Inf620 2025 Lecture Unsupervised Learning and Cluster: Kmedoids, Kmedian

Depto de Informática - UFV

Ricardo Ferreira
ricardo@ufv.br

2025

Lesson Plan

- Class Material (click here for the Colab)
- **Review**: K-means clustering
- **Today's Lesson**: Extending clustering techniques
  - K-median clustering
  - K-medoids clustering
  - Comparison of methods
  - Practical implementations

Review: K-Means Clustering

- Partitional clustering algorithm
- Goal: Partition data into k clusters where each observation belongs to the cluster with the nearest mean
- Objective function: Minimize the sum of squared distances

$$\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} |\mathbf{x} - \boldsymbol{\mu}_i|^2 \tag{1}$$

where $\boldsymbol{\mu}_i$ is the mean of points in cluster S_i

K-Means Algorithm

Algorithm 1 K-Means Algorithm

```
Input: Dataset X, number of clusters k
Initialize cluster centroids μ_1, μ_2, ..., μ_k randomly
repeat
    Assignment step: Assign each point to closest centroid
    S_i = x_j : |x_j − μ_i| ≤ |x_j − μ_l| for all l ≠ i
    Update step: Recalculate centroids as means
    μ_i = 1/|S_i| Σ_{x_j ∈ S_i} x_j
until centroids no longer change
Return: Clusters S_1, S_2, ..., S_k and centroids μ_1, μ_2, ..., μ_k
```

K-Means: Strengths and Limitations

Strengths

- Simple to implement
- Linear time complexity $O(nkdi)$
- Converges to a local optimum
- Scales well to large datasets

Limitations

- Sensitive to outliers
- Needs pre-specified k
- Only finds convex clusters
- Sensitive to initialization
- Not appropriate for categorical data

K-Median: Overview

- Variation of K-means that uses **median** instead of mean
- Objective function: Minimize the sum of L1 distances

$$\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} |\mathbf{x} - \mathbf{m}_i|_1 \quad (2)$$

where $\mathbf{m}_i$ is the coordinate-wise median of points in S_i

- More robust to outliers than K-means
- Uses Manhattan distance (L1 norm) instead of Euclidean distance

K-Median: Algorithm

Algorithm 2 K-Median Algorithm

Input: Dataset X, number of clusters k
Initialize cluster medians $m_1, m_2, \ldots, m_k$ randomly
repeat
  Assignment step: Assign each point to closest median
  $S_i = x_j : |x_j - m_i|_1 \leq |x_j - m_l|_1$ for all $l \neq i$
  Update step: Recalculate medians
  $m_i = \text{median}(x_j : x_j \in S_i)$ (coordinate-wise)
until medians no longer change
Return: Clusters $S_1, S_2, \ldots, S_k$ and medians $m_1, m_2, \ldots, m_k$

K-Median: Mathematical Formulation

- For a cluster S_i with points $x_1, x_2, \ldots, x_n$:
- The coordinate-wise median m minimizes:

$$\sum_{j=1}^{n} |x_j - m|1 = \sum j = 1^n \sum_{d=1}^{D} |x_{j,d} - m_d| \quad (3)$$

- Each dimension d of the median is computed independently:

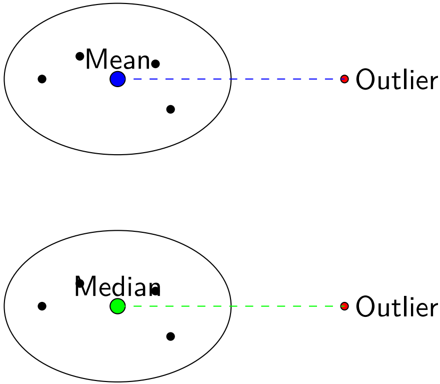
$$m_d = \text{median}(x_{1,d}, x_{2,d}, \ldots, x_{n,d}) \quad (4)$$

K-Median: Properties

- **Robustness**: Less influenced by outliers than K-means
- **Complexity**: $O(nkdi \log n)$ - slower than K-means due to median computation
- **Distance Metric**: Uses L1 norm (Manhattan distance)
- **Applications**:
  - Data with potential outliers
  - Data where Manhattan distance is more appropriate
  - Financial and economic data analysis

Visual Comparison: K-means vs K-median

K-means (sensitive to outlier)



K-median (more robust to outlier)

Mean vs Median: Numerical Example

Dataset 1: Without Outlier

- Values: 2, 4, 5, 6, 8
- Mean: $\frac{2+4+5+6+8}{5} = \frac{25}{5} = 5$
- Median: 5 (middle value)
- **Observation**: Mean = Median

Dataset 2: With Outlier

- Values: 2, 4, 5, 6, 8, 100
- Mean: $\frac{2+4+5+6+8+100}{6} = \frac{125}{6} = 20.83$
- Median: 5.5 (average of 5 and 6)
- **Observation**: Mean shifts dramatically, Median remains stable

Key Insight: The median is robust to outliers, making K-median clustering less sensitive to extreme values compared to K-means

