Report

Project Code: SRHC-AS

Project Title: Sales Grouping by Representatives using Single Linkage Agglomerative (Bottom-Up) Clustering Technique

Cosine Similarity

The Cosine similarity is a way to measure the similarity between two non-zero vectors with n variables. If the cosine value of two vectors is close to 1, then it indicates that they are almost similar. A zero value indicates that they are dissimilar.

$$similarity = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

In the code for cosine similarity, the **dot** function is used to calculate the dot product between the data points and the cluster means. The **norm** function is used to calculate the L2 norm of the data points and the cluster means. The **argmax** function is used to assign each data point to the closest cluster based on the cosine similarity. The **mean** function is used to calculate the new cluster means.

Note that this implementation has a time complexity of O(n^4), which makes it inefficient for large datasets. However, it should work well for small to medium-sized datasets.

Silhouette Coefficient

The Silhouette coefficient is a metric used to evaluate the performance of clustering algorithms like K-means. It provides a measure of how well each data point fits into its assigned cluster and how well-separated the clusters are from each other.

$$\text{Silhouette Coefficient}(s) = \frac{b - a}{\max(a, b)}$$

The Silhouette Coefficient is defined for each sample and is composed of two scores:

a. The mean distance between a sample and all other points in the same cluster.
b. The mean distance between a sample and all other points in the next nearest cluster.

The silhouette coefficient ranges from -1 to 1, where a value of 1 indicates that the point is well-clustered and a value of -1 indicates that the point is misclassified. higher Silhouette coefficient generally indicates better clustering performance, while a negative Silhouette coefficient indicates poor clustering performance.

Jaccard Similarity

The Jaccard similarity measures the similarity between two sets of data to see which members are shared and distinct. The Jaccard Similarity will be 0 if the two sets don't share any values and 1 if the two sets are identical. The Jaccard similarity can be computed as the size of the intersection divided by the size of the union of two sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Single Linkage Agglomerative (Bottom-Up) Clustering

The basic steps of the single linkage agglomerative clustering algorithm:

1. Calculate the cosine similarity between each pair of data points
2. Each data point is initially assigned to its own cluster
3. Find the two clusters with the smallest distance (highest similarity) and merge them into a new cluster
4. Recalculate the distances between the new cluster and all other clusters
5. Repeat steps 3 and 4 until all data points are in a single cluster

K-means clustering

The K-means clustering algorithm is implemented using the functions **cosine_similarity**, **intra_cluster_distance**, **nearest_cluster_distance**, and **kmeans_clustering**. The function **cosine_similarity** computes the cosine similarity between two vectors, while **intra_cluster_distance** computes the mean distance between a sample and all other points in the same cluster. **nearest_cluster_distance** computes the mean distance between a sample and all other points in the next nearest cluster. **kmeans_clustering** is the main function that performs K-means clustering on the data. It initializes the cluster means as k distinct data points, assigns each data point to the nearest cluster, updates the cluster means, assigns cluster labels to data points, and computes the Silhouette coefficient.

Point-wise summary of the code:

1. The necessary libraries (numpy and pandas) are imported.
2. The sales data is loaded from a CSV file using pandas.
3. The number of samples is reduced to 1000 by randomly selecting 1000 samples from the dataset for fast calculation.
4. The "Record" column is dropped from the dataset as it is unnecessary because it has only a “1” value in all rows.

5. The "Deal ID" column is a mixture of three features (Category+Year+ID). I have already Year column, so I extracted two features: "Category" and "ID", and the old "Deal ID" column is dropped from the dataset.
6. The categorical parameters (Country, Category, and Sales Rep) are encoded using one-hot encoding.
7. The dataset is converted to a numpy array for ease of use.
8. The cosine similarity function is defined to compute the cosine similarity between two vectors.
9. The intra_cluster_distance function is defined to compute the mean distance between a sample and all other points in the same cluster.
10. The nearest_cluster_distance function is defined to compute the mean distance between a sample and all other points in the next nearest cluster.
11. The k-means clustering algorithm is defined using the above-defined functions.
12. The save_cluster function is defined to save the clustering information to a file.
13. The optimal value of k is found using the Silhouette coefficient.
14. The k-means clustering algorithm is run for each value of k, and the clustering information for the best value of k is saved to a file.
15. The clusters information for each value of k is printed.
16. The Silhouette coefficient for each value of k is printed.
17. Single Linkage Agglomerative (Bottom-Up) Clustering function is defined using above-defined steps.
18. Map each set of case A (kmeans) to a distinct set of case B (agglomerative) (one-to-one and onto mapping) considering the Jaccard similarity. Print the Jaccard Similarity scores for all the k mappings.
19. **"kmeans.txt"** and **"agglomerative.txt"** files will have been generated during program execution. These files contain cluster information based on **K-means** and **Single Linkage Agglomerative (Bottom-Up)** methods respectively. I have made the clusters based on the **row number** of original dataset. So, in these files, each cluster contains a group of row numbers in sorted order.

Results

```
Clusters for k = 3:
Cluster 1 have 191 data points
Cluster 2 have 221 data points
Cluster 3 have 588 data points
Silhouette coefficient for k = 3: -0.06601669572319607

Clusters for k = 4:
Cluster 1 have 68 data points
```

```
Cluster 2 have 221 data points
Cluster 3 have 555 data points
Cluster 4 have 156 data points
Silhouette coefficient for k = 4: -0.12573718424975047

Clusters for k = 5:
Cluster 1 have 74 data points
Cluster 2 have 221 data points
Cluster 3 have 47 data points
Cluster 4 have 113 data points
Cluster 5 have 545 data points
Silhouette coefficient for k = 5: -0.14099967295886978

Clusters for k = 6:
Cluster 1 have 540 data points
Cluster 2 have 39 data points
Cluster 3 have 221 data points
Cluster 4 have 64 data points
Cluster 5 have 103 data points
Cluster 6 have 33 data points
Silhouette coefficient for k = 6: -0.16170881816277882

Optimal value of k: 3
```

	Country	Sales Rep	Sales	Year of Transaction Date	Category	ID	KmeansCluster	AggCluster
136044	Spain	Ralph Ritter	530.868	2013	ES	2239615	1	0
147440	United Kingdom	Frank Atkinson	295.452	2013	MX	101525	2	0
230105	Sudan	Shirley Schmidt	39.480	2011	CA	124464	2	0
170724	France	Elizabeth Moffitt	1041.336	2012	ID	29424	2	0
260142	United Kingdom	Patrick O'Donnell	195.440	2011	MX	106390	2	0
3062	United States	Rob Dowd	5.220	2014	CA	131653	2	0
226283	Syria	Chuck Sachs	167.724	2011	US	137309	2	0
250959	Spain	Gary Mitchum	758.352	2011	CA	138940	2	0
216035	Senegal	Erica Hernandez	633.480	2012	SG	2990	0	0
247379	Philippines	Neil Französisch	173.610	2011	ES	3897276	1	0

Jaccard Similarity for all k mappings

```
[[0.189      0.0052356  0.0052356 ]
 [0.22144289 0.         0.        ]
 [0.58917836 0.         0.        ]]
```

Note: I have executed on 1000 samples. Pandas sample function randomly selects rows. On every execution randomly 1000 rows will select. So, every time you will receive different outputs.