Shahid Beheshti University
Machine Learning
M.Sc - Fall 2024

Assignment 4

Theoretical

1. What is Clustering?

Clustering is a type of unsupervised learning where the goal is to group similar data points together. Unlike supervised learning, which uses labeled data, clustering algorithms aim to find inherent patterns in the data without pre-existing labels. Explain the differences between supervised and unsupervised learning, and describe why clustering is categorized under unsupervised learning.

2. What is the Role of Distance Metrics in Clustering?

Clustering algorithms rely on distance metrics to determine how similar or dissimilar data points are to one another. Explain the concept of a distance metric and its importance in clustering algorithms. What are the most common distance metrics used in clustering? Provide examples of how the Euclidean distance, Manhattan distance, and cosine similarity are used.

3. How Does K-means Clustering Work?

The K-means clustering algorithm is an iterative method that divides a dataset into a predetermined number of clusters (K). Describe the K-means algorithm in detail, including its initialization, iteration steps, and convergence criteria. What are the key advantages and limitations of K-means clustering?

How does K-means deal with the issue of selecting the optimal number of clusters? Discuss methods for determining the value of K, such as the Elbow Method or Silhouette Analysis.

K-means can suffer from initialization problems, where poor initial centroids lead to suboptimal clustering results. What are the common initialization challenges, and how does the K-Means++ algorithm address these issues? Compare K-Means++ to random initialization.

Finally, compare K-means with K-medoids in terms of their methodologies, robustness to outliers, and computational complexity.

4. How Does Hierarchical Clustering Work?

Hierarchical clustering builds a tree-like structure called a dendrogram that represents the data's hierarchical relationships. Explain how hierarchical clustering works, and differentiate between agglomerative and divisive hierarchical clustering. What are the key advantages and disadvantages of hierarchical clustering compared to K-means clustering?

5. How data distribution effects?

The role of data distribution is crucial in clustering algorithms, as certain methods rely on assumptions about the underlying structure of the data. Explain how different clustering algorithms account for or rely on data distribution. Specifically, describe the importance of distribution in Gaussian Mixture Models (GMMs) and the Expectation-Maximization (EM) algorithm.

How does the assumption of a Gaussian distribution influence the performance of these algorithms? Compare these to other clustering algorithms, such as K-Means, DBSCAN, and Spectral Clustering, in terms of their sensitivity to data distribution.

Finally, discuss the implications of incorrect distribution assumptions and the challenges of applying these algorithms to real-world datasets with non-Gaussian or irregular distributions. What techniques or preprocessing steps can be used to address these issues?

6. What is DBSCAN and How Does it Handle Outliers?

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that groups together points that are closely packed and marks points in low-density regions as outliers. Explain the working principle of DBSCAN and how it handles outliers differently from K-Means and Hierarchical Clustering. What are the key parameters of DBSCAN (eps and min$_samples),andhowdotheyaffecttheclusteringresults$?

Additionally, explain the working principles of OPTICS (Ordering Points To Identify the Clustering Structure) and HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise). How do these algorithms extend or improve upon DBSCAN, particularly in handling clusters with varying densities? Discuss their parameters, such as the reachability distance in OPTICS and the minimum cluster size in HDBSCAN, and their impact on clustering results and outlier detection.

7. How Can Clustering Results be Evaluated?

Once a clustering algorithm has been applied, it is important to evaluate the quality of the resulting clusters. Discuss different metrics used for evaluating clustering performance, such as the silhouette score, Davies-Bouldin index, and within-cluster sum of squares (WCSS). How do these metrics help in comparing different clustering algorithms and determining the best-performing model?

Provide a detailed explanation of the silhouette score, including how it is calculated for individual data points and how it aggregates into an overall score for the clustering result. What does a high or low silhouette score indicate about the quality of the clusters?

How does the silhouette score help in evaluating the separation between clusters and the cohesion within clusters? Compare its use to other metrics like the Davies-Bouldin index. Finally, discuss the limitations of the silhouette score and scenarios where it may not be the most suitable metric for clustering evaluation.

8. What is the Role of Dimensionality Reduction in Clustering?

Dimensionality reduction techniques, such as PCA (Principal Component Analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding), are commonly used before applying clustering algorithms. Explain the importance of dimensionality reduction in clustering tasks and how it can improve clustering performance. How do PCA and t-SNE differ, and when would you choose one over the other?

Discuss the relationship between PCA and Singular Value Decomposition (SVD). How does SVD underpin the mathematical foundation of PCA? What role do eigenvalues and eigenvectors play in this context?

