

# Cluster Analysis

Discovering Natural Groupings in Data

Juliho Castillo Colmenares

Tec de Monterrey

# Today's Agenda

1. Introduction to Cluster Analysis
2. Distance and Similarity Measures
3. Hierarchical Clustering Methods
4. K-Means and Non-Hierarchical Methods
5. Determining Optimal Number of Clusters
6. Validation Techniques
7. Practical Considerations
8. Applications and Best Practices

# What is Cluster Analysis?

**Definition:** An exploratory technique to discover natural groupings in data **without predefined categories**

# What is Cluster Analysis?

## Key Characteristics:

- Unsupervised learning method
- No training labels required
- Discovers hidden structure in data
- Groups similar observations together

**Goal:** Maximize within-cluster similarity and between-cluster dissimilarity

# Cluster Analysis vs. Discriminant Analysis

<b>Cluster Analysis</b>	<b>Discriminant Analysis</b>
Unsupervised learning	Supervised learning
Discovers unknown groups	Classifies into known groups
No training labels	Requires training labels
Exploratory	Predictive
Groups observations	Creates decision boundaries

# Applications: Marketing & Business

## Marketing

- Customer segmentation for targeted campaigns
- Market basket analysis

## Business

- Fraud detection
- Anomaly identification

# Applications: Science & Healthcare

## Biology & Medicine

- Disease subtype identification
- Gene expression analysis

## Social Sciences

- Community detection in networks
- Document clustering

# Distance and Similarity Measures

## Why Distance Matters

Clustering depends on measuring how “close” observations are to each other



# Common Distance Metrics

1. **Euclidean Distance** (L2 norm) - Most common
2. **Manhattan Distance** (L1 norm) - Robust to outliers
3. **Cosine Similarity** - For high-dimensional data
4. **Correlation Distance** - Pattern similarity

# Euclidean Distance

Formula:

$$d(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$$

# Euclidean Distance

## Properties:

- Straight-line distance in n-dimensional space
- Sensitive to scale differences
- Assumes equal importance of all dimensions

**Warning:** Always standardize variables with different scales!

# Manhattan Distance

Formula:

$$d(x, y) = \sum_{i=1}^p |x_i - y_i|$$

# Manhattan Distance

## When to Use:

- Data contains outliers or extreme values
- Variables represent counts
- High-dimensional spaces

**Advantage:** More robust than Euclidean distance

# Why Standardization is Critical

**Problem:** Variables on different scales dominate distance calculations

**Example:**

- Age: 20-80 years
- Income: 20,000-200,000 dollars

Without standardization, income dominates!

# Z-score Standardization

## Solution: Z-score Standardization

$$z_i = \frac{x_i - \mu}{\sigma}$$

Transform to mean = 0, standard deviation = 1

# Hierarchical Clustering

Builds a tree-like structure (dendrogram) showing nested clusters



# Two Approaches

**Agglomerative (Bottom-Up):** Most common

- Start: Each observation is its own cluster
- Process: Merge closest clusters iteratively
- End: All observations in one cluster

## Two Approaches

**Divisive (Top-Down):** Less common

- Start: All observations in one cluster
- Process: Split most heterogeneous cluster
- End: Each observation is its own cluster

# Linkage Methods

How to Measure Distance Between Clusters?

# Single Linkage (Nearest Neighbor)

$$d(C_1, C_2) = \min_{x \in C_1, y \in C_2} d(x, y)$$

Distance between closest points in the two clusters

# Complete Linkage (Farthest Neighbor)

$$d(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y)$$

Distance between farthest points in the two clusters

# Average Linkage

$$d(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y)$$

Average distance between all pairs of points

# Ward's Method

## Ward's Method

Minimizes within-cluster sum of squares

Tends to produce compact, equal-sized clusters

# Linkage Methods Comparison

Method	Outlier Sensitivity	Cluster Shape
Single Linkage	High	Elongated (chaining)
Complete Linkage	Low	Compact, spherical
Average Linkage	Medium	Balanced
Ward's Method	Medium	Compact, equal-sized



# Linkage Methods Comparison

**Recommendation:** Ward's method often works best in practice

# Dendrograms

Visualizing Hierarchical Structure

# Reading a Dendrogram

- Horizontal axis: Observations or clusters
- Vertical axis: Distance at which clusters merge
- Height of branches: Dissimilarity between merged clusters

# Determining Number of Clusters

- Look for large vertical gaps (jumps in fusion distance)
- Cut dendrogram where there's substantial increase
- Draw horizontal line: number of vertical lines crossed =  $k$  clusters

# The Chaining Effect

## Problem with Single Linkage:

Clusters form long, elongated chains rather than compact groups

# The Chaining Effect

## Why it Happens:

- Observations connect via intermediate points
- A-B-C-D form chain where each is close to neighbor
- But A and D are far apart

# The Chaining Effect

## Solution:

- Use complete or average linkage instead
- Or Ward's method for compact clusters

