Chapter 9: Unsupervised Learning Algorithms

Your Name

Your Institution

June 30, 2025

Introduction to Unsupervised Learning

What is Unsupervised Learning?

Unsupervised learning is a type of machine learning that identifies patterns in data without labeled outcomes. It explores the inherent structure of data.

Importance of Unsupervised Learning

- **Discover Hidden Patterns**: Helps in discovering the underlying structure in data without pre-existing labels.
- **Data Preprocessing**: Useful for dimensionality reduction and feature extraction.
- **Anomaly Detection**: Identifies outliers or unusual observations that do not conform to expected patterns.

Applications of Unsupervised Learning

1. **Customer Segmentation**
   - Businesses classify customers based on purchasing behavior to tailor marketing strategies.
   - *Example*: An online retailer grouping buyers who purchase similar items.
2. **Market Basket Analysis**
   - Finding relationships between items purchased together.
   - *Example*: Customers who buy bread often also buy butter.
3. **Image Compression**
   - Reduces the size of image files by identifying similar pixels.
   - *Example*: Clustering techniques group similar pixel values.
4. **Text Analysis and Topic Modeling**
   - Grouping documents into topics without explicit labels.
   - *Example*: Algorithms like Latent Dirichlet Allocation (LDA) categorize articles.

Common Algorithms in Unsupervised Learning

- **K-Means Clustering**
  - Partitions n observations into k clusters based on feature similarity.
  - *Formula*: Minimize the sum of squared distances between points and centroid.
- **Hierarchical Clustering**
  - Builds a hierarchy of clusters using agglomerative or divisive approaches.
- **Principal Component Analysis (PCA)**
  - Reduces dimensionality while preserving variance for visualization and analysis.

Conclusion

Key Takeaways

- Unsupervised learning does not rely on labeled data.
- It is essential for exploratory data analysis.
- Empowers organizations to make data-driven decisions by uncovering new insights.

What is Clustering?

Definition of Clustering

Clustering is an unsupervised learning technique that aims to group a set of objects such that objects in the same group (or cluster) are more similar to each other than to those in other groups.

- Similarity can be defined using various metrics (e.g., Euclidean, Manhattan).

Role in Unsupervised Learning

- **Data Exploration:** Understanding structure when labeled outcomes are not available, revealing inherent groupings.
- **Pre-processing:** Used as a preliminary step in other machine learning tasks to simplify datasets.
- **Anomaly Detection:** Identifying unusual data points that do not fit any cluster.

Differences Between Clustering and Classification

1. **Objective:**
   - Clustering: No predefined labels; goal is to find structures or patterns.
   - Classification: Uses labeled data to categorize instances into predefined categories.
2. **Supervision:**
   - Clustering: Unsupervised; no prior knowledge about group definitions.
   - Classification: Supervised; relies on a training dataset with existing labels.
3. **Methods:**
   - Clustering: K-means, hierarchical clustering, DBSCAN.
   - Classification: Decision trees, random forests, support vector machines.
4. **Application:**
   - Clustering: Customer segmentation, image compression, organizing computing clusters.
   - Classification: Spam detection, sentiment analysis, medical diagnosis.

Example: Clustering vs Classification

Example

Consider a dataset of various fruits with attributes like color, weight, and sweetness:

- **Clustering:** Discovering distinct clusters; e.g., citrus fruits (oranges, lemons) vs. berries (strawberries, blueberries).
- **Classification:** Using a labeled dataset to learn distinguishing features and categorize new unlabeled fruits.

Key Points

- Clustering is vital for exploratory data analysis.
- Unlike classification, clustering does not require predefined labels.
- It plays a significant role in various real-world applications, providing insights and informing further analysis.

Types of Clustering Techniques - Overview

Introduction to Clustering Techniques

Clustering is an unsupervised learning method that groups a set of objects such that objects in the same group (or cluster) are more similar to each other than to those in other groups.

- **Partitioning Methods**: Divide the dataset into distinct non-overlapping clusters.
- **Hierarchical Methods**: Build a hierarchy of clusters using either a bottom-up (agglomerative) or top-down (divisive) approach.

Types of Clustering Techniques - Partitioning Methods

1. Partitioning Methods

Partitioning methods create distinct non-overlapping subsets (clusters) so that each data point is assigned to exactly one cluster.

