Ahmedabad University

CSE523 Machine Learning

Weekly Report - 1

Project 5: Identify Hard stop and momentary stop using vehicle trajectory dataset.

Submitted to faculty: Mehul Raval

Date of Submission: 09-03-24

Roll No.	Name of the Student
AU2140040	Kathan Thakkar
AU2140171	Harsh Pandya
AU2140224	Dhruvi Rajput
AU2140230	Yax Prajapati

AIM OF THE WEEK:

Further Brainstorming, Reviewing algorithms that can be used and finding and reviewing the type of dataset required.

Introduction:

In this report, we have accumulated all the findings and progress that has been made for the selection of the clustering algorithm and a relevant dataset that we would like to use for this project.

Brainstorming:

Here are the few ways listed below which can be used to find the hard stop and soft stop.

Approach 1:

- Given the data first clean it and remove the unwanted part.
- Now cluster it based on the condition of v = 0 using DBSCAN as vehicle will be in stop condition. Now we have got the range of coordinates where vehicle stops.
- From the cluster get the range of coordinates.
- Check the velocity and acceleration of vehicles in the clusters at different times. This can we used to classify if it is hard stop or soft stop. As place where there is hard stops vehicles can't pass at high speeds while in soft stop if single is open then they can and this can be used to classify.

Approach 2:

- Given the data first clean it and remove the unwanted part.
- Now cluster it based on the condition of v = 0 using DBSCAN as vehicle will be in stop condition. Now we have got the range of coordinates where vehicle stops.
- After getting range of coordinates we can check velocity before vehicles arrive in range of the coordinates using time series algorithms. This can be used to identify the type of stop.

Approach 3:

- Given the data first clean it and remove the unwanted part.

- Now cluster it based on the condition of v = 0 using DBSCAN as vehicle will be in stop condition. Now we have got the range of coordinates where vehicle stops.
- Now we find the frequency of vehicles passing through the range of coordinates to classify.

Discovering new clustering algorithms

K-means clustering for spatio-temporal analysis:

K-means clustering can be a useful tool for spatio-temporal analysis, which involves analyzing data that has both spatial (locational) and temporal (time-based) components. Requirements needed to perform k-means clustering for spatio-temporal analysis:

Data:

1. Spatio-temporal data:

This is the core of your analysis. It should include location data (latitude, longitude, or other relevant spatial coordinates) and a corresponding timestamp for each data point. Examples include GPS data of moving objects, weather data collected at different locations over time, or crime incidents with location and time details.

2. Data cleaning and preprocessing:

Ensure the data is clean and formatted correctly. This may involve handling missing values, outliers, and ensuring consistent time formats.

Algorithm Requirements:

1. K-means implementation:

You'll need a software library or tool that implements the k-means algorithm. Many data analysis platforms and programming languages (Python, R) have built-in k-means functions.

2. Distance metric:

Since you're dealing with spatio-temporal data, a distance metric that considers both spatial and temporal aspects is needed. Euclidean distance might not be suitable here. Some options include spatio-temporal distance metrics that account for both spatial distance and temporal difference between data points.

DBScan algorithm for spatio-temporal analysis:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) can be effectively adapted for spatio-temporal analysis. Requirements needed to use it:

Data Requirements:

1. Spatio-temporal Data:

The data should include both spatial and temporal attributes for each data point. This could be latitude, longitude, and timestamp for tracking moving objects, or location and time of events.

Algorithm Requirements:

1. Distance Metric:

Define a distance metric that considers both spatial and temporal aspects. This could be a combined Euclidean distance for space and a difference metric for time (e.g., time difference between timestamps).

2. Epsilon (ε):

This parameter defines the maximum distance between points to be considered neighbors. It determines the granularity of clusters in both space and time.

3. MinPts:

This parameter defines the minimum number of neighbors a point needs to be considered a core point and form a cluster. It controls the density threshold for cluster formation.

Additional Considerations:

1. Parameter Tuning:

Epsilon (ε) and MinPts are crucial for successful clustering. There are techniques like k-distance graphs to help estimate these values, but might need to experiment to find the optimal settings for the data.

2. Variations:

Several variations of ST-DBSCAN exist, like using separate epsilon values for spatial and temporal aspects or incorporating weighting schemes for space and time.