Impact of Outliers on Clustering

K-means Example

$$\text{Cluster data:} \{2, 4, 5, 6, 8, 100\}$$
$$\text{Centroid:} \frac{2+4+5+6+8+100}{6} = 20.83$$

Sum of Squares:

$$(2 - 20.83)^2 = 354.52$$
$$(4 - 20.83)^2 = 283.52$$
$$(5 - 20.83)^2 = 250.59$$
$$\vdots$$
$$(100 - 20.83)^2 = 6256.75$$

Centroid pulled strongly toward outlier

Impact of Outliers on Clustering

K-median Example

$$\text{Cluster data:} \{2, 4, 5, 6, 8, 100\}$$
$$\text{Centroid:} \text{median} = 5.5$$

Sum of Absolute Distances:

$$|2 - 5.5| = 3.5$$
$$|4 - 5.5| = 1.5$$
$$|5 - 5.5| = 0.5$$
$$\vdots$$
$$|100 - 5.5| = 94.5$$

Centroid remains representative of majority

K-Medoids: Overview

- Variation of clustering where cluster centers are actual data points (medoids)
- Objective function: Minimize the sum of dissimilarities

$$\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} d(\mathbf{x}, \mathbf{o}_i) \tag{5}$$

where $\mathbf{o}_i$ is the medoid of cluster S_i and d is any distance function

- Even more robust to outliers and noise than K-median
- Can work with any distance/dissimilarity measure

K-Medoids: PAM Algorithm

Algorithm 3 Partitioning Around Medoids (PAM)

```
Input: Dataset X, number of clusters k, distance function d
BUILD phase: Select initial k medoids
repeat
    Assignment: Assign each point to closest medoid
    S_i = x_j : d(x_j, o_i) ≤ d(x_j, o_l) for all l ≠ i
    SWAP phase: For each medoid o_i and each non-medoid x
        Calculate cost change if o_i is replaced by x
        Select swap that gives greatest cost reduction
        If no cost-reducing swap exists, terminate
until no improvement in cost
Return: Clusters S_1, S_2, ..., S_k and medoids o_1, o_2, ..., o_k
```

K-Medoids: Visualization and References

Animated Gif - Click here

Key Papers: K-Medoids:

Kaufman, L., & Rousseeuw, P. J. (1990). *Partitioning Around Medoids (Program PAM)*. Finding Groups in Data: An Introduction to Cluster Analysis- click Scholar, 68-125.

K-Medoids: Variants

- **CLARA** (Clustering LARge Applications)
  - For large datasets where PAM is computationally expensive
  - Samples smaller subsets and applies PAM to each
  - Selects the best clustering among all samples
- **CLARANS** (Clustering Large Applications based upon RANdomized Search)
  - Improves on CLARA by using randomized search
  - Dynamically draws samples throughout the search process
  - Better balance between efficiency and effectiveness

K-Medoids: Properties

- **Robustness**: Highly robust to outliers and noise
- **Complexity**: $O(k(n-k)^2)$ per iteration - more expensive than K-means/K-median
- **Distance Metric**: Works with any distance/dissimilarity measure
- **Applications**:
  - Clustering with non-Euclidean distances
  - Categorical or mixed-type data
  - When interpretable centers are required (medoids are actual data points)
  - Bioinformatics, text clustering, social network analysis

When to Use Each Method

Use K-means when:

- Large datasets
- Speed is critical
- Data is numeric
- Clusters expected to be spherical
- Few outliers expected

Use K-median when:

- Some outliers present
- L1 distance is appropriate
- Medium-sized datasets
- Need robustness without sacrificing too much speed

Use K-medoids when:

- Many outliers present
- Non-Euclidean distances needed
- Categorical/mixed data
- Need interpretable centers
- Can handle computational cost

Working with Real Data

- **Data preparation:**
  - Feature scaling is important for all methods
  - Consider dimensionality reduction for high-dim data
  - Handle missing values appropriately
- **Choosing k:**
  - Elbow method (plot within-cluster sum of squares vs. k)
  - Silhouette analysis
  - Gap statistic
  - Domain knowledge
- **Evaluation metrics:**
  - Silhouette score
  - Davies-Bouldin index
  - Calinski-Harabasz index

Conclusion

- K-means, K-median, and K-medoids form a family of partitional clustering algorithms
- Each has its own strengths and weaknesses:
  - K-means: Efficient but sensitive to outliers
  - K-median: Balance between robustness and efficiency
  - K-medoids: Most robust but computationally expensive
- Consider your specific requirements:
  - Dataset size and dimensionality
  - Presence of outliers
  - Type of data (numeric, categorical, mixed)
  - Appropriate distance measure
  - Computational resources available

Further Reading

- Arthur, D., Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding.
- Kaufman, L., Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis.
- Park, H. S., Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering.
- Ng, R. T., Han, J. (2002). CLARANS: A Method for Clustering Objects for Spatial Data Mining.

Next class: Hierarchical Clustering Methods