K-means Clustering

Introduction, Formulation, and Applications

Sayan Chaki
sayan.chaki@inria.fr

March 19, 2025

Introduction to K-means Clustering

- **Definition**: K-means is an unsupervised machine learning algorithm used for partitioning data into K distinct non-overlapping clusters
- **Goal**: Group similar data points together while keeping different ones apart
- **History**: First proposed by Stuart Lloyd in 1957 (published in 1982)
- **Popularity**: One of the most widely used clustering algorithms due to its simplicity and efficiency

The Problem K-means Solves

Given:

- A dataset $X = \{x_1, x_2, ..., x_n\}$ where each $x_i \in \mathbb{R}^d$
- Number of clusters K

Find:

- K cluster centroids $\{\mu_1, \mu_2, ..., \mu_K\}$
- Assignment of each data point to exactly one cluster
- Such that the total within-cluster variation is minimized

When to Use K-means

Ideal conditions:

- Clusters are roughly spherical
- Clusters have similar sizes
- Data dimensionality is not too high

Appropriate applications:

- Customer segmentation
- Image compression
- Anomaly detection
- Document clustering
- Feature engineering (creating new features)

Requirements:

- Need to specify K in advance
- Requires a distance metric (typically Euclidean)

K-means Formulation

Core Idea: Iteratively assign points to the nearest centroid, then update centroids

Input:

- Dataset $X = \{x_1, x_2, ..., x_n\}$
- Number of clusters K

Algorithm:

1. Initialize K centroids randomly
2. **Repeat until convergence**:
   - **Assignment step**: Assign each point to nearest centroid
   - **Update step**: Recalculate centroids as mean of assigned points

Mathematical Formulation

Objective Function: Minimize the sum of squared distances

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|^2 \tag{1}$$

Where:

- $r_{ik} \in \{0, 1\}$ indicates if point x_i belongs to cluster k
- μ_k is the centroid of cluster k

Hard assignment constraint: Each point belongs to exactly one cluster

$$\sum_{k=1}^{K} r_{ik} = 1 \quad \forall i \tag{2}$$

Centroid Optimization

Assignment step: For fixed centroids μ_k, minimize J with respect to r_{ik}:

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x_i - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Update step: For fixed assignments r_{ik}, minimize J with respect to μ_k:

$$\mu_k = \frac{\sum_{i=1}^{n} r_{ik} x_i}{\sum_{i=1}^{n} r_{ik}} \tag{4}$$

This is the mean of all points assigned to cluster k.

Elbow Method for Choosing K

Challenge: K-means requires specifying K in advance

Solution: Elbow method helps determine appropriate K

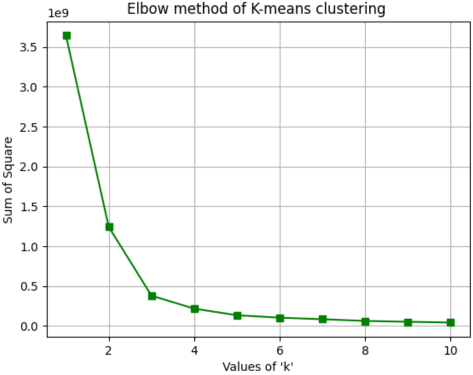
Method:

1. Run K-means with increasing values of K
2. Calculate Within-Cluster Sum of Squares (WCSS) for each K:

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \tag{5}$$

3. Plot WCSS against K
4. Look for the "elbow" where adding more clusters yields diminishing returns

Elbow Method Visualization



- WCSS decreases as K increases (always)
- The "elbow" indicates optimal K (K=3 in this example)
- Beyond this point, additional clusters provide marginal benefit

K-means vs. K-Nearest Neighbors (KNN)

K-means	**K-Nearest Neighbors**
Unsupervised learning	Supervised learning
Clustering algorithm	Classification/regression algorithm
Partitions data space	Makes predictions based on neighbors
"K" refers to number of clusters	"K" refers to number of neighbors
Creates new points (centroids)	Uses existing labeled points
Distance to centroids matters	Distance to neighbors matters
Training is iterative	No actual training phase

Application: Classifying Digits 0 and 1

Task: Cluster MNIST digit images (0s and 1s) using K-means

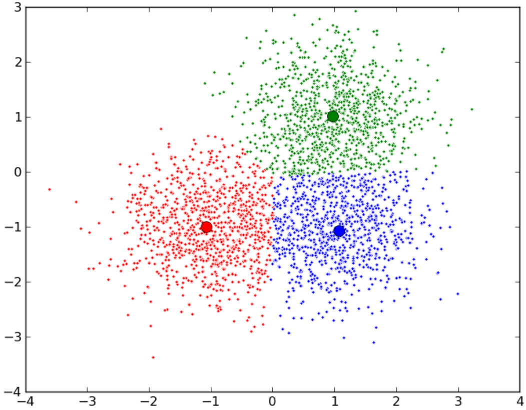
Procedure:

1. Represent each 28×28 image as a 784-dimensional vector
2. Apply K-means with $K=2$
3. Evaluate clustering against ground truth labels

Expected outcome:

- Cluster 1 should contain mostly 0s
- Cluster 2 should contain mostly 1s
- Visualization shows effective separation based on pixel patterns

Classification Results



Analysis:

- K-means achieves ∼97% accuracy despite not using labels
- Misclassifications often occur with unusual digit styles

Summary: K-means Clustering

Strengths:

- Simple, efficient algorithm
- Scales well to large datasets
- Guaranteed to converge (to local optimum)

Limitations:

- Requires specifying K
- Sensitive to initialization
- Assumes spherical clusters
- Affected by outliers

Extensions:

- K-means++: Smarter initialization
- Mini-batch K-means: For very large datasets
- Spectral clustering: For non-spherical clusters