- Hierarchical clustering does not require specifying K in advance.
- It is deterministic (no random initialization).
- It can be computationally expensive for large datasets.
- K-means is generally faster but may get stuck in local minima.

Limitations and Applications of K-means Clustering

Limitations of K-means Clustering

- **Requires predefined number of clusters** K: The algorithm needs the number of clusters to be set manually, which may not be known in advance.
- **Sensitive to initialization**: Different initial placements of cluster centroids can lead to different results due to convergence to local minima.
- **Assumes spherical clusters**: K-means performs poorly when clusters are non-spherical or have different densities and sizes.
- **Not robust to outliers**: Outliers can skew centroid positions significantly.
- **Only captures linear boundaries**: K-means fails to detect complex or non-convex cluster shapes.
- **Distance metric limitations**: By default, K-means uses Euclidean distance, which may not be appropriate for all datasets, especially those with categorical features.

Applications of K-means Clustering

- **Image compression**: Clustering similar colors in images and representing them with a reduced palette.
- **Customer segmentation**: Grouping customers by purchasing behavior or demographics for targeted marketing.
- **Document classification**: Organizing documents or news articles into topics based on word vector embeddings.
- **Market basket analysis**: Grouping frequently bought-together products into clusters.

- **Anomaly detection**: Identifying points far from any cluster as potential anomalies or outliers.
- **Genomic data analysis**: Clustering gene expression patterns across different conditions or organisms.

Exercises for Comprehension and Numerical Practice

Conceptual Questions

1. What are the main differences between K-means and Hierarchical clustering?
2. Why is the K-means objective function not guaranteed to converge to the global minimum?
3. Explain what is meant by:
   - Single-linkage
   - Complete-linkage
   - Average-linkage
4. How does the silhouette score help determine the optimal number of clusters?
5. What happens to the K-means objective function J after each E-step and M-step?

Numerical Exercises

1. **Manual K-means (2D data):** Given the data points: $A(1,1), B(2,1), C(4,3), D(5,4)$, run **one full iteration** of K-means with $K = 2$ and initial centroids: $\mu_1 = (1,1)$, $\mu_2 = (5,4)$. Show:
   - The assignment step

- The updated centroids
- The cost function J

2. **Silhouette Score Calculation:** Given a small cluster $C_1 = \{A, B\}$ and another cluster $C_2 = \{C\}$ with distances:

$$d(A, B) = 1, \quad d(A, C) = 5, \quad d(B, C) = 6$$

Compute the silhouette score $s(A)$. Use the formula:

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

where $a(i)$ is the average distance to other points in the same cluster and $b(i)$ is the minimum average distance to another cluster.

3. **Hierarchical Clustering Walkthrough:** Given the distance matrix below, perform the first two steps of agglomerative hierarchical clustering using **single-linkage**:

	A	B	C
A	0	3	4
B		0	2
C			0

Draw the intermediate clusters after each step.

Programming Task (Optional)

- Use scikit-learn to:

1. Generate synthetic 2D data using 'make$_b lobs$'.$RunK-means for K =$ 1 to $K = 10$.
2. Plot the elbow curve and silhouette scores.
3. Identify the optimal K visually and explain why.

References

- MIT OpenCourseWare, 6.0002: Introduction to Computational Thinking and Data Science, Fall 2016. Lecture 12.
- Stanford University, CS229: Machine Learning, Lecture Notes on Clustering and K-means.
`https://cs229.stanford.edu/`