Cluster Analysis

Lecture Notes

MA2003B - Application of Multivariate Methods in Data Science

Contents

1 Introduction to Cluster Analysis

1.1 What is Cluster Analysis?

Cluster Analysis is an exploratory data analysis technique used to discover natural groupings (clusters) in data **without predefined categories**. Unlike discriminant analysis, which classifies observations into known groups, cluster analysis seeks to identify previously unknown structure in the data.

Info: **Key Distinction:** Cluster analysis is **unsupervised learning** - we don't know the "true" groups beforehand. We let the data reveal its own structure.

1.2 Primary Objectives

1. **Grouping:** Partition observations into homogeneous groups
2. **Data reduction:** Simplify large datasets by representing groups
3. **Pattern discovery:** Identify natural structure in data
4. **Hypothesis generation:** Suggest new classifications for further study

1.3 Common Applications

- **Marketing:** Customer segmentation for targeted campaigns
- **Biology:** Taxonomy and species classification
- **Medicine:** Disease subtype identification
- **Social Sciences:** Community detection in networks
- **Image Processing:** Image segmentation and compression

2 Distance and Similarity Measures

2.1 Why Distance Matters

Clustering relies on measuring how “close” or “similar” observations are to each other. The choice of distance metric fundamentally affects the clustering results.

2.2 Common Distance Measures

2.2.1 Euclidean Distance (L2 Norm)

The most commonly used distance measure, representing straight-line distance in n-dimensional space.

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

Properties:

- Sensitive to scale differences between variables
- Assumes equal importance of all dimensions
- Works best when variables are continuous and measured on similar scales

Example: For points $x = (2, 3)$ and $y = (5, 7)$:

$d(x, y) = \sqrt{(5 - 2)^2 + (7 - 3)^2} = \sqrt{9 + 16} = 5$

2.2.2 Manhattan Distance (L1 Norm)

Also called “city block” distance, it measures distance as the sum of absolute differences.

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

When to use:

- Data contains outliers or extreme values (more robust than Euclidean)
- Variables represent counts or discrete quantities
- Working in high-dimensional spaces

Example: For the same points $x = (2, 3)$ and $y = (5, 7)$:

$d(x, y) = |5 - 2| + |7 - 3| = 3 + 4 = 7$

2.2.3 Other Distance Measures

Cosine Similarity: Measures angle between vectors (useful for text data)

$$\text{similarity}(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

Correlation-based Distance: $d(x, y) = 1 - r(x, y)$

Used when pattern similarity is more important than magnitude.

Warning: **Critical:** Always standardize variables measured on different scales before computing distances! Variables with larger scales will dominate the distance calculation otherwise.

2.3 Standardization

Before clustering, transform variables to have mean 0 and standard deviation 1:

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

This ensures all variables contribute equally to distance calculations.

3 Hierarchical Clustering

3.1 Overview

Hierarchical clustering builds a tree-like structure (dendrogram) showing nested clusters at different levels of granularity. It doesn't require specifying the number of clusters in advance.

3.2 Types of Hierarchical Clustering

3.2.1 Agglomerative (Bottom-Up)

- **Start:** Each observation is its own cluster
- **Process:** Iteratively merge the two closest clusters
- **End:** All observations in one cluster
- **Most common approach**

3.2.2 Divisive (Top-Down)

- **Start:** All observations in one cluster
- **Process:** Iteratively split the most heterogeneous cluster
- **End:** Each observation is its own cluster
- **Less common, computationally intensive**

3.3 Linkage Methods

Linkage methods define how to measure distance between **clusters** (not just individual points).

3.3.1 Single Linkage (Nearest Neighbor)

Distance = **minimum** distance between any two points in the clusters

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

Advantages:

- Can detect elongated, non-spherical clusters
- Handles irregular cluster shapes

Disadvantages:

- Susceptible to "chaining effect" - clusters connect via single intermediate points
- Sensitive to noise and outliers

3.3.2 Complete Linkage (Farthest Neighbor)

Distance = **maximum** distance between any two points in the clusters

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

Advantages:

- Less sensitive to outliers than single linkage
- Tends to produce compact, spherical clusters
- More robust to noise

Disadvantages:

- May break large clusters
- Biased toward finding clusters of similar size

3.3.3 Average Linkage

Distance = **average** of all pairwise distances between clusters

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

Advantages:

- Compromise between single and complete linkage
- Less affected by outliers than single linkage
- Generally produces good results

3.3.4 Ward's Method

Minimizes the **total within-cluster sum of squares** (variance)

At each step, merge the two clusters that result in the smallest increase in total within-cluster variance.