# K-Means Clustering

Most popular non-hierarchical method



# K-Means Algorithm

1. **Initialize:** Select  $k$  random observations as centroids
2. **Assignment:** Assign each point to nearest centroid
3. **Update:** Recalculate centroids as cluster means
4. **Repeat:** Steps 2-3 until convergence

**Convergence:** When assignments no longer change between iterations

# K-Means Objective Function

**Goal:** Minimize within-cluster sum of squares (WCSS)

$$\min \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

where  $\mu_i$  is the centroid of cluster  $C_i$

# K-Means Properties

- Always converges (finite partitions, monotonically decreasing WCSS)
- Typically converges in 10-30 iterations
- Fast:  $O(n \text{ times } k \text{ times } p \text{ times iterations})$

# K-Means: Advantages

- Fast and scalable to large datasets
- Simple to understand and implement
- Efficient for exploratory analysis

# K-Means: Limitations

- Requires specifying  $k$  in advance
- Sensitive to initialization (different starts  $\rightarrow$  different results)
- Assumes spherical clusters
- Sensitive to outliers
- Tends to create equal-sized clusters

# K-Means++ Initialization

**Problem:** Random initialization can lead to poor results

# K-Means++ Algorithm

1. Choose first centroid randomly
2. For each subsequent centroid:
  - Choose point with probability proportional to squared distance from nearest existing centroid
3. Repeat until k centroids selected

**Benefit:** Spreads out initial centroids, significantly improves results

# K-Medoids (PAM)

## Key Difference from K-Means:

- K-means: Centers are computed means (may not be actual points)
- K-medoids: Centers are actual data points (medoids)



# K-Medoids (PAM)

## Advantages:

- More robust to outliers
- Works with any distance metric
- Interpretable centers (actual observations)

**Disadvantage:** Slower than k-means (higher computational cost)

# How Many Clusters?

## The Fundamental Challenge:

No “ground truth” for correct number of clusters

# Multiple Approaches

1. **Elbow Method** - Look for bend in WCSS plot
2. **Silhouette Analysis** - Measure cluster quality
3. **Gap Statistic** - Compare to null reference
4. **Davies-Bouldin Index** - Ratio of compactness to separation
5. **Domain Knowledge** - Business requirements

# Elbow Method: Procedure

1. Run clustering for  $k = 1, 2, 3, \dots, K_{\max}$
2. Calculate WCSS for each  $k$
3. Plot WCSS vs.  $k$
4. Look for “elbow” - diminishing returns point

# Elbow Method: Interpretation

- WCSS always decreases as  $k$  increases
- Elbow indicates where additional clusters don't help much
- Choose  $k$  at the elbow point

**Limitation:** Elbow not always clear - may need other methods

# Silhouette Analysis

Measures how well each point fits within its cluster

# Silhouette Coefficient

**Silhouette Coefficient for observation  $i$ :**

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

where:

- $a(i)$  = avg distance to points in same cluster
- $b(i)$  = avg distance to points in nearest neighboring cluster

# Silhouette Interpretation

- $s(i) \approx +1$ : Well-matched to cluster
- $s(i) \approx 0$ : On border between clusters
- $s(i) \approx -1$ : Likely in wrong cluster



# Using Silhouette for Optimal k

Average Silhouette Width:

$$\bar{s} = \frac{1}{n} \sum_{i=1}^n s(i)$$

# Silhouette Procedure

1. Run clustering for different  $k$  values
2. Calculate average silhouette width for each  $k$
3. Choose  $k$  that maximizes  $\bar{s}$

**Advantage:** Provides both quality measure and optimal  $k$

# Cluster Validation

**Internal Validation** (using data only):

- Within-Cluster Sum of Squares (WCSS) - lower is better
- Silhouette Coefficient - higher is better
- Davies-Bouldin Index - lower is better
- Dunn Index - higher is better

# Cluster Validation

**External Validation** (when true labels available):

- Adjusted Rand Index (ARI)
- Normalized Mutual Information (NMI)

# Davies-Bouldin Index

Measures ratio of within-cluster dispersion to between-cluster separation

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left( \frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

# Davies-Bouldin Index

## Interpretation:

- Lower values indicate better clustering
- Compact clusters that are far apart
- Can compare different k values or methods

# Curse of Dimensionality

As dimensions ( $p$ ) increase, problems arise

# Curse of Dimensionality: Problems

1. Distance becomes less meaningful (all points appear equidistant)
2. Data becomes sparse (observations spread out)
3. Computational cost increases dramatically



# Curse of Dimensionality: Solutions

- Use PCA or feature selection before clustering
- Select only relevant variables
- Use specialized high-dimensional algorithms

**Rule:** If  $p$  is large relative to  $n$ , reduce dimensions first

# Handling Outliers

## Impact by Method:

Method	Sensitivity
K-means	High
Ward's Method	High
Single Linkage	Medium
K-medoids	Low (Robust)

# Handling Outliers: Strategies

- Pre-processing: Detect and remove outliers
- Use robust methods (k-medoids)
- Accept outlier clusters

# When to Use Hierarchical Clustering

- Small to medium datasets ( $n < 5,000$ )
- Want to explore different  $k$  values
- Need hierarchical structure
- Don't know  $k$  in advance

