Data Stream Processing Method for Clustering of Trajectories

Gary Reyes[1](✉), Laura Lanzarini[2], César Estrebou[2], and Aurelio Bariviera[3]

[1] Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil,
Cdla. Universitaria Salvador Allende, Guayaquil 090514, Ecuador
gary.reyesz@ug.edu.ec

[2] Facultad de Informática, Instituto de Investigación en Informática LIDI (Centro CICPBA) 1900 La Plata, Universidad Nacional de La Plata, Buenos Aires, Argentina
{laural,cesarest}@lidi.info.unlp.edu.ar

[3] Department of Business, Universitat Rovira i Virgili, Reus, Spain
aurelio.fernandez@urv.cat

Abstract. The constant advances in techniques for recording and collecting GPS trajectory information, the increase in the number of devices that collect this type of information such as video cameras, traffic sensors, smart phones, etc., has resulted in a large volume of information. Being able to process this information through data streams that allow intelligent analysis of the data in real time is an area where many researchers are currently making efforts to identify solutions. GPS trajectory clustering techniques allow the identification of vehicle patterns over large volumes of data. This paper presents a method that processes data streams for dynamic clustering of vehicular GPS trajectories. The proposed method here receives a GPS data stream, processes it using a buffer memory and the creation of a grid with the use of indexes, and subsequently analyzes each cell of the grid with the use of a dynamic clustering technique that extracts the characteristics of reduced zones of the study area, visualizing common speed ranges in interactive maps. To validate the proposed method, two data sets from Rome-Italy and Guayaquil-Ecuador were used, and measurements were made of execution time, used memory and silhouette coefficient. The obtained results are satisfactory.

Keywords: Buffer memory · Clusters · Trajectories · Cells · Data stream

1 Introduction

Nowadays, the constant increase in traffic volume in large cities causes problems in vehicular flow, so the analysis of generated data by vehicle monitoring and control systems, as well as the processing of GPS trajectories, becomes relevant [29]. Its study by means of descriptive techniques allows the identification of

R. Valencia-García et al. (Eds.): CITI 2022, CCIS 1658, pp. 151–163, 2022.
https://doi.org/10.1007/978-3-031-19961-5_11

relationships between vehicle trajectories, facilitating the analysis of vehicle flow. Currently, descriptive techniques provide solutions in a wide range of areas, such as health, finance, telecommunications, agriculture and transportation, among others [17].

Data clustering is a descriptive technique widely used in data mining to identify common characteristics between instances of the same problem [22]. Over time, researchers have proposed improvements to the identified limitations in some techniques, Bahmani et al. [6] achieved a correct initialization of the algorithm in a much shorter time, Dafir et al. [9] use parallel computing to improve the efficiency of algorithms or Han et al. [15] who use neural networks to improve the performance of the models. In other cases, techniques have been adapted to work in a specific context such as for spatial data mining [14,30,31], for GPS trajectory analysis [24] and even for real-time motion trend search [23] taking into account the dynamic nature of the data [12].

This work proposes a method that processes data streams for dynamic groupings of GPS trajectories, achieving a low memory consumption and a shorter processing time, which allows an agile analysis of the vehicular flow at a given time. A GPS trajectory is defined by a set of geographic locations, each of which is represented by its latitude and longitude, at an instant of time. As the GPS trajectory data collection progresses, the information in each cell of the grid is updated to reflect its average speed over a given period of time. These cells are delimited at the beginning of the process and their size depends on the desired accuracy of the analysis within the study area. The processing of this new representation in cells is analyzed using as a basis a dynamic clustering methodology of batch processing [26], which was adapted in this work to process data online and to which was incorporated the management of a buffer memory and the use of indexes in the creation of cells, allowing to reduce the memory consumption and the needed time to perform the calculations. As a result, areas with similar characteristics can be identified and an interactive map is generated in real time on which the speed ranges corresponding to the current vehicular flow and the areas where they occur can be observed. The comparison between the batch method and the proposed method will allow identifying the obtained differences for the memory consumption, the used execution time and the Silhouette coefficient assessment of the clusters.