- **Key Characteristics**:
  - Each cluster is represented by a centroid.
  - The process is iterative, refining clusters until convergence.
- **Example: K-Means Clustering**
  1. Initialize: Select K initial centroids randomly.
  2. Assign: Allocate each data point to the nearest centroid.
  3. Update: Recalculate the centroid based on assigned points.
  4. Repeat: Continue until centroids do not change significantly.
- **Formula**:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{1}$$

Where:

- J = total within-cluster variance

Types of Clustering Techniques - Hierarchical Methods

2. Hierarchical Methods

Hierarchical clustering creates a hierarchy of clusters through either a bottom-up (agglomerative) or top-down (divisive) approach.

- **Key Characteristics**:
  - Result is represented as a dendrogram (tree structure).
  - Clusters can be formed by cutting the dendrogram at certain levels.
- **Example: Agglomerative Clustering**
  1. Initial State: Treat each point as a single cluster.
  2. Combine: Merge closest pairs based on a distance metric.
  3. Repeat: Continue until all points form a single cluster or desired clusters are reached.
- **Common Distance Metrics**:
  - Euclidean distance: $\sqrt{\sum (x_i - y_i)^2}$
  - Manhattan distance: $\sum |x_i - y_i|$

K-Means Clustering Overview

K-Means Clustering is a widely-used unsupervised learning algorithm aimed at partitioning data into K distinct clusters based on the features of the data points.

- Grouping similar data points together
- Maximizing the distance between different clusters

Steps of the K-Means Algorithm

1. **Initialization**:
   - Choose number of clusters, K
   - Randomly select K initial centroids from data points
2. **Assignment Step**:
   - Calculate distance to each centroid and assign points to nearest centroid
   - Distance formula:

$$D(x_i, C_j) = \sqrt{\sum_{k=1}^{n} (x_i^{(k)} - C_j^{(k)})^2} \tag{2}$$

3. **Update Step**:
   - Recalculate centroids as the mean of assigned points
   - New centroid:

$$C_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i \tag{3}$$

4. **Convergence Check**:
   - Repeat assignment and update until centroids stabilize

Example of K-Means Clustering

Consider the following data points in a 2D space:

- (1, 2), (1, 4), (1, 0), (10, 2), (10, 4), (10, 0)
- **Choosing K**: Assume $K = 2$
- **Initial Centroids**: Randomly select (1, 2) and (10, 2)
- **Assignment**: Assign each point to the nearest centroid
- **Update**: Compute new centroids based on assigned points
- Repeat until centroids stabilize

Key Points to Consider

- **Distance Metrics**: Primarily uses Euclidean distance; other metrics can be applied based on data characteristics.
- **Random Initialization**: Different initial centroids may lead to varying results; it is advisable to run multiple iterations.
- **Applications**: Commonly used in market segmentation, document clustering, image compression, etc.

Visual Representation

Consider including a simple scatterplot illustrating:

- Initial centroids
- Data points
- Final clusters after convergence

This visual helps in understanding the clustering process intuitively.

Advantages and Disadvantages of K-Means - Overview

K-Means is a popular unsupervised learning algorithm used for clustering data into K distinct groups based on feature similarities. It's widely recognized for its:

- Simplicity
- Computational efficiency
- Versatility in various applications

Advantages of K-Means

1. **Computational Efficiency**:
   - Time complexity: $O(n \cdot K \cdot i)$
     - n = number of data points
     - K = number of clusters
     - i = number of iterations until convergence
2. **Simplicity and Ease of Implementation**:
   - Steps: Initialize K centroids, assign points, update centroids until convergence.
3. **Versatility**:
   - Applicable for customer segmentation, image compression, etc.
4. **Scalability**:
   - Handles large datasets efficiently, suitable for real-time applications.
5. **Deterministic**:
   - Same initial conditions yield the same clustering result.

Disadvantages of K-Means

1. **Choice of K**:
   - The required number of clusters K must be specified beforehand.
2. **Sensitivity to Initialization**:
   - Initialization affects final clusters; K-Means++ for better initial formation.
3. **Assumes Spherical Clusters**:
   - Assumption may not hold for all datasets.
4. **Outlier Sensitivity**:
   - Sensitive to outliers, skewing centroids and misassigning clusters.
5. **Not Suitable for Non-Convex Shapes**:
   - Struggles with clusters that aren't convex in shape.

Key Points and Conclusion

Key Points:

- K-Means is efficient and simple but requires careful consideration of K and initialization.
- Understand limitations related to data structure (shape, outliers).

