CS 6140:Data Mining Project Intemediate-Report

Ghazal Abdollahi
Ajantha Varadharaaj
Abdias Baldiviezo

March 25, 2023

1 Analyzing Traffic Flow Patterns on AzureFunctionsDataset2019 through Clustering

For context refer to our (Proposal). and our (Data Collection Report).
Data can be found in our (Github Repo) which is a preprocessed version of (Microsoft's AzurePublicDataset).

2 Progress made towards our goal

Our progress can me summarized in the findings that resulted of our collection, pre-processing, processing and implementation of clustering algorithms to our "Dataset". This application has showed differences in our data points with which the data can be separated and clustered.

3 What worked well and what did not?

Our team decided to each take some clustering methods and test them with our data.

1. Successes. Some notable successes in the development of our project were: first, being able to see clearer separations between the clusters after dimensionality reduction and second, recolecting different aspects of the relationships between datapoints from the implementation and hyperparameter tunning in each of our 5 different clustering methods.

2. Challenges. Using all features that our dataset contains has yielded suboptimal results, with a very confusing separation of the data, perhaps because of the low correlations between certain features, to mitigate this aspect we used dimensionality reduction in 2 forms: Feature extraction and Feature selection. Another challenge we faced was long computational times due to the size of our data (approx. 619368) for this we were driven to use a combination of PCA and sub-sampling, which decreased our wait times and gave us more freedom in trying different experiments.

4 What could be done to improve the basic approaches?

Some improvements that were done/considered during development were:

- Using PCA (one or more rounds), tune "n" principal components and variance threshold.

Figure 1: Outlier and Mine Omission

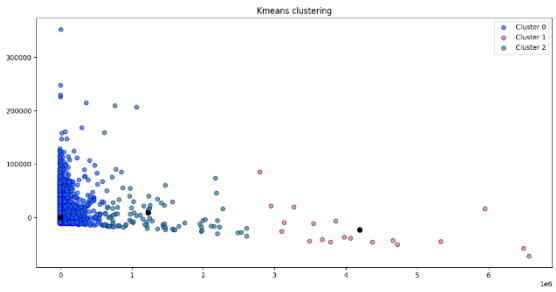


Figure 2: Normalized Data

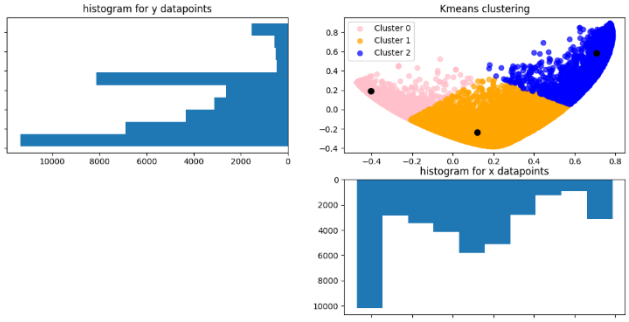


Figure 3: Normalized Data - Heatmap

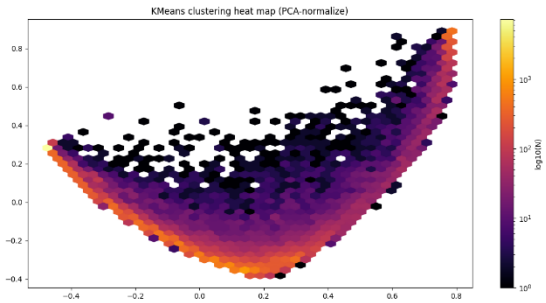
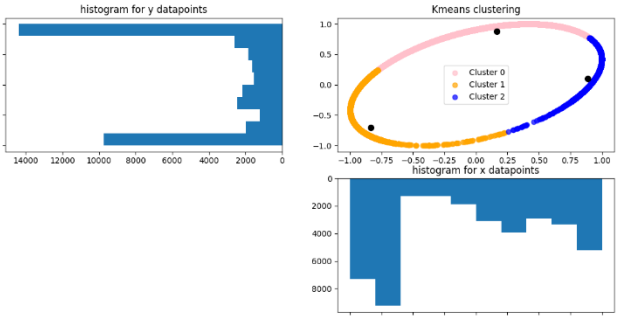


Figure 4: Cosine distance after 4 rounds of PCA



- Use HDBSCAN instead of DBSCAN which transforms space according to density/sparsity.
- Omitting the sparse points with a very significant Euclidean distance from all of the centroids.
- Ommiting mine values, whcih drag the centroids to themselves.
- Use libraries that make use of GPUs to achieve faster processing times (CuML).
- Using metrics to evaluate clustering performance, for example: Elbow method, ARI, and Silhouette Score (knee point).

5 Experiments

This is a compact summarization of many steps that went into these experiments.

1. **K-means, birch, and fuzzy** Preprocessing involved almost all of the items in the list above (Fig 1.) which drew the natural shape of the dataset and after normalization (Fig 2-3) and another round of PCA revealed a good separation between clusters. The cosine distance was also calculated between each pair (Fig 4).