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

Advantages:

- Tends to create compact, equal-sized clusters
- Minimizes variance within clusters
- Often produces the best results in practice

Disadvantages:

- Assumes spherical clusters
- Sensitive to outliers

Tip: **Practical Recommendation:** Ward's method is often the best choice for exploratory analysis. Use average linkage as an alternative. Avoid single linkage unless you specifically need to detect elongated clusters.

3.4 Dendrograms

A dendrogram visualizes the hierarchical clustering process as a tree diagram.

Reading a Dendrogram:

- Horizontal axis: Individual observations or clusters
- Vertical axis: Distance or dissimilarity at which clusters merge
- Height of branches: Indicates the distance between merged clusters

Determining Number of Clusters: Look for large vertical gaps (jumps in distance) - cut the dendrogram where there's a substantial increase in fusion distance.

Info: **Cutting the Dendrogram:** Draw a horizontal line across the dendrogram. The number of vertical lines it crosses equals the number of clusters at that level.

3.5 Chaining Effect

The “chaining effect” occurs when clusters form long, elongated chains rather than compact groups. This happens with single linkage when observations connect via intermediate points.

Example scenario: Observations A-B-C-D form a chain where each is close to its neighbor, but A and D are far apart. Single linkage would group them together.

4 Non-Hierarchical Clustering

4.1 K-Means Clustering

The most popular non-hierarchical method. Partitions data into k clusters by minimizing within-cluster variance.

4.1.1 Algorithm

1. **Initialize:** Randomly select k observations as initial cluster centers (centroids)
2. **Assignment:** Assign each observation to the nearest centroid
3. **Update:** Recalculate centroids as the mean of all points in each cluster
4. **Repeat:** Steps 2-3 until assignments no longer change (convergence)

4.1.2 Mathematical Objective

Minimize the total within-cluster sum of squares (WCSS):

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

where μ_i is the centroid of cluster C_i.

4.1.3 Convergence

The algorithm **always converges** because:

- Each iteration reduces or maintains WCSS
- There are finitely many possible partitions
- Typically converges in 10-30 iterations

Convergence is reached when cluster assignments no longer change between iterations.

4.1.4 Advantages

- **Fast:** Computational complexity O(n × k × p × iterations)
- **Scalable:** Works well with large datasets
- **Simple:** Easy to understand and implement
- **Efficient:** Generally converges quickly

4.1.5 Limitations

1. **Requires specifying k in advance** - must know or guess number of clusters
2. **Sensitive to initialization** - different starting points may yield different results
3. **Assumes spherical clusters** - struggles with elongated or irregular shapes
4. **Sensitive to outliers** - outliers can distort centroids
5. **Assumes equal cluster sizes** - tends to create similar-sized clusters

4.1.6 K-Means++ Initialization

An improved initialization method that spreads out initial centroids:

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

Benefit: Significantly reduces the risk of poor initialization and typically produces better results.

4.2 K-Medoids (PAM - Partitioning Around Medoids)

Similar to k-means but uses actual data points as cluster centers (medoids) instead of computed means.

4.2.1 Key Difference from K-Means

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

4.2.2 Advantages

- **More robust to outliers** - not affected by extreme values
- **Works with any distance metric** - not limited to Euclidean distance
- **Interpretable centers** - medoids are actual observations

4.2.3 Disadvantages

- **Slower:** Higher computational complexity than k-means
- **Less common:** Not as widely implemented in software

Tip: When to use K-medoids: Choose k-medoids when your data contains outliers or when you need cluster centers to be actual observations (e.g., choosing representative customers).

