Stay Point Analysis in Automatic Identification System Trajectory Data

Yihan Cai[1], Menghan Tian[1], Weidong Yang[1], Yi Zhang[2]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Shanghai Urban-Rural Construction and Transportation Development Research Institute, Shanghai, China

Abstract - *With the increasing popularity of location acquisition technology, huge amounts of trajectory data are getting available, including Automatic Identification Systems (AIS) data for vessel traffic services. However, the spatial points in the trajectory data are not equally important, and one kind of trajectory points, named stop points, carry more semantic meanings than other spatial points in the trajectory. Extracting and analyzing stay points will greatly help the subsequent data mining and trajectory compressing. In our work, we use real-world AIS dataset, and implement a stay point analysis system for it, whose features include preprocessing, extracting and clustering stay points, and distributed computing on Spark.*

Keywords: Temporal and spatial data mining, Mining big data, Stay point, Trajectory data

1 Introduction

With the rapid development of mobile computing technology and the pervasiveness of location-acquisition technologies, large amounts of spatial-temporal trajectory data are getting increasingly available. In the past decades, many techniques have been proposed for processing, managing, and mining trajectory data. And applying these techniques enables us to discover valuable knowledge with both practical significance and application scenarios from trajectory data.

Yet before mining trajectory data, we need to deal with a number of issues, such as noise filtering, trajectory compressing, segmentation, which are collectively known as trajectory preprocessing. In this paper, we focus on stay point analysis, which is either one stage in preprocessing or the target of data mining, to identify places where moving objects have stayed for a while.

Stay points can carry more semantic meanings than other points in the same trajectory. For example, when studying traveler behavior, if a moving target remains stationary in a certain geographic area of his tourist destination for more than a given period of time, it indicates that there may be a popular scenic area or a traffic station, instead of an ordinary road that travelers just pass by. In this way, stay points help finding out more on topics like behavior patterns, user similarity, destination recommendation, etc. For another example, the stay point detection of the ship's trajectory data may help us find the popular port, and infer its scale and economic status from stay time. The stay points on the sea can also indicate the center of the vessel activity, such as nice fisheries. In addition, the stay point can also help figure out how to avoid the loss of data meaning when trajectory compression is needed.

Therefore, the stay point analysis can help to find valuable knowledge, no matter in the specific mining task, as a preprocessing step for the trajectory data from the real world, or considered as the very target of data analysis.

In this paper, we use AIS dataset, and implement a stay point analysis system for it, whose features include preprocessing (removing outliers), extracting single-trajectory stay points and clustering multi-trajectory stay points, and distributed computing on Spark.

2 Related works

In the field of mining individual trajectory records, driven by the increasing convenience of data collecting, many researches have been conducted in the past few years based on individual trajectory data, including detecting places of interest of users, predicting routines of users between these locations, and recognizing user-specific activities.

In some researches, the step of stay point analysis is considered as one part of preprocessing before data mining. By extracting stay points, we can transform the trajectory from a series of spatial points with timestamp to a series of meaningful points, and further get a lot of practical applications, such as travel recommendations. [Zheng and Xie 2011b; Zheng et al. 2011c], destination prediction [Ye et al. 2009], taxi recommendations [Yuan et al. 2011b, 2013b] and estimation of gas consumption [Zhang et al. 2013, 2015][5].

In some applications, such as estimating the travelling time of routes [Wang et al. 2014] and driving recommendations [Yuan et al. 2013a], stay points stand for logs that should be removed from the trajectory during preprocessing.

Li et al. [2] search for highly relevant information between users in order to mine the similarity between users based on their location history. They propose a

stay-point detection algorithm based on time and space, and use stay points to describe users' behavior patterns. The similarity measurement framework they proposed is constructed on the basis of the stay points. In their work, a general stay-point detection algorithm for single targets was proposed, which in the actual application needs to be modified to adapt to a specific scenario. Mining user similarity hierarchically on the basis of stay points is also a typical mining application.