This proposed method can be used, together with other tools, by traffic managers in a city to plan urban roads, detect critical points in traffic flow, identify anomalous situations, predict future mobility behavior, analyze vehicular flow, among others. The method can be used to characterize data corresponding to GPS trajectories generated by a group of students from the University of Guayaquil in Ecuador and historical cab data from the city of Rome in Italy. The obtained results allow the identification, in each city, of different time instants where vehicles have common speeds, and through the measurements of execution time, used memory and Silhoutte coefficient, a greater efficiency than the batch method is evidenced.

This article is organized as follows: Sect. 1 discusses some related works that were identified in the literature and present various solutions to the problem, Sect. 2 describes the proposed processing method used, Sect. 3 presents the obtained results and finally Sect. 4 contains the conclusions and lines of future work.

1.1 Related Work

Clustering techniques have been used in trajectory analysis for several years. They are usually adaptations of conventional algorithms using similarity metrics specially designed for trajectories [8,18,32]. Such is the case of the Enhanced DBScan algorithm [21] which improves the traditional DBScan algorithm by using a proprietary density measurement method suggesting the new concept of motion capability and the introduction of data field theory. On the other hand, Ferreira et al. [11] have presented a new trajectory clustering technique that uses vector fields to represent the cluster centers and propose a definition of similarity between trajectories. Research efforts in this area continue today, as evidenced by several research papers [16,20].

Certain treatments can be considered in trajectory clustering, such as segmentation, dynamic clustering, or online processing of data streams. An important feature that must be taken into account when performing dynamic clustering is the way in which the centers of the groups are represented [27,28]. Most of the generated data in a data stream requires real-time data analysis and the used data must manifest in the output within seconds [10,19]. The main goal of data stream clustering is to recognize patterns in data, which arrive at varying speeds and structures and evolve over time [4].

In summary, it can be stated that clustering techniques have proven to perform well in the analysis of vehicular trajectories although their parameterization remains an interesting challenge. This is related to the fact that they are unsupervised techniques that usually combine distance and density metrics to control the construction of the clusters.

In this paper we have used a dynamic clustering algorithm for data streams. This type of algorithms process data streams managing to overcome some of the limitations of traditional clustering algorithms, which usually iterate over the data set more than once, causing higher memory usage and increasing the execution time [5,13]. This is of great importance for systems that depend on accurate results especially in real-time environments [25]. As the data distribution of each stream changes continuously, it is important that these clustering algorithms that process data streams generate dynamic groups, where the number of groups will depend on the distribution of the stream data [1,2].

In particular, this work uses a variation of the DyClee algorithm defined originally by Barbosa et al. [7], a dynamic clustering algorithm for tracking evolving environments capable of adapting the clustering structure as data are processed. Dyclee uses a two-stage clustering approach based on distance and density [3]. The used methodology in this work was previously defined by Reyes et al. [26]. As an interpretation tool for the user, the use of interactive automatic maps is incorporated to facilitate the visualization of the clustering result. In the published article by Reyes et al. [26], the proposed methodology was used to analyze defined trajectories over the city of Guayaquil-Ecuador in order to automatically obtain the frequent speed ranges for different time intervals.

In this work, the batch processing [26] was adapted to work on a micro-batch basis and a method was employed to improve the performance of the process of creating cells in a grid, making use of a buffer and indexes to optimize the time and amount of memory needed.

A method is proposed that, unlike traditional methods that process trajectories [16], consolidates the information of a given area and transforms it into cells with summarized information. This information is processed by a modified clustering algorithm defined by Reyes et al. [26] to use a different dimension, obtaining clusters that reflect a different perspective of patterns. To facilitate the exploration and visualization of the results, the use of an interactive tool is proposed.