5 Determining Optimal Number of Clusters

5.1 The Fundamental Challenge

Unlike supervised learning, we don't have a "ground truth" for the correct number of clusters. We must use heuristics and domain knowledge.

5.2 Elbow Method

Plot the total within-cluster sum of squares (WCSS) against the number of clusters k.

Procedure:

1. Run clustering for k = 1, 2, 3, ..., K_max
2. Calculate WCSS for each k
3. Plot WCSS vs. k
4. Look for an "elbow" - the point where adding more clusters yields diminishing returns

$$\text{WCSS}(k) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Interpretation:

- WCSS always decreases as k increases
- The "elbow" indicates where additional clusters don't substantially improve fit
- Choose k at the elbow point

Warning: **Limitation:** The elbow is not always clear - sometimes the curve is smooth with no obvious bend.

5.3 Silhouette Analysis

Measures how well each point fits within its assigned cluster compared to other clusters.

5.3.1 Silhouette Coefficient

For each observation i:

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

where:

- $a(i)$ = average distance to other points in the same cluster (cohesion)
- $b(i)$ = average distance to points in the nearest neighboring cluster (separation)

Range: $s(i) \in [-1, +1]$

Interpretation:

- $s(i) \approx +1$: Point is well-matched to its cluster (far from neighbors)
- $s(i) \approx 0$: Point is on the border between clusters
- $s(i) \approx -1$: Point may be assigned to the wrong cluster

5.3.2 Average Silhouette Width

Calculate the mean silhouette coefficient across all observations:

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

Usage: Run clustering for different values of k and choose the k that maximizes the average silhouette width.

Info: **Silhouette Plot:** Create a bar chart showing silhouette coefficients for each observation, grouped by cluster. Wide, tall bars indicate good clustering.

5.4 Other Methods

5.4.1 Gap Statistic

Compares WCSS to its expected value under a null reference distribution (random data).

Idea: Choose k where the gap between observed and expected WCSS is largest.

5.4.2 Davies-Bouldin Index

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

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

Lower values indicate better clustering - compact clusters that are far apart.

5.4.3 Domain Knowledge

Often the most important factor:

- Business requirements (e.g., “we need 3-5 customer segments for our marketing capacity”)
- Interpretability (fewer clusters are easier to explain)
- Actionability (can we act differently for each cluster?)

6 Validation and Quality Assessment

6.1 Internal Validation

Measures clustering quality using only the data itself (no external labels).

6.1.1 Within-Cluster Sum of Squares (WCSS)

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

Lower is better - indicates tight, compact clusters.

Limitation: Always decreases with more clusters, reaching 0 when k = n.

6.1.2 Between-Cluster Sum of Squares (BCSS)

$$\text{BCSS} = \sum_{i=1}^{k} n_i \, \|\mu_i - \mu\|^2$$

where μ is the overall data mean.

Higher is better - indicates well-separated clusters.

6.1.3 Silhouette Coefficient

Already discussed - ranges from -1 to $+1$, higher is better.

6.1.4 Davies-Bouldin Index

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

Lower is better - indicates compact, well-separated clusters.

6.1.5 Dunn Index

$$\text{Dunn} = \frac{\min_{i \neq j} d(C_i, C_j)}{\max_k \text{diameter}(C_k)}$$

Ratio of minimum inter-cluster distance to maximum cluster diameter.

Higher is better - indicates well-separated, compact clusters.

6.2 External Validation

When “true” labels are available (e.g., for method comparison).

6.2.1 Adjusted Rand Index (ARI)

Measures agreement between clustering results and true labels, adjusted for chance.

Range: [−1, +1]

- 1 = perfect agreement
- 0 = random clustering
- Negative = worse than random

6.2.2 Normalized Mutual Information (NMI)

Measures information shared between clustering and true labels.

Range: [0, 1], higher is better.

