INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of Computer Science and Engineering

CS60050 – Machine Learning

Project - 3

Airline Passenger Segmentation using Single Linkage Agglomerative (Bottom-Up) Clustering Technique

Submitted By:

Shailesh Chaudhary - 22CS60R37

Introduction

- Clustering is an unsupervised learning technique that is used to group similar data points together. It is a popular technique in the field of data science, machine learning, and data mining. Clustering algorithms are broadly categorized into two categories: K-Means clustering and Hierarchical clustering.

- K-means clustering is a popular and widely used clustering algorithm. It is a centroid-based algorithm that groups the data points into k clusters based on their similarity. The algorithm starts by selecting k random centroids and then iteratively assigns the data points to their closest centroid. The centroid is updated at each iteration based on the new cluster assignments, and the process continues until convergence. The K-means algorithm is computationally efficient and works well for large datasets with a moderate number of clusters.

- Hierarchical clustering is another widely used clustering algorithm that creates a hierarchy of clusters. It is of two types: Agglomerative and Divisive. Agglomerative clustering starts with each data point as a separate cluster and then merges the most similar clusters iteratively until all the points belong to a single cluster. The divisive clustering, on the other hand, starts with all the data points as a single cluster and then divides it into smaller clusters recursively until each point belongs to its own cluster. Hierarchical clustering is computationally expensive and works well for smaller datasets with a hierarchical structure.

Implementation

K-mean clustering:

- In this study, we implemented k-means clustering on a given dataset. The aim of clustering is to group similar data points together into distinct clusters based on a similarity metric. In this implementation, we considered k=3 clusters and used cosine similarity as the distance measure.

- To begin with, we randomly initialized k cluster means as k distinct data points. The algorithm then iterated for 20 iterations. During each iteration, the algorithm assigned each data point to the nearest cluster means based on cosine similarity. Next, the algorithm recalculated the cluster means based on the mean of all the data points assigned to that cluster. This process was repeated for 20 iterations.

- After the iterations were completed, the clustering information was saved in a file. The file contained information on which cluster each data point belonged to, along with the cluster mean for each cluster. This information can be used for further analysis or visualization of the clustering results. Overall, k-means clustering proved to be an effective technique for grouping similar data points together into distinct clusters based on cosine similarity.

Result of K-Mean Clustering:

- **Optimal Value of K** : After running the k-mean clustering code we got the best value for k is 3 among k=3,4,5,6
- **Maximum Silhouette Coefficient :** On k=3 we got the maximum Silhouette Coefficient = 0.8586

```
Silhouette Coefficient for 3 clusters: 0.8586404878582597
Silhouette Coefficient for 4 clusters: 0.8563748841432676
Silhouette Coefficient for 5 clusters: 0.8039836048397391
Silhouette Coefficient for 6 clusters: 0.7645816430897951

Maximum Silhouette Coefficient of 0.8586404878582597 is achieved for 3 clusters.
```

Implementation of Single Linkage Agglomerative (Bottom-Up) Clustering algorithm:

Below is the explanation of the implementation of the Single Linkage Agglomerative (Bottom-Up) Clustering algorithm

The class AgglomerativeHierarchicalClustering takes in the data to be clustered, the number of desired clusters, and the distance measure to be used. In this implementation, cosine similarity is used as the distance measure.

The init_clusters method initializes the clusters and indices of the data points. Initially, each data point is considered as its own cluster. The find_closest_clusters method finds the two closest clusters based on the given distance measure. The merge_and_form_new_clusters method merges the two closest clusters and forms new clusters.

The run_algorithm method runs the clustering algorithm until the desired number of clusters is achieved. It merges the two closest clusters at each iteration and updates the clusters and indices. Finally, the getCentroids method returns the indices of the data points that represent the centroids of the clusters.

Overall, this implementation of single linkage agglomerative clustering algorithm is a simple and efficient way to cluster data based on similarity.

Jaccard Similarity:

Jaccard similarity is a measure of similarity between two sets of data. It is defined as the size of the intersection divided by the size of the union of the sets. The Jaccard similarity coefficient ranges between 0 and 1, where 0 indicates no similarity and 1 indicates complete similarity between the sets. Jaccard similarity is commonly used in clustering and classification problems to compare the similarity between different groups of data.

Below is the explanation of is the implementation of Jaccard Similarity**:**

The code reads the cluster information from two separate text files 'kmeans.txt' and 'agglomerative.txt' and computes the Jaccard similarity between corresponding sets of clusters. The Jaccard similarity is a measure of similarity between two sets, and is calculated as the ratio of the size of the intersection of the two sets to the size of their union.

The code first reads the cluster information from each file and stores it in a list of sets, where each set contains the data points belonging to a particular cluster. Then, it computes the Jaccard similarity between each pair of clusters from the two lists using the 'jaccard_similarity' function. The resulting Jaccard similarity matrix is stored in a two-dimensional list 'jaccard_matrix'. Finally, the code prints the Jaccard similarity matrix to the console.

Result of Jaccard Similarity:

Below is the similarity matrix for cluster generated using k-mean clustering and Single Linkage Agglomerative (Bottom-Up) Clustering

```
[0.8995132127955494, 0.04429482636428065, 0.012168933428775949]
[0.033655567490153956, 0.2576271186440678, 0.0]
[0.013019891500904159, 0.0, 0.4067796610169492]
```

Time taken to run the program:

K-mean clustering: It take approximately one minute to execute.

Single Linkage Agglomerative (Bottom-Up) Clustering : For 1000 datapoints it take approximately two hours and for our dataset of 3000 datapoints its going to take approximately 4 to 5 hours.

CONCLUSION

Based on the analysis performed on the Indian airline's customer dataset, we have clustered the data using both k-means clustering and hierarchical clustering algorithms. We found that the optimal number of clusters is 3, as it gave us the highest Silhouette Coefficient value of about 0.85.

We used cosine similarity as the distance measure for both algorithms, and single linkage agglomerative (bottom-up) clustering with single linkage strategy was used to divide the data into 3 sets of data points.

To evaluate the effectiveness of the clustering algorithms, we used the Silhouette

Coefficient metric, which gave us a score of 0.85 for k=3. This indicates that the clusters are well-separated and dense.

Finally, we computed the Jaccard similarity between the corresponding sets of clusters obtained from the k-means and hierarchical clustering algorithms. The Jaccard similarity score was 0.23, which is a relatively good value, indicating that the two clustering algorithms produced similar results.

In conclusion, our analysis shows that the Indian airline's customer dataset can be clustered into 3 optimal clusters using both k-means and hierarchical clustering algorithms, with a high degree of similarity between the two clustering results.