2 Proposed Method

This work proposes an adapted method for online processing or micro-batches for dynamic clustering of trajectories in large volumes of information based on batch processing, using a data buffer that reduces required memory consumption to process large amounts of GPS trajectory information. In addition, the proposed method makes use of indexes to improves the processing that transforms the GPS data into cells will be used by the clustering algorithm. The data corresponding to GPS points are presented as consecutive micro-batches of ordered data with respect to the time stamp of each record.

2.1 General Processing

Unlike batch processing, whose required data for processing are previously obtained from some repository, in online or micro-batch processing the data are received as the required time for clustering execution elapses.

The construction of the method contemplates the use of sequential processes, which are created and executed when required during processing.

The first processing step is the reception of GPS data, which are collected from the repository progressively over a certain period of time, which in this work was set at three minutes, and stored in a buffer memory. The second step is responsible for the transformation of these GPS data to summarized information in cells using indexes to improve processing performance; then in a third step the cells are clustered according to the speed ranges, generated during the clustering process. The last step is the generation of interactive maps for the visualization of clustering results. A general scheme of the micro-batch processing can be seen in Fig. 1, the processes used for the treatment of each cycle, which involves the processing of all the steps of the method for a data flow, are repeated iteratively; to process each set of data, the necessary cycles will be performed until the required period of time is covered and a real time processing will be used.

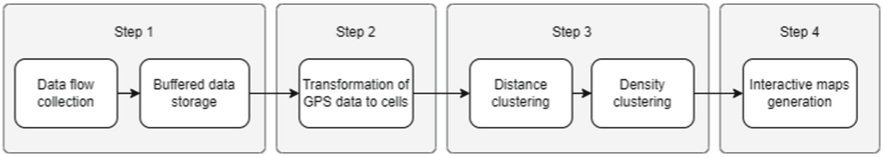

Fig. 1. General processing scheme of the micro-batch method

Step 1: Use of a Buffer. A part of the processing is in charge of storing in a temporary memory constantly GPS data collected from some repository and according to the period of time being analyzed, in order to contemplate different execution periods, use is made of a temporary storage memory that selects and stores small batches from the entire main data set; this processing called micro-batch allows the buffer of the micro-batch method does not require loading in memory the entire data, which reduces the consumption of this and allows to reduce considerably the calculations necessary for the generation of grid cells.

Each GPS data contains its respective time stamp that is not modified, which ensures that each data is allocated to a single buffer. The size of this temporary buffer can contain from one to a number n of GPS data, and this capacity will be smaller if the collection intervals of the data streams for the buffer decrease. The GPS data collected by a buffer is organized temporally in a sorted list according to the time stamp, this allows to maintain a consistency in the collection and will avoid errors in future processes.

Step 2: Use of Indexes to Create Cells with Summary Information. A cell is the representation of a data set with summarized information over a delimited area or grid.

Use of Indexes. The cell creation process makes use of indexes that are generated from single cells containing GPS information records. This process starts with the generation of indexes corresponding to the location of the cells; this generation evaluates each GPS point on the areas of the cells or grid, using the corresponding axes on the plane, so that an index for longitude and one for latitude are obtained independently of each other. Then we proceed to the extraction of these indexes and a subsequent filtering from which only a list of unique indexes containing GPS data will be obtained; one of the advantages of performing this step is to discard from the processing those cells that do not contain any GPS point assigned to them.

Shape Conformation of the Cells. The next step is to use the unique index list, which determines the cells to be processed for the transformation of the data into cells containing summary information. The cell formation consists of the classification and transformation of each GPS data into cells. During the transformation, the information in common of the GPS points is extracted and a summary of this data is obtained, which will form the contained information in each of the cells.

Unlike the cell creation process using the batch method whose representation can be seen in Fig. 2 (A), and which performs a cell to cell traversal, which is not very efficient because it evaluates all the cells of the area or grid, even evaluating cells in which there is no GPS point on that area; using the micro-batch method whose representation can be seen in Fig. 2 (B), the evaluation of cells that do not contain data is avoided, obtaining a better performance for this process.