7 Practical Considerations

7.1 Curse of Dimensionality

As the number of dimensions (variables) increases:

1. **Distance becomes less meaningful** - all points appear equidistant in high dimensions
2. **Data becomes sparse** - observations spread out, making clusters harder to find
3. **Computational cost increases** - both time and memory requirements grow

Solutions:

- **Dimensionality reduction:** Use PCA or feature selection before clustering
- **Feature selection:** Choose only relevant variables
- **Specialized methods:** Use algorithms designed for high dimensions

Warning: **Rule of Thumb:** If p (dimensions) is large relative to n (observations), consider dimensionality reduction before clustering.

7.2 Standardization and Scaling

7.2.1 Why Standardize?

Variables measured on different scales will dominate distance calculations.

Example: Age (20-80) vs. Income (20,000-200,000)

- Without standardization, income dominates
- Euclidean distance primarily reflects income differences

7.2.2 When to Standardize

Always standardize when:

- Variables have different units (e.g., kg, cm, years)
- Variables have different scales (e.g., 0-1 vs. 0-1000)
- You want all variables to contribute equally

Consider not standardizing when:

- All variables have the same units and scale
- Scale differences are meaningful (e.g., different importance)
- Using correlation-based distances

7.2.3 Standardization Methods

Z-score standardization:

$$z = \frac{x - \mu}{\sigma}$$

Min-max scaling:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

7.3 Handling Mixed Data Types

7.3.1 Continuous + Categorical Variables

Option 1: Gower Distance

Combines different distance measures for different variable types:

$$d_{\text{Gower}(i,j)} = \frac{\sum_{k=1}^{p} \delta_{ijk} d_{ijk}}{\sum_{k=1}^{p} \delta_{ijk}}$$

where:

- For continuous: $d_{ijk} = |x_{ik} - x_{jk} \frac{|}{\text{range}_k}$
- For categorical: $d_{ijk} = 0$ if same, 1 if different
- $\delta_{ijk} = 1$ if comparison is valid, 0 if missing

Option 2: Dummy Encoding

Convert categorical variables to binary dummy variables, then use standard distance measures.

Caution: This changes the geometry of the data.

7.3.2 Pure Categorical Data

Use specialized distance measures:

- **Matching coefficient:** Proportion of matching categories
- **Jaccard distance:** For binary data
- **Hamming distance:** Number of position differences

7.4 Outliers and Robustness

7.4.1 Impact of Outliers

- **K-means:** Highly sensitive - outliers pull centroids away from dense regions
- **Hierarchical (Ward's):** Sensitive - outliers inflate variance
- **Single linkage:** Moderately sensitive - may create singleton clusters
- **K-medoids:** Most robust - medoids are actual points

7.4.2 Strategies for Handling Outliers

1. **Pre-processing:** Detect and remove outliers before clustering
2. **Robust methods:** Use k-medoids or other robust algorithms
3. **Outlier clusters:** Accept that some clusters may be outliers
4. **Trimming:** Use trimmed versions of distance measures

7.5 Computational Considerations

7.5.1 Complexity

- **K-means:** O(n × k × p × iterations) - very efficient
- **Hierarchical:** O(n^2 × p) for distance matrix, O(n^3) for some linkages
- **K-medoids:** O($k(n-k)^2$ × p) - slower than k-means

7.5.2 Scalability Recommendations

- **Small datasets (n < 1,000):** Any method works
- **Medium datasets (n = 1,000-10,000):** K-means or k-medoids

- **Large datasets (n > 10,000):** K-means with sampling, or specialized algorithms
- **Very large datasets (n > 1M):** Mini-batch k-means, approximate methods

8 Cluster Analysis Workflow

8.1 Step-by-Step Procedure

8.1.1 1. Problem Definition

- Define the objective: What questions are we trying to answer?
- Determine what makes a "good" cluster in your context
- Consider how results will be used

8.1.2 2. Variable Selection

- Choose relevant variables (domain knowledge)
- Remove redundant variables (high correlation)
- Consider creating derived features
- Decide whether to include or exclude certain variables

8.1.3 3. Data Preprocessing

Handle Missing Values:

- Imputation
- Deletion
- Special clustering methods for incomplete data

Deal with Outliers:

- Detect using box plots, z-scores, Mahalanobis distance
- Decide: remove, transform, or use robust methods

Transform Variables:

- Log transform for skewed distributions
- Square root for count data
- Box-Cox transformations

