Mathematics and Technical Explanation of K-Means Clustering

K-Means clustering is a centroid-based unsupervised learning algorithm that partitions a dataset into K non-overlapping clusters, where each data point belongs to the cluster with the nearest centroid (mean). It minimizes intra-cluster variance, making it a form of vector quantization. Originally from signal processing, it's widely used in data science for segmentation, anomaly detection, and feature engineering. The algorithm assumes spherical clusters of similar size and is sensitive to initialization and outliers.

en.wikipedia.org

1. Mathematical Objective: Minimizing the Within-Cluster Sum of Squares (WCSS)

The core goal is to find K centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$ that minimize the WCSS (also called inertia), which quantifies how tightly points are grouped around centroids. For a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ where each $\mathbf{x}_i \in \mathbb{R}^d$ (N samples, d dimensions), and cluster assignments C_k (set of points in cluster k):

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2$$

- $\| \cdot \|_2^2$: Squared Euclidean distance (L2 norm squared), measuring dissimilarity.
- $\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i$: Centroid as the arithmetic mean of points in C_k (optimal for Euclidean distance).
- Intuition: This is a least-squares optimization problem, akin to variance minimization in statistics. J is non-convex, so the algorithm finds local minima via iteration.

  towardsdatascience.com

From a probabilistic view, K-Means approximates the expectation-maximization (EM) algorithm for a Gaussian mixture model with equal variances and hard assignments (instead of soft probabilities).

2. The Lloyd's Algorithm: Iterative Optimization

K-Means uses an iterative approach (Lloyd's algorithm) to approximate the global minimum of J. It alternates between two steps until convergence (e.g., centroids stabilize or max iterations reached). math.ucdavis.edu

Step 1: Assignment (E-Step)

Assign each point $\mathbf{x}_i$ to the nearest centroid:

$$c_i = \arg\min_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2, \quad \forall i = 1 \to N$$

- c_i: Cluster label for $\mathbf{x}_i$ (hard assignment).
- This partitions the space into Voronoi cells (regions closer to one centroid than others).

Step 2: Update (M-Step)

Recalculate centroids as means of assigned points:

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}_i : c_i = k} \mathbf{x}_i, \quad \forall k = 1 \to K$$

- If a cluster is empty ($|C_k| = 0$), reinitialize $\boldsymbol{\mu}_k$ randomly or drop it.
- Convergence: Monotonically decreases J (or stays constant); guaranteed in finite steps since assignments are finite.

Initialization: Often random or via K-Means++ (select initial centroids far apart to avoid poor local minima):

- Pick first centroid uniformly at random.
- For subsequent: Probability proportional to squared distance to nearest existing centroid.

Complexity: O(N K d I), where I is iterations (typically 10-50). stanford.edu

3. Derivation of Optimality

Why means? Differentiate J w.r.t. $\boldsymbol{\mu}_k$:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = -2 \sum_{\mathbf{x}_i \in C_k} (\mathbf{x}_i - \boldsymbol{\mu}_k) = 0 \implies \boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i$$

- This is the sample mean, minimizing squared error (like OLS regression centroid).

For assignments: Fixing centroids, assigning to nearest minimizes J greedily.

4. Example: 2D Data Clustering

Consider N=6 points in $\mathbb{R}^2$: A(1,1), B(1,2), C(2,1), D(5,5), E(5,6), F(6,5). Set K=2.

Initialization: Random centroids $\boldsymbol{\mu}_1 = (1, 1)$, $\boldsymbol{\mu}_2 = (5, 5)$.

Iteration 1 - Assignment:

- Dist to $\boldsymbol{\mu}_1$: A=0, B=1, C=1, D≈5.66, E≈6.40, F≈6.40 → Cluster 1: A,B,C
- Dist to $\boldsymbol{\mu}_2$: → Cluster 2: D,E,F

Update:

- $\boldsymbol{\mu}_1 = \frac{(1,1)+(1,2)+(2,1)}{3} = (1.33, 1.33)$
- $\boldsymbol{\mu}_2 = \frac{(5,5)+(5,6)+(6,5)}{3} = (5.33, 5.33)$

Iteration 2: Assignments unchanged → Converge.

J final ≈ 2.67 (sum of squared dists).

If poor init (e.g., both near A), may converge to suboptimal J=10+.

5. Architecture/Process Diagram (Text-Based Representation)

K-Means is iterative; here's a flowchart in ASCII (visualize as blocks):

text Collapse Wrap Copy

```
Start
  ↓
Input: Data X (N x d), K
  ↓
Initialize Centroids (Random or K-Means++)
  ↓
Loop until convergence:
  |
  |  Assignment: For each x_i, c_i = argmin ||x_i - μ_k||²
  |
  |  Update: For each k, μ_k = mean of {x_i | c_i = k}
  |
  ↓ (Check if centroids changed < ε)
Output: Clusters C_1..K, Centroids μ_1..K
```

- Arrows: Flow; Loop: Iteration box.
- In code (pseudocode): While not converged, assign, update.

6. Limitations and Variants

- Assumes isotropic clusters (fails on elongated shapes).
- Sensitive to K (use Elbow method: Plot J vs. K, find "elbow").
- Outliers skew means (use K-Medoids with medians).
- Variants: Fuzzy C-Means (soft assignments via probabilities).

This provides a rigorous foundation; for proofs (e.g., convergence), see MacQueen (1967).

en.wikipedia.org