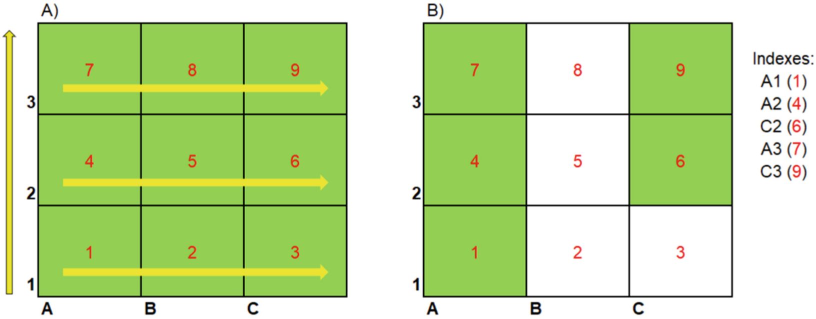

Fig. 2. Differences between the used methods in the formation of cells

Step 3: Use of a Clustering Technique. The cells are then clustered according to their velocity characteristic.

Clustering by Distance. The operation of this clustering process consists of identifying nearby clusters based on the velocity of the cells; the assignment of a cell to some group is determined by the smallest difference in velocities between the cell and the nearby clusters.

Density Clustering. In this clustering, classification of existing clusters is performed to categorize them into dense or sparse groups based on the number of GPS points contained in the cells within the clusters.

Step 4: Visualization of Results. After obtaining the clusterings of the cells, an interactive map is generated in which each resulting clustering can be visualized for each processed cycle and represented by a color scale according to the velocity of each cluster. Figure 3 shows the representation of the clusterings on an interactive map.

3 Results

3.1 Used Data

A collected dataset on the city of Guayaquil-Ecuador and a extracted dataset from a public repository belonging to the city of Rome-Italy were used to validate

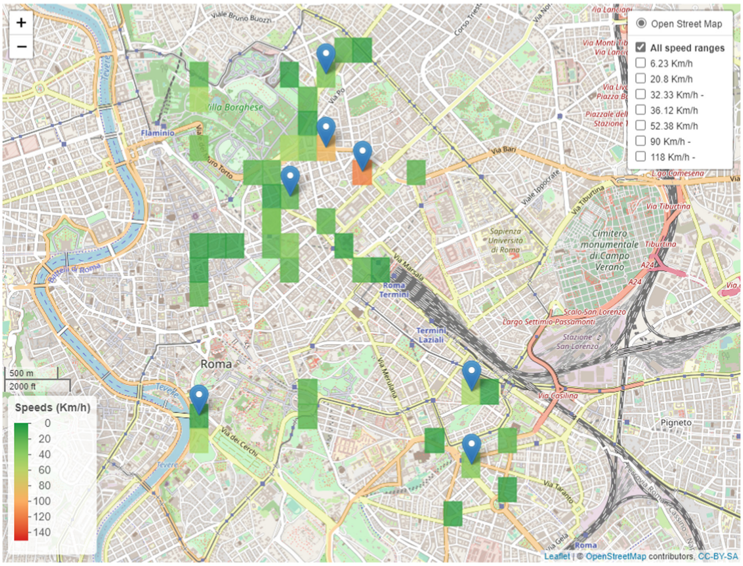

Fig. 3. Visualization of clustering results for the city of Rome in a time period of 3 min.

the micro-batch method. In the data selection process, for each of the data sets, first the day with the highest number of stored records was established and then the period of time with the highest number of vehicles circulating in the selected areas for analysis was identified (Guayaquil resulted in 30557 GPS points and Rome resulted in 33793 GPS points). A description of each dataset is presented below:

Guayaquil Dataset. It is a dataset collected by university students[1] tracing routes by means of some means of transportation such as cabs or motorcycles, the data corresponds to 218 trajectories, collected on October 28, 2017. The collection method used data collection from smartphone devices at 5-s intervals between consecutive locations. The structure of the collected records include id_trajectory, latitude, longitude, time, username, email, and transport type. The used data for this dataset comprises a time period between 16:30 and 18:30 due to the highest concentration of records. As a result of this filtering process, 30557 records were obtained, representing 206 trajectories of the entire dataset.