Standardize:

- Z-score standardization if variables on different scales
- Consider domain knowledge about variable importance

8.1.4 4. Choose Clustering Method

Hierarchical if:

- Small to medium dataset
- Want to explore different numbers of clusters
- Need hierarchical structure (taxonomy)
- Don't know k in advance

K-means if:

- Large dataset
- Approximately know the number of clusters
- Need speed and efficiency
- Clusters are roughly spherical

K-medoids if:

- Outliers present
- Need cluster centers to be actual observations

- Can afford extra computation

Other methods if:

- Non-spherical clusters (DBSCAN)
- Mixed data types (k-prototypes)
- High dimensions (subspace clustering)

8.1.5 5. Determine Number of Clusters

- Use elbow method
- Calculate silhouette coefficients
- Try multiple values of k
- Consider business constraints
- Validate interpretability

8.1.6 6. Run Clustering

- For k-means: run multiple times with different initializations
- For hierarchical: create dendrogram and examine
- Save cluster assignments and centers

8.1.7 7. Validate Results

Internal validation:

- Check silhouette coefficients
- Examine within/between cluster variance
- Look for negative silhouette values

Stability:

- Re-run with bootstrap samples
- Check if clusters are consistent

Interpretability:

- Can you describe each cluster clearly?
- Do clusters make business sense?

8.1.8 8. Interpret and Profile Clusters

- Examine cluster centers/medoids
- Calculate cluster means for each variable
- Create visualizations (parallel coordinates, scatter plots)
- Name/label clusters based on characteristics
- Identify distinguishing features of each cluster

8.1.9 9. Validation and Iteration

- Test on new data if possible
- Refine based on domain expert feedback
- Consider alternative numbers of clusters
- Try different methods or distance measures

8.1.10 10. Report and Act

- Document methodology and rationale
- Present cluster profiles clearly
- Provide actionable insights

- Discuss limitations and next steps

9 Advanced Topics

9.1 Density-Based Clustering

9.1.1 DBSCAN (Density-Based Spatial Clustering)

Unlike k-means, DBSCAN can find clusters of arbitrary shape and identify outliers.

Key Parameters:

- ε: neighborhood radius
- MinPts: minimum points to form dense region

Advantages:

- Finds non-spherical clusters
- Robust to outliers (marks them as noise)
- No need to specify number of clusters

Disadvantages:

- Struggles with varying densities
- Sensitive to parameters
- Higher computational cost

9.2 Fuzzy Clustering

Each observation has a degree of membership in each cluster (soft assignment) rather than belonging to exactly one cluster (hard assignment).

Fuzzy C-means:

- Membership values range from 0 to 1
- Memberships for each point sum to 1
- Useful when boundaries between clusters are unclear

9.3 Model-Based Clustering

Assumes data comes from a mixture of probability distributions (typically Gaussian).

Gaussian Mixture Models (GMM):

- Each cluster is a Gaussian distribution
- Use EM algorithm to estimate parameters
- Provides probabilistic cluster assignments
- Can estimate optimal k using BIC or AIC

9.4 Subspace and Projected Clustering

For high-dimensional data, different clusters may exist in different subspaces (subsets of variables).

Approaches:

- Find clusters in different variable subsets
- Identify relevant dimensions for each cluster
- Useful when curse of dimensionality is severe

10 Common Pitfalls and Best Practices

10.1 Common Mistakes to Avoid

Warning: **Pitfall 1: Not standardizing variables**

Variables with larger scales will dominate distance calculations, leading to meaningless clusters based primarily on scale rather than pattern.

Warning: **Pitfall 2: Using k-means with non-spherical clusters**

K-means assumes spherical clusters and will force non-spherical data into inappropriate spherical groups.

Warning: **Pitfall 3: Ignoring outliers**

Outliers can severely distort clustering results, especially with k-means and Ward's method.

Warning: **Pitfall 4: Over-interpreting results**

Clustering will always find some structure, even in random data. Validate that clusters are meaningful and stable.

Warning: **Pitfall 5: Using too many variables**

High dimensionality makes clustering difficult. Focus on relevant, non-redundant variables.