Alvares et al. [6] mentioned that trajectory data are usually provided as sample points and do not contain semantic information that is essential for understanding these data, so trajectory data mining becomes expensive and complicated. If we can enrich the trajectory using semantic geographical information, we can simplify the query, analysis and mining of mobile object data. Therefore, they proposed a data preprocessing model to add the semantic information of candidate stay points to the trajectory so that trajectory data analysis can be performed in different application fields. The model has to some extent universality, and also reflects the importance of stay point analysis to simplify trajectory mining.

Ashbrook and Starner [9] propose that wearable computers have the potential to act as intelligent agents in daily life, and can determine by context which task to help the user to complete. Location is the most commonly seen form of context that these smart agents use to determine user tasks, and the location context can also be used to create a predictive model of the user's future movement. The system they proposed will automatically extract meaningful locations under multiple scales from GPS datasets collected over a long period of time. These stay points will then be merged into a Markov model to build a predictive model of user actions. This model can be used with various applications together.

Meanwhile in some work on trajectory data mining, the stay point is regarded as the target of mining.

Damiani et al. [1] give a time-aware, density-based clustering algorithm to identify meaningful regions in animal migration under low sampling rates. The stay points(areas) in their work stands for a geographical area where wild animals reside or forage. Their work defined parameters about density and presence to specify dense sub-trajectories which are temporally disjoint. This work was inspired from the perspective of animal ecology, but the methods proposed in this paper may also be applied to more general fields, such as human mobility research on large time scales.

Stylianou [10] mentioned that most of the stay point extraction algorithms rely on the use of experimental fine-tuning thresholds, whose accuracy may decrease under different data sets. Thus this work proposed a method that transforms the user's trajectory path into a two-dimensional discrete time serial curve, and stay point is converted into a local minimum of the first derivative of the curve. When there is a good trajectory sampling technique, the accuracy of this algorithm is high, but in other words it also shows that this algorithm is limited by the sampling technique. Besides, the algorithm cannot extract a region where the moving target wandering.

Kang et al. [8] described a time-based algorithm for extracting a stay point from trajectories so as to obtain important places for the user which could be used in location-aware systems installed on various mobile devices to help these systems understand meaningful places to a specific user, such as his home or working place. In order to obtain general moving patterns, this framework needs a long-time-span dataset.

3 Preliminary

In this section, we will define some terms used in this paper and introduce the basic architecture of our work.

Trajectory data (logs): Basically, trajectory data refers to the information obtained by sampling the motion of one or more moving objects in the spatio-temporal environment, including location, timestamp, velocity, etc. The data of these sampling points can be transformed into a sequence according to their time serials. That is to say, a trajectory $T= \{p_1, p_2, \dots, p_n\}$, while each $p_i \in T$ contains its geographic location and a timestamp such as (p_i.latitude), (p_i.longitude), and (p_i.timestamp), and sometimes other related information.

Stay points: A stay point is a geographic region where the moving object stays for a while, which carries more semantic meanings than other points in the trajectory. Since spatio-temporal points are not equally important in trajectories, extracting stay points does great help from compressing trajectories to analyze user patterns.

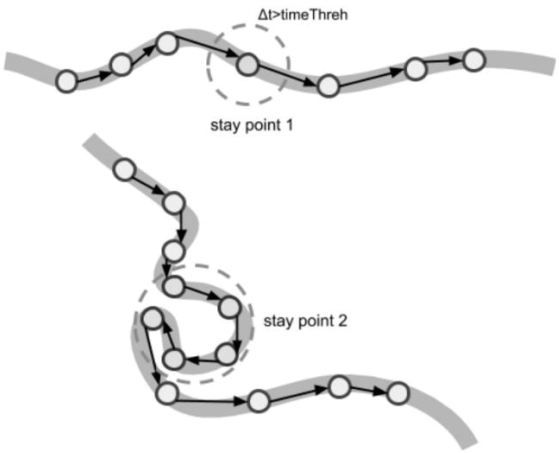

Fig.1 Two Categories of Stay Points

Stay points can be classified into two categories. The first category is one single location where the target object remains stationary longer than a time threshold t, such as stay point 1 in figure 1, where the target stays for a period of time. This kind of stay points often occurs when moving objects lose their signal at a certain position (e.g., the mobile phone loses its GPS signal when its customer is shopping in a department store) and finally restore after they come out. The second type, as stay point 2 shown in figure 1, is more commonly seen