Rome Dataset. The data in this dataset collected over the city of Rome[2] was collected from GPS devices located in cabs corresponding to the day February

[1] Guayaquil dataset is available at https://github.com/gary-reyes-zambrano/Guayaquil-DataSet.

[2] Roma dataset available at https://github.com/gary-reyes-zambrano/Roma-Dataset.

12, 2014 and contains 137 trajectories with time intervals between records of 10 s. The record structure of this dataset contains id_trajectory, latitude, longitude, time, speed, direction. The analysis performed covers the time from 18:00 to 20:00 h, obtaining a total of 33793 records representing 137 trajectories of the entire dataset.

3.2 Obtained Results

Eight (8) consecutive runs were performed on the Rome and Guayaquil datasets. Each run contemplates data of 15 min duration, so for Rome and Guayaquil two(2) hours of data streams were considered for processing.

The results are presented below for the achieved measurements by the run time, silhoutte coefficient and used memory indicators.

Execution Times. During the execution of the method, the execution times required for each part of the processing of both the batch method and the micro-batch method were measured, until the clustering results were obtained.

The times have been calculated after having performed in sequence eight(8) runs containing data in periods of duration of 3 min. The results in minutes, can be observed for Rome in Table 1 and for Guayaquil in Table 2, and it is evident that the micro-batch method obtains better total times than the batch method.

The results of the average duration of the execution times of each cycle, measured in seconds for the eight (8) executions can be observed in Table 3 and show that the micro-batch method has shorter times.

Table 1. Rome execution times

	Batch method every 3 min (min)	Micro-batch method every 3 min (min)
Preprocessing	32:59	00:00
Execution 1	02:42	03:10
Execution 2	00:58	03:41
Execution 3	01:08	02:50
Execution 4	00:57	03:31
Execution 5	01:25	03:46
Execution 6	01:20	03:48
Execution 7	01:16	02:49
Execution 8	01:20	03:17
Total Time	**44:06**	**26:52**

Unlike the batch method, in which the cell conformation is a simple process that is performed separately and evaluates all the cells, the micro-batch method

Table 2. Guayaquil Execution Times

	Batch method every 3 min (min)	Micro-batch method every 3 min (min)
Preprocessing	24:59	00:00
Execution 1	00:08	01:02
Execution 2	00:10	02:07
Execution 3	00:54	01:59
Execution 4	01:09	03:13
Execution 5	00:44	01:39
Execution 6	00:19	02:13
Execution 7	00:22	01:17
Execution 8	00:09	01:19
Total Time	**28:54**	**14:48**

Table 3. Average Time per Cycle (in seconds)

	Batch method every 3 min (secs)	Micro-batch method every 3 min (secs)
Rome	59,94	13,43
Guayaquil	45,76	23,38

applies improvements to this process and performs it when the buffer is received, first identifying in each buffer only the cells that contain a GPS point through the use of indices and then performs the conformation of these cells, thus avoiding the analysis of cells that do not contain any GPS data. This is shown in Table 1 and Table 2 when observing the times in each execution, in the micro-batch method the times are higher because they are considering the conformation of cells that in the batch method was performed in a preprocessing stage, the improvement is evidenced when totaling all the times where independently of the organization of the processes the micro-batch method has required less time.

Silhouette Coefficient. To determine the quality of the formed clusters, the use of the Silhouette coefficient was considered, which establishes a scale with values from −1 indicating that the elements might not have been assigned correctly, to 1 representing better clusters. The Silhouette score is based only on the elements of dense clusterings, excluding sparse clusterings.