10.2 Best Practices

Tip: **Practice 1: Try multiple methods**

Compare hierarchical, k-means, and other approaches. Robust clusters will appear across methods.

Tip: **Practice 2: Validate stability**

Run clustering multiple times with different initializations or bootstrap samples. Good clusters should be consistent.

Tip: **Practice 3: Visualize extensively**

Use scatter plots, parallel coordinates, heatmaps, and dendrograms to understand cluster structure.

Tip: **Practice 4: Use domain knowledge**

Statistical metrics are important, but clusters must make practical sense in your domain.

Tip: Practice 5: Document decisions

Record why you chose certain methods, parameters, and number of clusters. Clustering involves many subjective choices.

11 Summary and Key Takeaways

11.1 Fundamental Concepts

1. **Cluster analysis discovers natural groupings** in data without predefined categories

2. **Distance measures are crucial** - choice affects results fundamentally:
   - Euclidean for continuous, similar scales
   - Manhattan for robustness to outliers
   - Correlation for pattern similarity

3. **Standardization is essential** when variables have different scales

11.2 Clustering Methods

Hierarchical Clustering:

- Creates tree structure (dendrogram)
- Don't need to specify k in advance
- Linkage method matters:
  - Single: can find elongated clusters, prone to chaining
  - Complete: compact clusters, robust to outliers
  - Average: good compromise
  - Ward's: minimizes variance, often best results

K-means:

- Fast, scalable, widely used
- Requires specifying k
- Assumes spherical clusters
- Sensitive to initialization and outliers
- Use k-means++ for better initialization

K-medoids:

- More robust to outliers than k-means
- Centers are actual data points
- Slower than k-means

11.3 Determining Optimal k

1. **Elbow method:** Look for bend in WCSS vs. k plot
2. **Silhouette analysis:** Choose k that maximizes average silhouette width
3. **Domain knowledge:** Consider business requirements and interpretability
4. **Try multiple values:** Compare and validate

11.4 Validation

- **Silhouette coefficient:** $+1$ = perfect, 0 = borderline, -1 = wrong cluster
- **Davies-Bouldin Index:** Lower is better
- **Stability:** Rerun with different seeds/samples
- **Interpretability:** Can you explain and use the clusters?

11.5 Practical Workflow

1. Define objective and select variables

2. Preprocess: handle missing values, outliers, transform, standardize
3. Choose method based on data characteristics and goals
4. Determine k using multiple criteria
5. Run clustering (multiple times if k-means)
6. Validate and interpret results
7. Refine and iterate as needed

11.6 Critical Considerations

- **Curse of dimensionality:** Reduce dimensions if p is large
- **Outliers:** Detect and handle appropriately
- **Mixed data types:** Use specialized distances or preprocessing
- **Scale matters:** Always consider whether to standardize
- **No "true" answer:** Clustering is exploratory; validate and iterate

12 Practice Questions

Test your understanding with these questions based on the lecture material:

12.1 Conceptual Questions

1. What is the fundamental difference between cluster analysis and discriminant analysis?
2. Why is standardization important before clustering? Give an example where lack of standardization would cause problems.
3. Explain the “chaining effect” in hierarchical clustering. Which linkage method is most susceptible to it?
4. What are the main advantages and disadvantages of k-means clustering?
5. How does the silhouette coefficient help determine if a point is well-clustered?

12.2 Application Questions

6. You have customer data with variables: age (years), income (dollars), purchases (count), and loyalty_score (0-100). Should you standardize before clustering? Why?
7. Your elbow plot shows a smooth curve with no clear “elbow”. What should you do?
8. You run k-means with k=3 five times and get different results each time. What might explain this, and how can you address it?
9. Your dataset has 50,000 observations and 20 variables. Which clustering method would you recommend and why?
10. After clustering customers, you find one cluster has 5,000 members and another has only 50. Is this a problem? What might it indicate?

