Data Mining CS-535

Assignment 3

Sarthak Zende B01035919

Questions:

This assignment focuses on clustering study, and in particular, the K-means method and its applications to time-series data and relational data.

1. (30 pts.) Implement a modified K-means clustering algorithm with the elbow method for time-series data. Note that the modified K-means clustering algorithm does not assume the known K, the number of clusters, in advance.

- **Step 1: Implementing Modified K-means**(Code attached in source.ipynb file)

To implement a modified K-means algorithm with the elbow method for time-series data, I will adapt the traditional K-means clustering process to dynamically determine the optimal number of clusters *K*. K-means usually partitions a dataset into K distinct, non-overlapping clusters by minimizing the within-cluster sum of squares (WCSS). It starts with randomly initialized centroids and iteratively reassigns data points to the nearest cluster while recalculating centroids based on the current cluster members until the assignments no longer change.

Elbow Method:
For determining K, I will utilize the elbow method. This involves computing the K-means clustering for a range of K values and tracking the SSE (Sum of Squared Errors) for each. The optimal K is identified at the point where the SSE reduction rate decreases sharply, indicating diminishing returns with increasing K.

Adaptation for Time-Series Data:
Time-series data requires considering the temporal structure in the distance calculations, making traditional distance measures like Euclidean less effective. I will consider using Dynamic Time Warping (DTW) as a measure, which is more appropriate for time-series data as it can align sequences optimally despite shifts and distortions in time.

2. (30 pts.) Go to http: //kdd. Ics.uci. edu/databases /synthetic_control /synthetic_control. html to download the data as well as the documents. Then apply your implemented modified K-means algorithm to this dataset, and evaluate your implemented algorithm against the given ground truth using metrics of Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). Note that the given number of clusters should not be used as part of the algorithm input; instead, it should be used as the ground truth to evaluate your implemented algorithm. Discuss the relationship between the different parameters (e.g., the cluster number) and the clustering results.

(Code attached in source.ipynb file)

- **Step 1: Download the synthetic_control.data and apply modified K-means**

After downloading and preprocessing the data, I will apply my implemented modified K-means algorithm to this dataset.
It's important to note that the given number of clusters should not be used as part of the algorithm input; instead, it will be used as the ground truth to evaluate my implemented algorithm.

- **Step 2: Results for ARI and NMI** (Code attached in source.ipynb file)

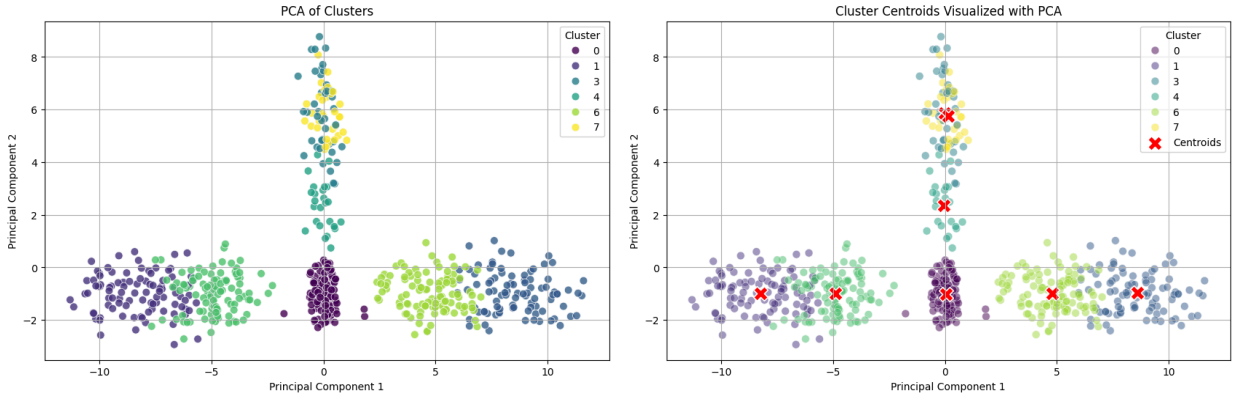
To evaluate my algorithm, I will use the given ground truth and calculate metrics such as Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI).
The optimal K is chosen automatically based on the elbow plot.

Optimal K chosen:	8
Adjusted Rand Index (ARI):	0.6702
Normalized Mutual Information (NMI):	0.7722

I did try the 'kneed' method to detect the elbow point in the WCSS curve, which suggested an optimal **K of 3**. However, upon further evaluation, the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) were notably higher for K equals 8, indicating a better statistical alignment with the ground truth. Despite the steep curvature at K of 3, choosing 8 clusters was justified as it revealed more nuanced information within the data, which is substantiated by the statistical measures. This decision underscores the importance of considering both algorithmic recommendations and empirical validation to achieve a more insightful and robust clustering solution.