# When to Use K-Means

- Large datasets ( $n > 5,000$ )
- Approximately know  $k$
- Need speed and efficiency
- Clusters roughly spherical

# Cluster Analysis Workflow

1. **Define objective** - What questions to answer?
2. **Select variables** - Domain knowledge
3. **Preprocess data** - Handle missing values, outliers
4. **Standardize** - If variables on different scales
5. **Choose method** - Based on data characteristics

# Cluster Analysis Workflow

6. **Determine k** - Multiple criteria
7. **Run clustering** - Multiple times for k-means
8. **Validate results** - Internal and stability checks
9. **Interpret clusters** - Profile and name clusters
10. **Refine and iterate** - Based on insights

# Common Pitfalls to Avoid

1. **Not standardizing** when variables have different scales
2. **Using k-means** with non-spherical clusters
3. **Ignoring outliers** - can severely distort results
4. **Over-interpreting** - clustering always finds structure, even in random data



# Common Pitfalls to Avoid

5. **Using too many variables** - curse of dimensionality
6. **Running k-means once** - try multiple initializations
7. **Choosing k without validation** - use multiple methods

# Best Practices

1. **Try multiple methods** - Compare hierarchical, k-means, etc.
2. **Validate stability** - Bootstrap samples, different initializations
3. **Visualize extensively** - Scatter plots, dendrograms, parallel coordinates

# Best Practices

4. **Use domain knowledge** - Statistical metrics + practical sense
5. **Document decisions** - Why certain methods, parameters chosen
6. **Check interpretability** - Can you explain and use clusters?

# Key Takeaways: Fundamental Concepts

- Cluster analysis discovers natural groupings (unsupervised)
- Distance measures are crucial (Euclidean, Manhattan)
- Standardization essential for different scales

# Key Takeaways: Methods

- Hierarchical: Creates tree structure, multiple k values
- K-means: Fast, scalable, requires specifying k
- K-medoids: Robust alternative to k-means

# Key Takeaways: Validation

- Elbow method and silhouette analysis for optimal  $k$
- Multiple validation measures for quality assessment

# Summary: Method Selection Guide

Situation	Recommended Method
Small dataset ( $n < 1,000$ )	Hierarchical (Ward's or Average)
Large dataset ( $n > 10,000$ )	K-means with k-means++
Outliers present	K-medoids or preprocessing
Non-spherical clusters	DBSCAN or hierarchical

# Summary: Method Selection Guide

Situation	Recommended Method
Don't know k	Hierarchical, then elbow/silhouette
High dimensions	PCA first, then k-means
Mixed data types	Gower distance with hierarchical



# Advanced Topics (Beyond This Course)

## Density-Based Methods:

- DBSCAN - finds arbitrary shapes, identifies outliers

## Model-Based:

- Gaussian Mixture Models (GMM) - probabilistic approach

# Advanced Topics (Beyond This Course)

## Fuzzy Clustering:

- Soft assignment (membership degrees)

## Subspace Clustering:

- For high-dimensional data, different subspaces

# Real-World Applications: Business

## Marketing & Business:

- Customer segmentation for targeted marketing
- Product recommendation systems
- Market basket analysis

# Real-World Applications: Healthcare

## Healthcare:

- Patient stratification for personalized medicine
- Disease subtype identification
- Medical image segmentation

# Real-World Applications: Finance

## Finance:

- Fraud detection and anomaly identification
- Credit risk assessment
- Portfolio diversification

# Example: Customer Segmentation

**Scenario:** E-commerce company with 100,000 customers

## Variables:

- Purchase frequency
- Average order value
- Product category preferences
- Time since last purchase
- Customer lifetime value

## Example: Customer Segmentation Process

1. Standardize variables (different scales)
2. Try k-means for  $k = 2$  to 10
3. Use elbow method and silhouette analysis
4. Identify  $k = 5$  optimal clusters
5. Profile each segment
6. Develop targeted marketing strategies

# Recommended Resources: Books

## Textbooks:

- Everitt et al. (2011) - Cluster Analysis (5th ed.)
- James et al. (2021) - Introduction to Statistical Learning



# Recommended Resources: Software

## Software:

- Python: scikit-learn (KMeans, AgglomerativeClustering)
- R: stats package (kmeans, hclust)

# Recommended Resources: Online

## Online:

- StatQuest YouTube channel
- Scikit-learn documentation
- Coursera/edX courses on unsupervised learning

# Questions?

**Thank you for your attention!**

Juliho Castillo Colmenares  
julihocc@tec

Office: Tec de Monterrey CCM Office 1540

Office Hours: Monday-Friday, 9:00 AM - 5:00 PM

# Next Steps: This Week

## For This Week:

- Review lecture notes thoroughly
- Practice with provided examples
- Complete practice questions
- Prepare for E06 quiz

# Next Steps: Preparation for Evaluation

## Preparation for Evaluation:

- Understand distance measures and when to use each
- Know linkage methods and their properties
- Practice interpreting dendrograms
- Understand k-means algorithm and convergence
- Be able to explain validation methods