K-Means Clustering: Algorithm and Analysis

Machine Learning Foundations

November 24, 2025

Abstract

This document presents a comprehensive study of K-means clustering, including algorithm implementation, convergence analysis, cluster quality metrics (silhouette score, inertia), the elbow method for optimal K selection, and comparison with other clustering approaches. We demonstrate practical considerations for initialization and scaling.

1 Introduction

K-means clustering partitions n observations into K clusters by minimizing within-cluster variance:

$$J = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2 \tag{1}$$

where C_k is the set of points in cluster k and μ_k is the centroid.

The algorithm alternates between:

1. **Assignment**: Assign each point to nearest centroid
2. **Update**: Recompute centroids as cluster means

2 Computational Environment

3 Data Generation

Dataset: $n =$ **??** samples, $p =$ **??** features, $K_{true} =$ **??** clusters.

4 K-Means Implementation

5 Algorithm Convergence

Algorithm converged in **??** iterations with inertia $J =$ **??**.

6 Elbow Method for Optimal K

Elbow method suggests $K =$ **??**, silhouette analysis suggests $K =$ **??** with score **??**.

7 Silhouette Analysis

8 Initialization Comparison

Mean inertia: Random = **??**, K-Means++ = **??**.

9 Cluster Visualization

10 Cluster Statistics

11 Results Summary

Table 1: K-Means Clustering Summary

Metric	Value
Dataset Size	??
True Clusters	??
Optimal K (Elbow)	??
Optimal K (Silhouette)	??
Best Silhouette Score	??
Final Inertia	??
Convergence Iterations	??

12 Conclusion

This analysis demonstrated:

- K-means algorithm implementation with K-means++ initialization
- Convergence visualization showing centroid updates
- Elbow method and silhouette analysis for optimal K selection
- Importance of initialization (K-means++ outperforms random)
- Voronoi regions showing cluster boundaries
- Detailed cluster quality metrics

The K-means++ initialization consistently achieves lower inertia (**??** vs **??** for random), and the optimal number of clusters matches the true value of **??**.