The obtained results of the silhouette score for the two data sets are presented in Table 4, for the results of Rome the micro-batch method has obtained a lower score, but its deviation has been reduced considerably; for the results of Guayaquil, the results of the Silhouette coefficient have been higher and its deviation has also decreased.

According to the obtained results in the Rome runs it can be observed that the results of the batch method have a high mean dispersion compared to the micro-batch method which has a lower mean dispersion, in this sense it can be stated that the results of the micro-batch method present a better allocation of the points to the groupings.

In the case of Guayaquil, the scores of the batch method have a mean dispersion similar to that of the micro-batch method; however, the silhouette scores of the micro-batch method are higher.

Table 4. Silhouette Results Summary

	Rome		Guayaquil	
	Mean Score	Desv.	Mean Score	Desv.
Batch method every 3 min	0,567	0,029	0,545	0,114
Micro-batch method every 3 min	0,565	0,018	0,556	0,111

Used Memory by GPS Data. This memory corresponds to the weight of the GPS data when received and corresponds to the information of the records of each collected point by GPS devices.

With respect to the measurement performed, whose results measured in bytes can be observed in Table 5, it is observed a larger memory space used by the batch method, this is because all GPS data are kept loaded in memory during each execution, whereas in the micro-batch method GPS data are received in different data streams during the execution of the method. In this sense, in the micro-batch method the memory space allocated is released as the different cycles are processed over time and varies depending on the volume of collected data for a cycle and the processing frequency. In addition to the above, the micro-batch method shows that the resulting weights in Guayaquil are lower than those in Rome due to the inclusion of additional filters that allow data cleaning to eliminate atypical data such as recorded data with zero velocities, resulting in a lower consumption of memory space.

Table 5. Average of Used Memory Space by GPS data (in bytes)

	Batch method every 3 min (bytes)	Micro-batch method every 3 min (bytes)
Rome	1218192	227581
Guayaquil	1309824	211865

Used Memory by Cells. This memory corresponds to the weight measured after having transformed the GPS point data into divisions of cells with fixed dimensions whose information contemplates a summary of the GPS points that are located on the areas of each cell, this causes that the memory space measured by the cells are lower if compared with the used memory by the GPS points.

The results of the measurements on the different datasets can be seen in Fig. 4, the average of used memory during the eight (8) executions and obtained after the transformation of the data into cells in the micro-batch method are lower compared to the Batch method.

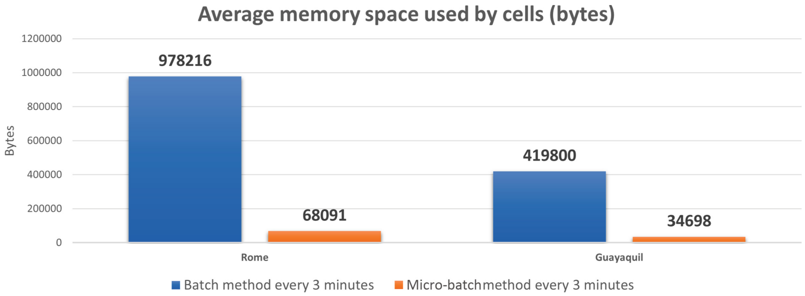

Fig. 4. Average of used memory space by cells

4 Conclusions

In this paper, a method that processes data streams for dynamic clustering of vehicular GPS trajectories has been proposed.

The proposed method receives a data stream, then by means of a buffer memory and the creation of cells with indexes it processes the stream that will be analyzed by a dynamic clustering technique; in order to do this, the trajectory information has been represented in cells and the clustering of cells by the speed dimension has been performed. Finally, the results of the clustering cells are visualized in interactive maps that allow observing the different groups, identifying common speeds at different time instants, which allows making decisions regarding the traffic of a city.

The execution time, the Silhouette coefficient, and the used memory were measured; the results are favorable for the proposed method. As lines of future work, it is proposed to analyze the implementation of the method using a platform for real-time processing and for distributed data processing through a parallel processing architecture.