in trajectories. It means that the moving object or user keeps moving around within an area (e.g., a visitor is taking a walk within the scenic area), or in fact the object stays still but its position readings are shifting around. Therefore, a class II stay point actually contains signals of multiple trajectory data.

Figure 2 demonstrates the framework of our experiment. In this paper, we designed and implemented a stay point analysis system for large-scale real-world AIS data, whose features include preprocessing (removing outliers), extracting single-trajectory stay points, and clustering stay points for multiple trajectories.

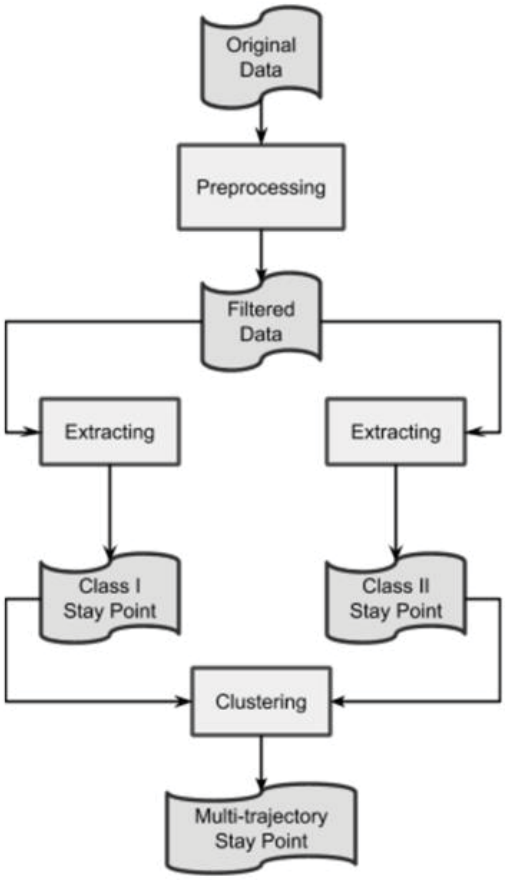

Fig.2 Stay Point Analysis System Framework

4 Algorithms for stay point analysis

4.1 Preprocessing

As the AIS signal might be unstable, and there might be some interference during the transmission of data, the existence of noises is inevitable in received data, which may affect the analysis of stay points. Therefore, data cleaning is a necessary step, where we use mean filter and median filter to detect outliers and remove them.

In the mean (or median) filter, the estimated value of one spatial point z_i is the average (or median) of z_i and its n-1 neighboring points, preceding and following. That is, the mean (median) filter can be considered as a sliding window covering the neighboring values of point z_i in the trajectory. When dealing with extreme outliers, the median filter has better robustness than the mean filter.

Both of these two filters are useful when removing with a single noise point, e.g., p_5 in Figure 3, in a highly-sampled trajectory. However, if multiple consecutive outliers occurred in the trajectory, such as p_{10}, p_{11},and p_{12}, the median and average filters will require a larger-sized sliding window to work properly, which can lead to greater errors between estimated values and actual values. So when the sampling rate of the trajectory is very low (e.g., the distance between two consecutive points may be larger than hundreds of meters), mean or median filters are no longer a good choice.

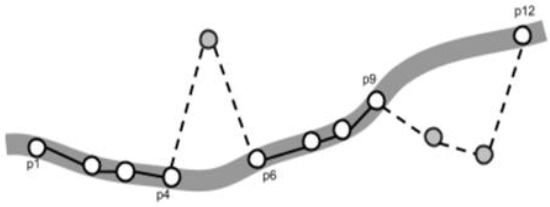

Fig.3 Mean/Median Filter on Trajectory

Since the data sets we has over 600 million of high-sampling-rated data, we can use these two types of filters which are easier to implement and apply under this amount of data.