12.3 Interpretation Questions

11. What does it mean if a point has a silhouette coefficient of -0.3?
12. You get a Davies-Bouldin Index of 0.5 for k=3 and 0.8 for k=5. Which is better?
13. In a dendrogram, you see a large vertical jump between 3 and 4 clusters. What does this suggest?
14. Your average silhouette width is 0.25. Is this good clustering?
15. K-means gives you compact, equal-sized clusters, but hierarchical clustering with average linkage gives very different results. What should you do?

13 Answers to Practice Questions

13.1 Conceptual Answers

1. **Cluster analysis** discovers unknown groupings (unsupervised), while **discriminant analysis** classifies into predefined groups (supervised).
2. Standardization ensures all variables contribute equally to distance. Example: Without standardization, income (0-200,000) would dominate over age (0-100) in Euclidean distance calculations.
3. Clusters form long chains rather than compact groups when similar observations link via intermediate points. **Single linkage** is most susceptible.
4. **Advantages:** Fast, scalable, simple, works well with large data. **Disadvantages:** Requires knowing k, sensitive to initialization, assumes spherical clusters, sensitive to outliers.
5. Silhouette close to +1 means the point is much closer to its own cluster than to neighboring clusters (well-clustered). Close to 0 means borderline between clusters. Negative means likely in wrong cluster.

13.2 Application Answers

6. **Yes, standardize.** Income (thousands) would dominate age, purchases, and loyalty_score if not standardized, making the clustering essentially based only on income.
7. Use alternative methods: (1) Silhouette analysis, (2) Gap statistic, (3) Domain knowledge about appropriate number of segments, (4) Try multiple k values and compare interpretability.
8. Different **initializations** lead to different local optima. Solutions: (1) Run multiple times and choose best result (lowest WCSS), (2) Use k-means++ initialization, (3) Compare stability across runs.
9. **K-means** - hierarchical clustering would require O(n^2) distance matrix (50,000 × 50,000 = 2.5 billion calculations). K-means scales much better to large n.
10. Not necessarily a problem. Small cluster might be: (1) Valuable niche segment (VIP customers), (2) Outliers that should be investigated, or (3) Indication that k is too high. Examine cluster characteristics to determine.

13.3 Interpretation Answers

11. The point is **likely in the wrong cluster** - it's closer to a neighboring cluster than to its assigned cluster. Consider it misclassified.
12. **k=3 is better.** Lower Davies-Bouldin Index indicates more compact, better-separated clusters ($0.5 < 0.8$).
13. Suggests that **k=3 is optimal** - merging from 3 to 4 clusters causes a large increase in within-cluster heterogeneity, indicating natural separation at 3 clusters.
14. **Moderate clustering.** Silhouette width of 0.25 indicates some structure but not strong separation. Values 0.2-0.5 suggest reasonable but not excellent clustering. Consider if this is acceptable for your application.
15. **Compare both results carefully:** (1) Examine cluster profiles and interpretability, (2) Test stability with bootstrap samples, (3) Validate with domain knowledge, (4) Try intermediate methods like k-medoids. Different methods capture different aspects of structure - neither is necessarily "wrong."

14 Additional Resources

14.1 Recommended Reading

Textbooks:

- Everitt, B., et al. (2011). *Cluster Analysis* (5th ed.). Wiley.
- Kaufman, L., & Rousseeuw, P. J. (2005). *Finding Groups in Data*. Wiley.
- James, G., et al. (2021). *An Introduction to Statistical Learning* (2nd ed.). Springer. Chapter 12.

Papers:

- Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. *SODA '07*.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation. *Journal of Computational and Applied Mathematics*, 20, 53-65.

14.2 Software Implementations

Python:

- scikit-learn: KMeans, AgglomerativeClustering, DBSCAN
- scipy.cluster: Hierarchical clustering and dendrograms
- sklearn.metrics: Silhouette, Davies-Bouldin scores

R:

- stats: kmeans(), hclust(), dist()
- cluster: pam(), silhouette()
- factoextra: Visualization tools

14.3 Online Resources

- Scikit-learn Clustering Documentation: comprehensive guide with examples
- StatQuest YouTube: Visual explanations of clustering concepts
- Towards Data Science: Practical tutorials and case studies

End of Lecture Notes

Good luck with your studies!