Indian Institute of Technology Kharagpur
Machine Learning (CS60050)
Spring 2022-23
Project 3: Seed Type Determination Rating using Single Linkage Agglomerative (Bottom-Up) Clustering Technique (STHC-AS)

Avi Amalanshu (20EC30063)
Guided by Prof. Aritra Hazra and TA Abhinav Bohra

April 15, 2023

1 Abstract

This report presents a project on clustering a given dataset into an optimal number of clusters using k-means and hierarchical clustering algorithms. The dataset consists of seed information, and the goal is to identify similar groups of data points based on cosine similarity as the distance measure. K-means clustering was performed with k=3 clusters, and the algorithm was iterated for up to 20 iterations. However, it was found that even with up to 40 iterations, the optimal value of k varied widely, indicating the sensitivity of the results to initialization. Evaluation of the clustering algorithm was done using the Silhouette coefficient, which measures the quality of clustering based on the mean distance between points within clusters and between clusters. The optimal value of k was determined to be 3 based on the highest Silhouette coefficient.

Further analysis was done by performing hierarchical clustering using a single linkage agglomerative (bottom-up) approach with the same notion of similarity as in the k-means algorithm. However, it was observed that single linkage clustering tended to create singleton clusters with outliers, resulting in poor Jaccard similarity coefficients between corresponding sets of clusters from k-means and hierarchical clustering. The Jaccard similarity coefficients were found to be approximately 0.9, 0.5, and 0.05, indicating that the clustering results were not consistent across the two algorithms. These findings suggest that the choice of clustering algorithm and the value of k can significantly impact the clustering results and should be carefully considered in similar analyses.

2 K-means clustering

K-means clustering is a popular unsupervised machine learning algorithm that aims to partition a given dataset into k distinct clusters based on similarity measures. The algorithm begins by randomly initializing k cluster centroids, which are then iteratively updated to minimize the sum of squared distances between data points and their respective centroid. This process continues until convergence is reached or a maximum number of iterations is reached.

Algorithm 1 K-means Clustering

Input: Dataset $X = \{x_1, x_2, \ldots, x_n\}$, Number of clusters k
Output: Cluster assignments $\{C_1, C_2, \ldots, C_k\}$, Centroid of each cluster $\{c_1, c_2, \ldots, c_k\}$
Initialize centroids: $c_1, c_2, \ldots, c_k \leftarrow$ randomly select k data points from X
while not converged **do**
    for $i = 1$ to n **do**
        Assign x_i to the nearest centroid:

$$C_i \leftarrow \arg\min_j \text{dist}(x_i, c_j)$$

    for $j = 1$ to k **do**
        Update centroids:

$$c_j \leftarrow \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Return $\{C_1, C_2, \ldots, C_k\}$, $\{c_1, c_2, \ldots, c_k\}$

3 Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering algorithm that iteratively merges clusters of data points based on a similarity measure until a stopping criterion is met. It can be categorized into two main approaches: top-down and bottom-up. This approach results in a hierarchical tree-like structure, also known as a dendrogram, which represents the sequence of cluster mergers (or splitting).

3.1 Top-Down

In top-down agglomerative clustering, also known as divisive clustering, the algorithm starts with all data points in a single cluster and recursively divides the clusters into smaller sub-clusters until each data point is in its own cluster. This approach requires selecting a dissimilarity measure and a stopping criterion to determine when to stop the recursive splitting of clusters.

3.2 Bottom-Up

In bottom-up agglomerative clustering, also known as agglomerative nesting or simply agglomerative clustering, the algorithm starts with each data point as an individual cluster and iteratively merges pairs of clusters based on a similarity measure (here: cosine similarity $\frac{<a,b>}{\sqrt{<a,a>}\sqrt{<b,b>}}$).

There are several methods for defining the similarity between clusters in bottom-up agglomerative clustering. Some commonly used linkage methods include:

3.2.1 Complete Linkage

The similarity between two clusters is defined as the maximum dissimilarity between any pair of data points from the two clusters. This method tends to produce compact, well-separated clusters.

3.2.2 Centroid Linkage

The similarity between two clusters is defined as the dissimilarity between their centroids, which are the mean vectors of the data points in each cluster. This method tends to produce clusters with similar sizes and shapes.

3.2.3 Single Linkage

The similarity between two clusters is defined as the minimum dissimilarity between any pair of data points from the two clusters. This method tends to produce elongated clusters and is sensitive to noise and outliers. However, it may be more efficient to calculate since it lends to greedy methods: an interesting way to represent single-linkage clustering is as a minimum spanning tree (MST) with the most expensive k-1 edges removed. In this representation, each data point is initially treated as a separate cluster, and the algorithm iteratively builds a tree connecting the clusters by selecting the edge with the smallest dissimilarity between clusters. This process continues until all data points are in a single cluster. Once the MST is constructed, the k-1 edges with the highest dissimilarity are removed, resulting in k clusters with the data points in each cluster being connected by edges with the lowest dissimilarity.

4 Methodology

We wrote Python code that

- Reads the csv file into a Pandas dataframe
- Performs k-means clustering for $k = 3, 4, 5, 6$ and calculates the silhouette coefficient s for each clustering.
- Selects k with the largest s and performs single-linkage clustering with both brute force and greedy algorithms

- Compares Jaccard coefficients between the obtained clusterings

5 Results

5.1 Silhouette Coefficients and values of k

```
k = 3, s = 0.5824521339040021
k = 4, s = 0.5230638508698913
k = 5, s = 0.563970047251916
k = 6, s = 0.5421660289617972
Greatest value of s = 0.5824521339040021 for k = 3
```

5.2 Pair-wise Jaccard Coefficients of the three models

```
Jaccard scores for set-wise closest clusters in k-means and brute force:
[0.5323741007194245, 0.015151515151515152, 0.9154929577464789]
Jaccard scores for set-wise closest clusters in k-means and greedy:
[0.5323741007194245, 0.015151515151515152, 0.9154929577464789]
Jaccard scores for set-wise closest clusters in brute force and greedy:
[1.0, 1.0, 1.0]
```

6 Key Observations

- The silhouette coefficient is roughly $s = 0.5824521339040021$ for $k = 3$
- The highest silhouette coefficient is associated with $k = 3$. Over 10 runs of the experiment, we got on average $k = 4, s = 0.5230638508698913$, $k = 5, s = 0.5196839778631188$, $k = 6, s = 0.40442882908080907$
- On some runs of the experiment, we got different best-k values. The final clustering is heavily dependent on the random initialization. Across uniform initializations, s has high variance $\forall\ k$. This is somewhat stabilized by choosing a larger number of iterations, however, not entirely so. On one run with $n = 400$ iterations, we got
- The Jaccard coefficients for single-linkage agglomerative clustering by Kruskal's algorithm and by brute force are 1. This indicates the two algorithms generate identical clusterings.
- The Jaccard coefficients for single-linkage agglomerative clustering and k-means clustering are around [0.915, 0.532, 0.015]. Upon further investigation, we see that the data with index 39 always gets placed in its own singleton set. Since the single-linkage strategy depends only on *the* very shortest possible link between two clusters, it is very sensitive to outliers, here despite the existence of the target label as a heuristic. One of the clusters is roughly identical to one of the k-means ones, but the rest of

the data is mostly allotted to one cluster. Since the target groups are roughly equally sized, this leads to a Jaccard coefficient of $\approx \frac{1}{2}$ for the set which corresponds (nearly) to the union of the sets which do not form the high-scoring set.