4.2 Extracting single-trajectory stay points

For every single trajectory, stay points can be divided into two categories. The first category of stay points is a location where the moving target remains stationary for a period of time, for example, a harbor. And the second category represents for one area where the target wanders around, or the target remains stationary actually but the signal readings keep moving around. In these two situation, the signal of moving target always stays within a certain range of distance (such as the movement of a fishing boat over a fishery).

Normally, we extract these two category of stay points according to their definitions. Videlicet, through finding the spatial distance where the ship spends over a certain threshold time, these stay points can be automatically detected from the AIS data.

The algorithm we use firstly checks whether the distance between one anchor point (for example, p_4 in Figure 4) and its successor is greater than a given threshold, and then calculates the time span between the anchor point and the last successor within the distance threshold (that is, the time span between p_4 and p_8). If the interval is greater than the given time threshold, a stay point is detected (indicated by p_4, p_5, p_6, p_7, and p_8). Hereafter, the anchor will be moved to the last successor to detect next possible stay point.

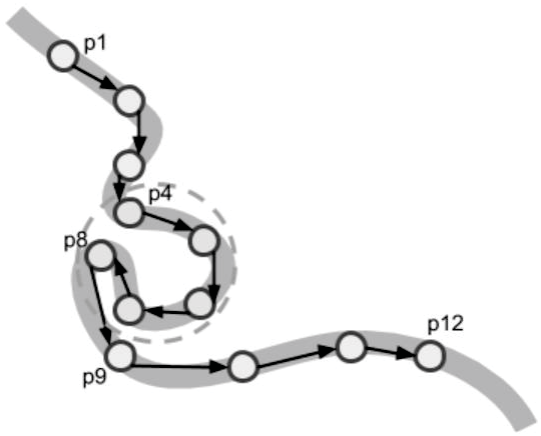

Fig.4 Detecting Stay Points Based on Time and Space

For every extracted stop point, we will store its geographic information (such as latitude and longitude), arrival time, departure time, and the ID of its original route.

There are two reasons why we choose to detect stay points based on spatio-temporal properties of AIS data. First, if we simply clustered all AIS points in all trajectories together based on density, we will possibly miss some important regions which carry semantic meanings. Because the AIS source may lose the satellite signal, or there are few AIS points generated in some weak signal areas, the density of trajectory points in these regions cannot meet the conditions for establishing clusters. Moreover, in this case, the places where ships repeatedly pass but do not have semantic meanings will be extracted instead. Since the number of AIS trajectory points will be much larger than the magnitude of stay points, the calculation of clustering will be particularly important (and difficult). Second, those grid-based partitioning methods, on the other hand, will introduce problems on grid boundaries, which can also lead to the loss of significant regions.

We implemented and modified our algorithm according to the stay point detection algorithm proposed by Li et al. [2].

```
Stay Point Detection Algorithm
Input: A AIS log P
       a distance threshold distThreh
       a time span threshold timeThreh
       a minimal number of AIS points in one stay point miniPoints
Output: A set of stay points SP
1. i=0, pointNum = |P|; //the number of AIS points
2. while i < pointNum do,
3.       j=i+1; Token=0;
4.       while j < pointNum do,
5.              dist=Distance(pi, pj);
                //calculate the distance between points
6.              if dist > distThreh then
7.                     ΔT=pj.T-pi.T;
                       //calculate the time span between two points
8.                     if ΔT>timeThreh and j>=i+miniPoints then
9.                            S.coord=GetCentroid({pk | i<=k<=j})
10                           S.arvT= pi.T; S.levT=pj.T;
11.                          SP.insert(S);
12.                          i=j; Token=1;
13.                    break;
14.             j=j+1;
15.     if Token!=1 then i=i+1;
16. return SP.
```

Fig.5 Stay Point Detection Algorithm

In order to restrict the minimal number of AIS points in one stay point of the second category, we have added constraints *miniPoints* to the algorithm. In addition, for the purpose of being able to process the continuous stream of trajectory data in real time, we implemented this algorithm on Spark to support distributed computing.