Compared to traditional methods, the proposed method has advantages such as low memory consumption, low processing times, high quality clustering, the results are displayed by cells which allows reflecting the traffic status in certain areas.

On the other hand, among the disadvantages of the proposed method, it has been identified that the performance of the algorithm is relative to the amount of data being processed, the incorrect calibration of initial parameters can affect the quality of the results, and the method requires historical information, that is, for a certain number of minutes before showing optimal groupings.

References

1. Ackermann, M.R., Lammersen, C., Sohler, C., Swierkot, K., Raupach, C.: StreamKM++: a clustering algorithm for data streams. ACM J. Exp. Algorithmics **17**, 173–187 (2012)
2. Aggarwal, C.C.: Data streams: an overview and scientific applications. In: Gaber, M. (ed.) Scientific Data Mining and Knowledge Discovery. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-02788-8
3. Aggarwal, C.C., Yu, P.S., Han, J., Wang, J.: A framework for clustering evolving data streams. In: Freytag, J.C., Lockemann, P., Abiteboul, S., Carey, M., Selinger, P., Heuer, A. (eds.) Proceedings 2003 VLDB Conference, pp. 81–92. Morgan Kaufmann, San Francisco (2003). https://doi.org/10.1016/B978-012722442-8/50016-1, www.sciencedirect.com/science/article/pii/B9780127224428500161
4. Ahmed, R.: Stream clustering (2020). https://doi.org/10.13140/RG.2.2.18295.04007
5. Babcock, B., Widom, J.: Models and Issues in Data Stream Systems (2002)
6. Bahmani, B., Moseley, B., Vattani, A., Kumar, R., Vassilvitskii, S.: Scalable k-means++ (2012)
7. Barbosa Roa, N., Travé-Massuyès, L., Grisales-Palacio, V.H.: DyClee: dynamic clustering for tracking evolving environments. Pattern Recognit. **94**, 162–186 (2019). https://doi.org/10.1016/j.patcog.2019.05.024https://www.sciencedirect.com/science/article/pii/S0031320319301992
8. Choong, M.Y., Chin, R.K.Y., Yeo, K.B., Teo, K.T.K.: Trajectory pattern mining via clustering based on similarity function for transportation surveillance. Int. J. Simul.-Syst. Sci. Technol. **17**(34), 1–19 (2016)
9. Dafir, Z., Lamari, Y., Slaoui, S.C.: A survey on parallel clustering algorithms for big data. Artif. Intell. Rev. **54**(4), 2411–2443 (2021). https://doi.org/10.1007/s10462-020-09918-2
10. Ding, S., Wu, F., Qian, J., Jia, H., Jin, F.: Research on data stream clustering algorithms. Artif. Intell. Rev. **43**(4), 593–600 (2015). https://doi.org/10.1007/s10462-013-9398-7
11. Ferreira, N., Klosowski, J.T., Scheidegger, C., Silva, C.: Vector field k-means: Clustering trajectories by fitting multiple vector fields (2012)
12. Fotakis, D., Piliouras, G., Skoulakis, S.: Efficient online learning for dynamic k-clustering (2021). arXiv:2106.04336, https://doi.org/10.48550/ARXIV.2106.04336
13. Garofalakis, M., Gehrke, J., Rastogi, R.: Data Stream Management (2016)
14. Han, J., Kamber, M., Tung, A.K.: Spatial clustering methods in data mining. Geographic data mining and knowledge discovery, pp. 188–217 (2001)
15. Han, P., Wang, W., Shi, Q., Yue, J.: A combined online-learning model with k-means clustering and GRU neural networks for trajectory prediction. Ad Hoc Networks 117, 102476 (2021). https://linkinghub.elsevier.com/retrieve/pii/S1570870521000433, https://doi.org/10.1016/j.adhoc.2021.102476