Figure 5: Cumulative variance ratio versus the number of Components

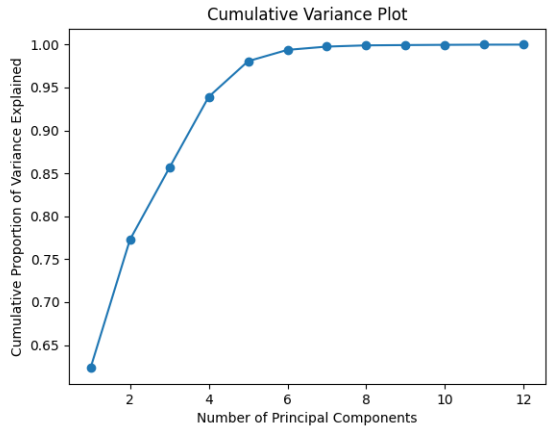


Figure 6: Finding Knee Point

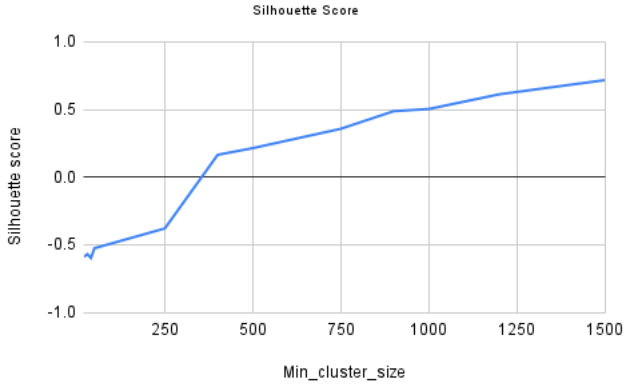


Figure 7: Approximate number of clusters using DBSCAN

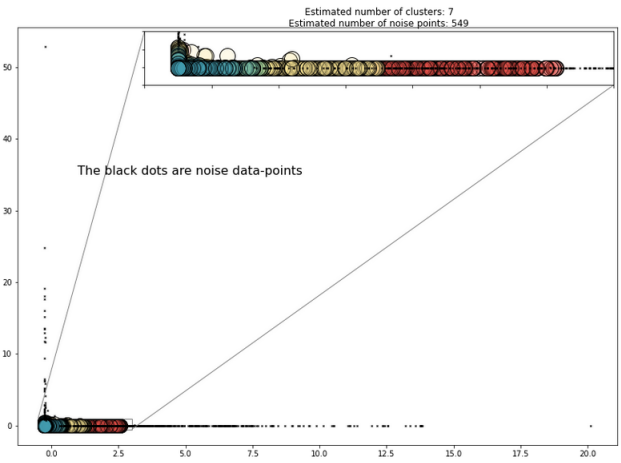
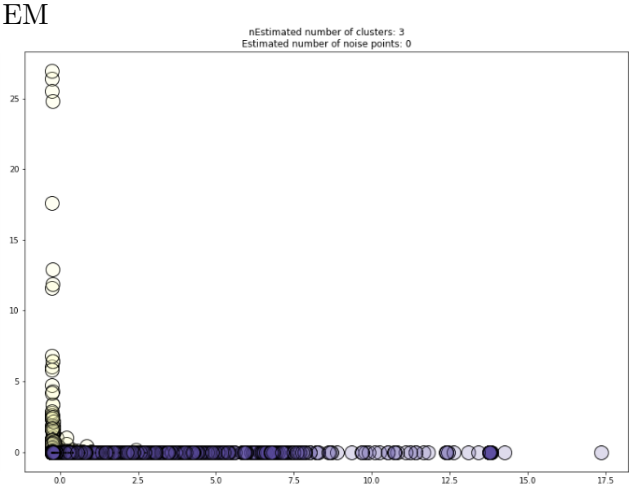


Figure 8: Approximate number of clusters using EM



2. **DBSCAN hyperparameters** Preprocessing to implement DBSCAN was crucial and relied on analysis (Fig 5) to find the correct hyperparameters for clustering e.g. Silhouette Score (Fig 6).

3. **DBSCAN subsampling** Feature selection was used for dimensionality reduction, as well as subsampling for reasonable computing times. The size of the samples was 20000 out of $\approx$ 650000 datapoints. The features used are "Average", "Count", "Trigger"(Fig 7). Important parameters were eps (¨..maximum distance between two samples for one to be considered as in the neighborhood of the other¨) and *min_samples* (¨The number of samples or total weight in a neighborhood for a point to be considered as a core point. This includes the point itself.¨), after trying different combinations of these and because of the size of the dataset, the following parameters were the ones that showed more promising results: $eps = 0.5$ and $min\_samples = 100$.

4. **Expectation maximization clustering (Gaussian Mixture)** The covariance matrix and the number of components (The number of mixture components.) were the most important parameters in this experiment. The best separation of data was achieved using $n\_components = 3$ and $covariance\_type = diag$ (meaning diagonal). Using features ("Average", "Count", "Trigger"), it produced a separation of data vertically and horizontally in contrast to DBSCAN (Fig 8).