4.3 Clustering multiple-trajectory stay points

After obtaining the stay point data of each single trajectory, the stay points of multiple routes in the same type of vessels over a given duration can be calculated by clustering. In our work, we use the K-Means method based on Spark.ml.

There are two commonly used clustering methods in trajectory mining: K-Means and DBSCAN. K-Means method aims to divide a number of sample points into a predetermined K clusters. Clusters are compact inside and independent of each other. In other words, the clustering standard is that each point belongs to the cluster whose center is nearest to it, given the measure of distance. On the other hand, given a set of samples, DBSCAN will divide adjacent points into a set of high-density areas and mark those noise points in low-density areas.

For K-Means, the number of clusters K is an input parameter. Thus choosing an inappropriate K value may result in poor clustering results. In addition, K-Means method gives out a local optimal solution, which may lead to 'anti-intuitive' erroneous results. Compared with the K-clustering algorithm, the DBSCAN algorithm does not need to declare the number of clusters in advance. It also can find irregularly shaped clusters and identify noise points. However, if the points in the trajectory data have different densities that vary greatly, the clustering results of DBSCAN may be poor because it would be difficult to find the suitable pair of parameters for all clusters in this case.

In fact, each of these two algorithms has its own limitations. Taking in to account the magnitude of our AIS dataset, the varying density of single-trajectory stay points, and the demand of distributed computing, we choose K-Means in our framework.

5 Architecture

In our work, we use real-world AIS data as our dataset, with more than 600 million original data records in it and a sampling time span of six months. To support this amount of data, we store these AIS data in HDFS, use YARN for resource scheduling, and use Spark integrated Hive for distributed computing.

The table shows the amount of data used and produced in our work.

Table 1 Amount of Data

AIS Tables	Amount of Data
Original Data	682824994
Filtered Data	681122161
Category I Stay Point	1418155
Category II Stay Point	4395929
Stay Point Clusters	298063

The AIS data we use was collected from July to December in 2016 over Taiwan Strait from different types of vessels. AIS data needs to be sent every few seconds to ensure the timeliness of the information, so the sampling rate of dataset is rather high. Then the data, such as ship position, can be automatically transmitted from the sensor on vessels to the AIS receiver, including dynamic and static data of all vessels. And its high refreshing rate and real-time performance lead to a large scale of data volume.

Under the amount of data, our distributed system is built on 1 master node (name node) and 2 slave nodes (data node) with 64GB of memory. The performance would be improved greatly with more computing nodes and more memory since our framework is expandable.

The figure below illustrate the architecture of our system.

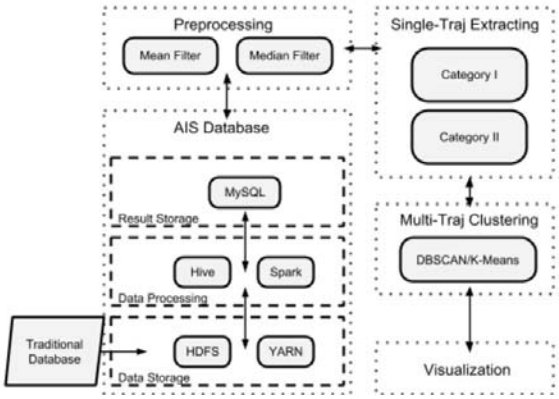

Fig.6 Architecture of Our Distributed System

The stay points obtained by multi-trajectory clustering can be visualized to figure out their locations in the actual geographical space clearly, thereby more intuitively helping to analyze the geographic distribution, scale index of the candidate place of interest, such as harbors, fisheries, etc.