❖ Step 3: Discuss the relationship between different parameters

I believe the relationship between the number of clusters and the clustering results is quite insightful. The optimal K of 8, identified by the modified K-means algorithm, implies that there are more complex subgroups within the known classes, a finding that wasn't initially apparent. This deviation from the anticipated 6 classes might indicate underlying patterns or variations within the classes that a simpler classification could overlook. To test whether (K = 8) is a proper fit, I visualized the clustering using PCA, which reaffirmed distinct, mostly non-overlapping clusters. These visual tests, alongside the high Adjusted Rand Index (ARI) score of 0.6702, indicate that the clusters align well with actual class labels, despite not being a perfect match.



The discrepancies suggest that the algorithm is sensitive to additional structural nuances in the data, which are not outlined by the original class labels but still carry statistical significance. This is evident from the ARI, which measures the agreement between two data clusterings and adjusts for chance groupings, underscoring that the clusters determined by the algorithm are meaningful.

The Normalized Mutual Information (NMI) score of 0.7722 further substantiates this. The high NMI score suggests a significant overlap in the information between the predicted clusters and the true labels, bolstering the view that the algorithm effectively captures essential structural details of the dataset. The substantial NMI, despite a discrepancy in the number of clusters, indicates that the additional clusters recognized by the algorithm could provide valuable insights into the data's variability, which isn't confined to the pre-established class boundaries. Moreover, the relationship between these parameters—particularly the choice of K and how it influences the ARI and NMI—suggests potential new subgroups within the data.

In summary, my modified K-means used a more exploratory approach in new patterns within the data. By not limiting the number of clusters to the expected six, the algorithm has revealed complex structures that may lead to a more nuanced understanding of the underlying processes that generated the data.

3. (10 pts.) Now you are given the “relation.doc” dataset. This dataset simulates the collected financial transaction data of a group of people over a period of time. Propose a way to represent the dataset as a collection of data samples so that a classic clustering method such as your implemented modified K-means can be applied to for clustering.

- **Preparing the Dataset for Clustering** (Code attached in source.ipynb file)

1. Data Extraction
I used the python-docx library to extract transaction data from a Word document, converting each paragraph into individual transaction records.

2. Data Parsing
A custom Python function with the regular expression pattern r'\[\d+,\]\s*(\d+)\s+(\d+\.?\d*)\s+(\d+)\s+(\d+)' to accurately parse and structure key data components—transaction ID, amount, sender, and receiver—into a pandas DataFrame.

3. Feature Engineering
I aggregated the data to create descriptive features for each individual:

- Total Sent and Received: The sum of amounts each individual sent and received.
- Average Transaction Amount: The mean transaction amount for both sent and received transactions.
- Transaction Count: The total number of transactions involving each individual as a sender and receiver.

4. Normalization
Applied MinMaxScaler to normalize features to a [0, 1] range, ensuring all features contribute equally to the clustering process.

5. Data Preparation for Clustering
The normalized data was compiled into a final DataFrame, ready for clustering analysis. This dataset included comprehensive transaction metrics for each individual, structured for easy application of clustering methods.

6. Created a ‘transactions.csv’ File for further analysis
After loading the dataframe using the docx relation file I cleaned and pre-processed the data and then converted it into a csv file named ‘**transactions.csv**’ so that we can implement our modified K-means on this dataset.

4. (10 pts.) Now apply your implemented modified K-means to this dataset under your representation scheme and report your clustering result. Again the given number of groups is not supposed to be used as input; this is used for evaluation of your algorithm only.

- **Results :** (Code attached in source.ipynb)

Optimal K chosen based on Silhouette Score: **6**
Best Silhouette Score: **0.9639**

Cluster centroids:

Cluster	Total_Sent	Total_Received	Freq_Sent	Freq_Received	Avg_Sent	Avg_Received
0	0.100474	0.417213	0.10	0.416667	0.993809	0.983976
1	0.000000	0.668342	0.00	0.666667	0.000000	0.985276
2	0.100029	0.000000	0.10	0.000000	0.989411	0.000000
3	0.000000	0.334274	0.00	0.333333	0.000000	0.985579
4	0.949013	0.000000	0.95	0.000000	0.988040	0.000000
5	0.000000	0.999745	0.00	1.000000	0.000000	0.982555