16. Hu, H., Lee, G., Kim, J.H., Shin, H.: Estimating micro-level on-road vehicle emissions using the k-means clustering method with GPS big data. Electronics **9**(12), 2151 (2020)
17. Jain, A.: Data clustering: 50 years beyond k-means. 2009. Pattern Recognition Letters (2009)
18. Kim, J., Mahmassani, H.S.: Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories. Transp. Res. Procedia **9**, 164–184 (2015)
19. Kolajo, T., Daramola, O., Adebiyi, A.: Big data stream analysis: a systematic literature review. J. Big Data **6**(1), 47 (2019). https://doi.org/10.1186/s40537-019-0210-7
20. Lou, J., Cheng, A.: Behavior from Vehicle GPS/GNSS Data. Sensors (2020)
21. Luo, T., Zheng, X., Xu, G., Fu, K., Ren, W.: An improved DBSCAN algorithm to detect stops in individual trajectories. ISPRS Int. J. Geo-Inf. **6**(3), 63 (2017). www.mdpi.com/2220-9964/6/3/63, https://doi.org/10.3390/ijgi6030063
22. Madhulatha, T.S.: An overview on clustering methods. arXiv preprint arXiv:1205.1117 (2012)
23. Mao, J., Song, Q., Jin, C., Zhang, Z., Zhou, A.: Online clustering of streaming trajectories. Front. Comput. Sci. **12**(2), 245–263 (2018). https://doi.org/10.1007/s11704-017-6325-0
24. Mazimpaka, J.D., Timpf, S.: Trajectory data mining: a review of methods and applications. J. Spat. Inf. Sci. **2016**(13), 61–99 (2016)
25. Paulino, D.C., Guimarães, L.N.F., Shiguemori, E.H.: Hybrid adaptive computational intelligence-based multisensor data fusion applied to real-time UAV autonomous navigation. Inteligencia Artif. **22**(63), 162–195 (2019). https://journal.iberamia.org/index.php/intartif/article/view/237, https://doi.org/10.4114/intartif.vol22iss63pp162-195
26. Reyes, G., Lanzarini, L., Estrebou, C., Maquilón, V.: Vehicular flow analysis using clusters, pp. 261–270 (2021)
27. Reyes, G., Lanzarini, L., Hasperué, W., Bariviera, A.F.: GPS trajectory clustering method for decision making on intelligent transportation systems. J. Intell. Fuzzy Syst. **38**(5), 5529–5535 (2020). www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/JIFS-179644, https://doi.org/10.3233/JIFS-179644
28. Reyes, G., Lanzarini, L., Hasperué, W., Bariviera, A.F.: Proposal for a pivot-based vehicle trajectory clustering method. Transp. Res. Rec. **2676**(4), 281–295 (2022). https://doi.org/10.1177/03611981211058429
29. Reyes, G., Maquilón, V., Estrada, V.: Relationships of compression ratio and error in trajectory simplification algorithms. In: Valencia-García, R., Bucaram-Leverone, M., Del Cioppo-Morstadt, J., Vera-Lucio, N., Jácome-Murillo, E. (eds.) Technologies and Innovation, pp. 140–155. Springer International Publishing, Cham (2021)
30. Tork, H.F.: Spatio-temporal clustering methods classification. In: Doctoral Symposium on Informatics Engineering, vol. 1, pp. 199–209. Faculdade de Engenharia da Universidade do Porto Porto, Portugal (2012)
31. Varghese, B.M., Unnikrishnan, A., Jacob, K.: Spatial clustering algorithms-an overview. Asian J. Comput. Sci. Inf. Technol. **3**(1), 1–8 (2013)
32. Wang, H., Sha, Y., Wang, D., Nazari, H.: A gene expression clustering method to extraction of cell-to-cell biological communication. Inteligencia Artif. **25**(69), 1–12 (2022). https://journal.iberamia.org/index.php/intartif/article/view/701, https://doi.org/10.4114/intartif.vol25iss69pp1-12