6 Conclusion

With the popularity of location acquisition technique, the Automatic Identification System, as an automatic tracking system for vessel traffic services, generates huge amounts of spatio-temporal data on ship routes all over the world.

In this paper, we implemented AIS data stay analysis based on distributed environment. The features include: first, data cleaning was performed on the original data, i.e., using mean and median filters to remove outliers; second, single-trajectory stay points were extracted for two categories; at last, multiple-trajectory stay points were found through clustering single-trajectory ones.

The system is built on the Hadoop Distributed File System framework, using YARN for resource scheduling with a hierarchical data storage structure. Python was chosen for algorithm development and distributed computing was implemented on Spark.

Based on existing work, our work modified the algorithm for the second category of stay points on a single shipping line, and implemented it on the distributed framework. For future work, the clustering algorithm for multi-trajectory stay points can be further optimized using a variant of the density-based clustering algorithm.

Through stay point analysis, we can understand the application of knowledge in trajectory data deeper that can be subsequently mined in the real world, whether it is from the commercial scene of human life or the movement of biological and natural phenomena in nature.

7 Acknowledge

This work was supported in part by Shanghai Innovation Action Project (No.16DZ1100200, No. 16DZ1110102).

8 References

[1] M. L. Damiani, H. Issa, F. Cagnacci, Extracting stay regions with uncertain boundaries from GPS trajectories: a case study in animal ecology, SIGSPATIAL'14, November 04 - 07 2014, Dallas/Fort Worth, TX, USA.

[2] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, Wei-Ying Ma, Mining User Similarity Based on Location History, ACM GIS '08, November 5-7, 2008. Irvine, CA, USA.

[3] Khoa A. Tran, Sean J. Barbeau, Miguel A. Labrador, Automatic Identification of Points of Interest in Global Navigation Satellite System Data: A Spatial Temporal Approach, IWGS '13:, November 05 - 08 2013, Orlando, FL, USA.

[4] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, Loren Terveen, Discovering Personally Meaningful Places: An Interactive Clustering Approach, ACM Transactions on Information Systems, Vol. 25, No. 3, Article 12, Publication date: July 2007.

[5] Yu Zheng, Trajectory Data Mining: An Overview, ACM Trans. Intell. Syst. Technol. 6, 3, Article 29 (May 2015), 41 pages.

[6] Alvares, L. O., Bogorny, V., Kuijpers, B., et al, A model for enriching trajectories with semantic geographical information[A].Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems[C].2007.1-8.

[7] Palma, A. T., Bogorny, V., Kuijpers, B., et al., A clustering-based approach for discovering interesting places in trajectories [A]. Proc. of ACM Symposium on Applied Computing[C].2008.863-868.

[8] Kang, J. H., Welbourne, W., Stewart, B., et al., Extracting places from traces of locations[J], ACM SIGMOBILE Mobile Computing and Communications Review,2005,9(3):58-68.

[9] Ashbrook, D. & Starner, T. Pers Ubiquit Comput (2003) 7: 275. https://doi.org/10.1007/s00779-003-0240-0.

[10] G. Stylianou, Stay-point Identification as Curve Extrema, arXiv:1701.06276 [cs.OH].

[11] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma. 2010c. Understanding transportation modes based on GPS data for Web applications. ACM Transactions on the Web 4, 1 (2010), 1–36.

[12] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma. 2009a. GeoLife2.0: A location-based social networking service. In Proceedings of the 10th IEEE International Conference on Mobile Data Management. IEEE, 357–358.

[13] Y. Zheng and X. Xie. 2011b. Learning travel recommendations from user-generated GPS traces. ACM Transactions on Intelligent Systems and Technology 2, 1 (2011), 2–19.

[14] Y. Zheng, X. Xie, andW.-Y.Ma. 2010d. GeoLife: A collaborative social networking service among user, location and trajectory. IEEE Data Engineering Bulletin 33, 2 (2010), 32–40.

[15] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. 2011c. Recommending friends and locations based on individual location history. ACM Transaction on the Web 5, 1 (2011), 5–44.