Example: Consider a dataset of customer purchase behavior:

- **Ideal Situation**: K-Means groups similar customers for targeted marketing.
- **Challenging Situation**: Incorrect K leads to ineffective strategies.

Conclusion: K-Means is powerful when applied correctly, but understanding its limitations is crucial for effective clustering.

Hierarchical Clustering

Overview

Hierarchical clustering is an unsupervised learning technique that organizes data into nested clusters based on their characteristics. This method is fundamentally divided into two approaches:

- Agglomerative (bottom-up)
- Divisive (top-down)

Agglomerative Approach

Definition

The agglomerative method starts with each data point as its own cluster and merges them into larger clusters.

Process

1. Start with n clusters (each data point is its own cluster).
2. Calculate distances between all pairs of clusters using a metric (e.g., Euclidean distance).
3. Merge the two closest clusters.
4. Repeat until only one cluster remains or a designated number is reached.

Linkage Methods

- Single Linkage: Distance between closest points.
- Complete Linkage: Distance between farthest points.

Divisive Approach

Definition

The divisive method starts with all data points in one cluster and recursively splits them into smaller clusters.

Process

1. Start with all items in one cluster.
2. Select a cluster to split based on a criterion (e.g., variance).
3. Split the cluster into two sub-clusters.
4. Repeat until each cluster contains a single item or designated number is reached.

Key Points to Emphasize

- No predefined number of clusters.
- Results can be represented using dendrograms.
- Computationally intensive for larger datasets.

Distance Calculation and Example Code

Distance Calculation (Euclidean Distance)

For two points $A(x_1, y_1)$ and $B(x_2, y_2)$:

$$\text{Distance}(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4}$$

Python Code Sample

```
from sklearn.cluster import AgglomerativeClustering
import numpy as np

# Sample Data
data = np.array([[1, 2], [1, 4], [1, 0], [4, 2], [4,
    4], [4, 0]])

# Applying Agglomerative Clustering
model = AgglomerativeClustering(n_clusters=2, linkage=
    'ward')
clusters = model.fit_predict(data)
```

Dendrogram Representation: Introduction to Dendrograms

A **dendrogram** is a tree-like diagram used to visualize the arrangement of clusters formed during hierarchical clustering. It provides a graphical representation of the data's clustering process, illustrating how clusters are formed step-by-step or bottom-up.

Dendrogram Representation: How Dendrograms Work

- **Hierarchical Clustering**: The dendrogram results from hierarchical clustering, either *agglomerative* (bottom-up) or *divisive* (top-down).
- In *agglomerative clustering*, each data point starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

Dendrogram Representation: Interpretation of a Dendrogram

1. **Branches**: Each branch represents a cluster of points that have been merged together.
2. **Height**: The height indicates the distance (or dissimilarity) at which clusters are merged.
3. **Leaf Nodes**: The endpoints of the branches (leaf nodes) represent individual data points or observations.

Dendrogram Representation: Example

Consider a dataset with 5 data points. When plotting the dendrogram for this set:

- At the bottom, you would see 5 leaves, each representing one data point.
- Clusters might merge: points A and B at height 1.5, points C and D at height 2.0.
- One cluster containing all points merges at height 4.0.

Dendrogram Representation: Determining the Number of Clusters

A key functionality of a dendrogram is to help decide the appropriate number of clusters. This can be done by:

- **Cutting the Dendrogram**: A horizontal line can be drawn across the dendrogram to determine how many clusters exist below that line.
- **Elbow Method**: Look for "elbows" where the distance between merges significantly increases, indicating a natural division between clusters.

Dendrogram Representation: Summary and Key Points

Key Points to Emphasize

- Dendrograms provide a visual means of evaluating and understanding cluster structures.
- Height in the dendrogram represents dissimilarity; higher branches indicate more distinct clusters.
- An optimal number of clusters can be determined visually by assessing where to cut the dendrogram.

Dendrograms are a powerful tool in hierarchical clustering that visually illustrates the clustering process and aids in determining an appropriate number of clusters.

Dendrogram Representation: Visual Aid Suggestion

Consider including a simple dendrogram sketch in your presentation. Label the height of clusters and show potential cut lines for cluster determination.

Evaluation of Clustering Results

Overview

Evaluating clustering performance is essential for determining the quality of formed clusters. Since clustering is an unsupervised learning method, traditional metrics like accuracy are not applicable. We discuss two significant metrics: the **Silhouette Score** and the **Davies-Bouldin Index**.