Link:
https://www.researchgate.net/publication/223059919_ST-DBSCAN_An_algorithm_for_clustering_spatial-temporal_data?__cf_chl_rt_tk=3pZag1ATwk7tW8cEOt9kFhd1Zp1Wkn73FSDRwMvK0Nc-1709977550-0.0.1.1-1834

Difference between K mean and DBSCAN algo:

K means clustering	DBSCAN clustering
Clusters is sensitive to the number of clusters specified	Number of clusters need not to be specified.
Clusters formed may have more or less spherical shape	Shape of the cluster may be arbitrary and may not have the same feature size.
In the field where one need to work with anomalous data, This algorithm causes problems as anomalous points will be assigned to the same clusters as “normal” data points	This algorithm, on other hand, locates regions of high density that are separated from one another by regions of low density.
This may not work fine with noisy data	This might work well with noisy data
It requires one parameter: Number of clusters (K)	It requires two parameters: Radius and minimum points. Radius: to include enough points within it, it is a dense area. Minimum points: number of points required in a neighborhood to be defined as cluster.

K-means clustering allocates each point to a cluster, even if it does not belong to a specific cluster. As a result, noise points or outliers might distort the clustering results. It assumes that clusters are spherical and equal in size. This assumption may not be valid for vehicle trajectories, as stop conditions at signalized junctions and parked cars might differ in form and density. This also necessitates that the number of clusters (K) be given beforehand. However, in our project, the number and distribution of stop conditions (clusters) might fluctuate, making it difficult to find an adequate value for K.

DBSCAN may recognise noise points as outliers and does not assign them to a cluster. This feature enables DBSCAN to successfully handle outliers such as sporadic movements or mistakes in trajectory data, which is critical for discriminating between actual stops and random variations in vehicle movement. DBSCAN makes no assumptions regarding cluster structure or density. Instead, it detects clusters based on the density of points in a given neighborhood. This versatility enables DBSCAN to capture the irregular forms and changing densities of halt circumstances. DBSCAN does not require the number of clusters to be specified beforehand. Instead, it automatically detects clusters based on the density of points. This adaptability is advantageous for analyzing vehicle trajectories, where the number and location of stop conditions may vary over time and space.

Progress on dataset:

Team was focused on finding and identifying appropriate datasets that could be suitable for our project needs. We looked at a variety of sources, including academic archives, government transportation organizations, and open data portals. Despite our best attempts, we were unable to locate a dataset that met our unique needs for analyzing vehicle trajectories at signalized junctions and parked cars. We approached our TA for assistance with finding an appropriate dataset for our assignment. We described our project's objectives and the precise parameters we were looking for in the dataset. The TA acknowledged our request and agreed to assist us in finding a dataset that fit the requirements of our research.

Why using DBScan ?

1. DBScan has the ability to handle clusters of arbitrary shapes and sizes. Stop conditions at intersections and can be diverse and complex in nature. Therefore, DBScan can handle these diverse patterns of vehicle movements.
2. Based on the density of data points inside a certain radius (epsilon), DBSCAN automatically identifies clusters. Because it enables the algorithm to adjust to differing concentrations of stops across different sections of the intersection, this is very helpful for identifying stop circumstances.
3. It is possible that the number of unique stop conditions (clusters) will not be determined beforehand. DBSCAN works well in situations where the number of clusters is unknown or fluctuates at distinct intersections because it doesn't require you to define the number of clusters in advance.
4. In comparison to other clustering algorithms like K-means, DBSCAN is less sensitive to parameter selection, even if it still involves tweaking of parameters like epsilon (ε) and minPts (minimum number of points). As a result, there is less chance of overfitting and DBSCAN is more resilient to changes in intersection data.
5. DBSCAN is an effective method for analyzing big datasets of vehicle movements across crossings because of its comparatively low computing complexity. Because of its nearly linear time complexity in relation to dataset size, intersection data analysis is made scalable.

Conclusion:

So this week, we focused on several algorithms.We gain a good understanding of each of them in order to finalize a clustering algorithm for our project. We found all the differences between the algorithms to consider for selection.And eventually, we finalized the DBScan algorithm to use for our project.Moreover, we are looking forward to finding a perfect dataset for this project.

References:

Jain, S. (2022, October 31). *Difference between K-Means and DBScan Clustering*. GeeksforGeeks. Retrieved March 9, 2024, from https://www.geeksforgeeks.org/difference-between-k-means-and-dbscan-clustering/

rbhatia46/Spatio-Temporal-DBSCAN: Spatio Temporal DBSCAN algorithm in Python. Useful to cluster spatio-temporal data with irregular time intervals, a prominent example could be GPS trajectories collected using mobile devices. (n.d.). GitHub. Retrieved March 9, 2024, from https://github.com/rbhatia46/Spatio-Temporal-DBSCAN

ST-DBSCAN: An algorithm for clustering spatial–temporal data | Request PDF. (n.d.). ResearchGate. Retrieved March 9, 2024, from https://www.researchgate.net/publication/223059919_ST-DBSCAN_An_algorithm_for_clustering_spatial-temporal_data