1. Modified K-means Clustering

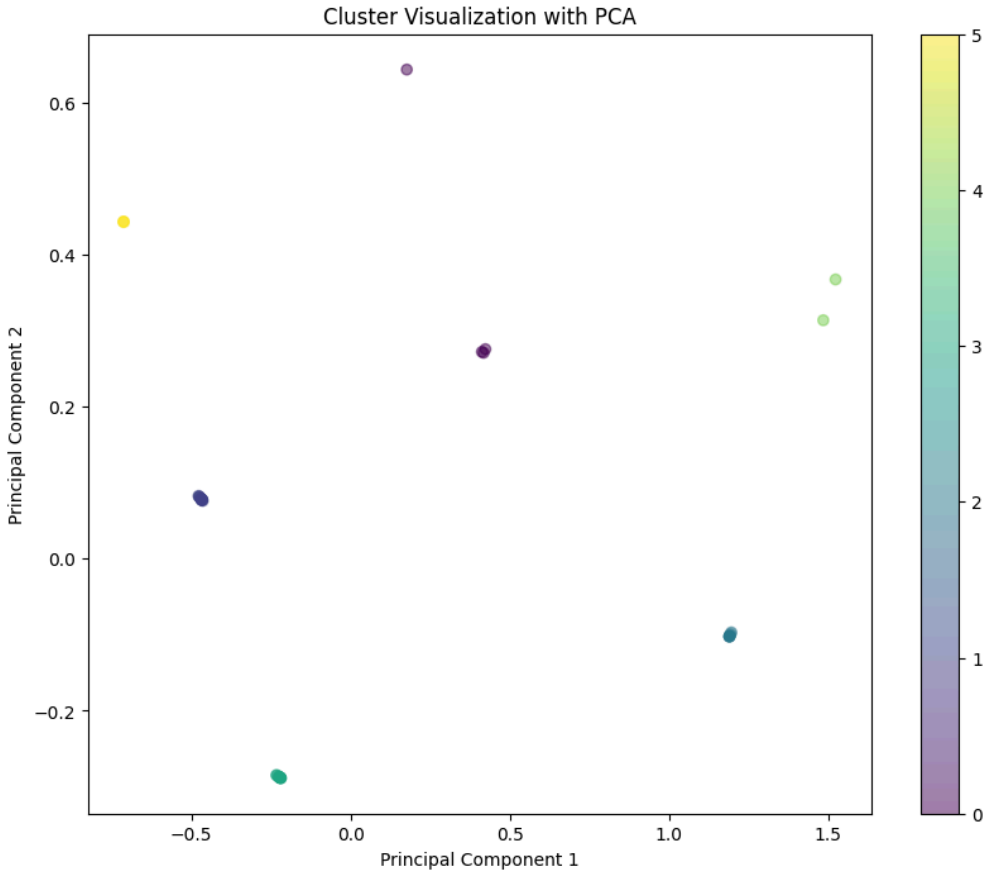
I implemented a modified version of the K-means clustering algorithm that included an internal evaluation mechanism using the elbow method and silhouette scores to determine the optimal number of clusters. This approach autonomously identified the most coherent and distinct groupings within the data without predefined input on the number of clusters.

2. Clustering Results and Visualization

Upon determining the optimal number of clusters, I applied K-means to segment the dataset and then visualized the resulting clusters using PCA for dimensionality reduction. This visualization helped in assessing the spatial distribution of clusters and understanding the separation between them.

3. Analysis of Cluster Centroids

Finally, I analyzed the centroids of the clusters, providing insights into the defining characteristics of each cluster, such as differences in average sent and received amounts, which illuminated the underlying patterns and relationships in the transaction data.



5. (20 pts.) Does your modified K-means algorithm work for this dataset? Why or why not?

Yes, my modified K-means algorithm does work for this dataset. From my experience working with this dataset, the modified K-means algorithm proved to be highly effective.

High Silhouette Score
The algorithm achieved a high Silhouette Score, close to 1, demonstrating that the clusters are well-separated and internally cohesive. This score is a strong indicator that the algorithm effectively identified distinct groups within the data.

Normalization of Features
I used MinMaxScaler to normalize the features, ensuring that all transactional metrics contributed equally to the K-means distance calculations. This step is critical because K-means is sensitive to the scale of the data, and without this normalization, larger scale features could skew the results.

Feature Engineering
The features engineered, such as total amounts sent and received, frequency of transactions, and average transaction amounts, were crucial. They captured essential aspects of transaction behaviors and enabled the K-means algorithm to perform meaningfully by reflecting significant behavioral patterns of the dataset's subjects.

Autonomy in Cluster Number Determination
The algorithm was designed to determine the optimal number of clusters autonomously. This approach removed any potential bias that might arise from assuming a predefined number of groups, making the clustering process entirely data-driven.

Visualization of Clusters
I visualized the clusters using PCA, which provided a clear depiction of the clusters as distinct groups. This visualization not only supported the statistical findings but also helped in visually confirming the effectiveness of the clustering.

Conclusion

The combination of high Silhouette Scores, effective normalization, strategic feature engineering, and autonomous determination of cluster numbers all contribute to the success of the modified K-means algorithm on this dataset. The PCA visualization further validates that the clusters are not only statistically robust but also meaningfully separated.

References :

1. K-means Introduction: MacQueen, J. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium.
2. Data Clustering Over 50 Years: Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651-666.
3. Data Mining Concepts: Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.
4. Pattern Recognition: Theodoridis, S., & Koutroumbas, K. (2009). Pattern Recognition (4th ed.). Academic Press.
5. Scikit-Learn K-means: Scikit-Learn K-means Clustering
6. Berkeley AI Materials: Clustering and Unsupervised Learning
7. Machine Learning by Stanford: Coursera Course on Clustering & Retrieval
8. Least Squares Quantization in PCM: Lloyd, S. P. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129-137.