Additionally, describe how PCA and t-SNE handle data distributions differently. Why is PCA more suited for preserving global structures, while t-SNE excels at capturing local patterns? Discuss the implications of these differences for clustering tasks, particularly when the dataset has a complex or non-linear distribution.

At last, highlight the strengths and limitations of each method, and provide examples of scenarios where PCA or t-SNE might be more appropriate for dimensionality reduction before clustering.

9. Defining different loss functions in imbalance data

In machine learning, selecting an appropriate loss function is crucial for achieving optimal performance, particularly when dealing with imbalanced datasets. Discuss the role of loss functions in model training and how they impact the learning process. What challenges arise when the data is imbalanced, and why do standard loss functions, such as cross-entropy, often fail in such scenarios?

Explain how loss functions can be adapted to address data imbalance, including approaches such as weighted loss, focal loss, and class-balanced loss. When and how should you decide to change the loss function based on the nature of the imbalance? Provide examples of practical situations where each of these loss functions would be appropriate.

10. Principal Component Analysis (PCA)

Given the following data matrix X with 4 observations and 3 features:

$$X = \begin{bmatrix} 2 & 4 & 6 \\ 4 & 6 & 8 \\ 6 & 8 & 10 \\ 8 & 10 & 12 \end{bmatrix}$$

Tasks:

1. Perform Principal Component Analysis (PCA) on the given data matrix X.
2. Reduce the dimensionality of the dataset from 3 features to 2 features.
3. Provide the final transformed data in the new 2-dimensional space.

Note: Show all your calculations and clearly present the final results.

Practical: Clustering the ECG Heartbeat Categorization Dataset

Dataset

The ECG Heartbeat Categorization Dataset consists of labeled ECG heartbeat data from 12 different classes of heartbeats. The dataset contains various features derived from the raw ECG signals. These features can be used for clustering purposes, even though the data is labeled. The challenge is to explore different clustering methods on this dataset and compare their effectiveness.

- **Link to the dataset:** `https://www.kaggle.com/datasets/shayanfazeli/heartbeat`

Tasks

1. Data Preprocessing

Start by loading the ECG Heartbeat dataset and examining its structure. Perform the necessary preprocessing steps such as handling missing values, normalizing or standardizing the data, and encoding categorical variables if required.

Check the dataset for class imbalance and decide how to handle it, such as through oversampling, undersampling, or other techniques.

2. K-means Clustering

Apply the K-means clustering algorithm to the dataset. Use the Elbow method to determine the optimal number of clusters. Visualize the clusters using scatter plots or other appropriate techniques and interpret the results.

Compare your findings with the ground truth labels, if applicable, to understand the algorithm's clustering behavior.

3. Hierarchical Clustering

Implement hierarchical clustering (both agglomerative and divisive if possible) and visualize the resulting dendrogram. Use the dendrogram to select an appropriate number of clusters.

Compare the results with those obtained using K-means clustering. Discuss the differences in the cluster distribution and interpretability.

4. DBSCAN

Apply the DBSCAN algorithm to the dataset and experiment with different values for the parameters `eps` and `min_samples`. Identify the number of clusters and outliers detected by DBSCAN.

Discuss how DBSCAN handles noise and outliers differently from K-means and hierarchical clustering.

5. PCA for Dimensionality Reduction

Apply PCA to reduce the dimensionality of the dataset. After applying PCA, use the reduced data to perform clustering with K-means, hierarchical clustering, and DBSCAN.

Compare the clustering results with and without PCA. Analyze whether dimensionality reduction improves the performance or interpretability of the clustering methods.

6. Cluster Evaluation and Comparison

Use various clustering evaluation metrics, such as the silhouette score, Davies-Bouldin index, and within-cluster sum of squares, to assess the results of all the clustering algorithms.

Compare the performance of K-means, hierarchical clustering, and DBSCAN using these evaluation metrics. Discuss which algorithm performed best on this dataset and why.

Deliverables

- **Code Implementation:** Provide your implementation of the clustering algorithms in Python. Use libraries like `sklearn`, `scipy`, and `matplotlib` for the clustering and visualization tasks.
- **Report:** Write a detailed report that includes:
  - A description of the dataset and preprocessing steps.
  - Explanation of each clustering method implemented.
  - Visualizations of the clustering results (e.g., scatter plots, dendrograms).
  - Evaluation of the performance of each algorithm.
  - A comparison of the clustering methods based on the results and evaluation metrics.
  - Insights and conclusions from the analysis.