Silhouette Score

- **Definition**: Measures how similar an object is to its own cluster versus other clusters. Range is from -1 to 1:
  - **1**: Well-clustered points
  - **0**: Points on the boundary
  - **-1**: Points possibly misclassified
- **Formula**:

$$s = \frac{b - a}{\max(a, b)} \tag{5}$$

Where:

- a: Average distance to points in the same cluster
- b: Average distance to points in the nearest cluster

Davies-Bouldin Index

- **Definition**: Evaluates cluster separation and compactness. A lower index indicates better clustering.
- **Formula**:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right) \tag{6}$$

Where:

- k: Number of clusters
- s_i: Average distance of points in cluster i to its centroid (compactness)
- d_{ij}: Distance between centroids of clusters i and j (separation)

Real-World Applications of Clustering - Introduction

- Clustering is an unsupervised learning technique that groups a set of objects based on similarity.
- Provides insights facilitating data-driven decision-making across various sectors.
- Applications span fields such as marketing, social networking, and image processing.

Real-World Applications of Clustering - Key Applications

1. **Market Segmentation**
   - Divides a market into distinct buyer groups with different needs.
   - Example: Retail company uses K-means to segment customers by purchasing behavior.
   - **Benefits**: Enhanced satisfaction and increased sales.
2. **Social Network Analysis**
   - Examines social structures using network and graph theories.
   - Example: Platforms like Facebook group users based on interactions.
   - **Benefits**: Better understanding of user behavior and targeted advertising.
3. **Image Processing**
   - Groups pixels based on color similarity or intensity in images.
   - Example: K-means segmentation separates objects in photographs.
   - **Benefits**: Advanced editing, object recognition, and video analysis.

Real-World Applications of Clustering - Conclusion

- Clustering techniques enable insights from large datasets.
- Enhances strategic decision-making across various domains.
- Summarized Key Points:
  - Powerful for identifying natural groupings in data.
  - Utilized widely in marketing, networking, and image processing sectors.
  - Promotes improved customer experiences and informed business strategies.

K-means Clustering Example

Pseudocode

```
# K-means clustering example
def k_means(data, k):
    centroids = initialize_centroids(data, k)
    while not converged:
        labels = assign_clusters(data, centroids)
        centroids = update_centroids(data, labels, k)
    return labels, centroids
```

Updating Centroids Formula

The new centroid for each cluster is calculated as:

$$C_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i \tag{7}$$

Where C_j is the centroid of cluster j, and S_j is the set of points in cluster j.

Conclusion and Key Takeaways - Importance of Unsupervised Learning

- **Definition**: Unsupervised learning is a type of machine learning where the model learns from unlabeled data, identifying patterns without explicit instructions.
- **Purpose**: Aims to explore data's underlying structure to extract meaningful insights, making it powerful for exploratory data analysis.

Conclusion and Key Takeaways - Significance of Clustering Techniques

- **Clustering**: A key unsupervised learning method that groups similar data points into clusters, aiding in pattern identification.
- **Common Algorithms**:
  - **K-Means**: Partitions data into K distinct clusters based on feature similarity.
  - **Hierarchical Clustering**: Creates a tree-like structure by building a hierarchy of clusters.
  - **DBSCAN**: Groups points closely packed together and marks outliers in low-density regions.

Conclusion and Key Takeaways - Applications and Final Thoughts

- **Real-World Applications**:
  - **Market Segmentation**: Tailors marketing strategies by identifying customer groups.
  - **Social Network Analysis**: Identifies communities within networks, enhancing understanding of connectivity.
  - **Image Processing**: Organizes image pixels into segments for simplified representation and object recognition.
- **Key Takeaways**:
  - Unsupervised learning reveals hidden structures, leading to deeper insights.
  - Versatile across various domains: marketing, healthcare, and image analysis.
  - Clustering serves as a preliminary step for supervised learning, aiding in feature engineering.

Conclusion - Final Remarks

- Unsupervised learning and clustering techniques are vital in machine learning, allowing for exploration of data beyond predefined categories.
- Understanding these concepts empowers practitioners to effectively leverage modern datasets, fostering innovation and informed decision-making across various industries.

Python Example of K-Means

```
from sklearn.cluster import KMeans
import numpy as np

# Sample data
data = np.array([[1, 2], [1, 4], [1, 0],
                 [4, 2], [4, 4], [4, 0]])

# Apply K-Means
kmeans = KMeans(n_clusters=2, random_state=0).fit(data
    )

# Predicted cluster for each point
print(kmeans.labels_)
```