Design of intelligent acquisition system for moving object trajectory data under cloud computing

Abstract: In order to study the intelligent collection system of moving object trajectory data under cloud computing, information useful to passengers and taxi drivers is collected from massive trajectory data. This paper uses cloud computing technology, through clustering algorithm and density-based DBSCAN algorithm combined with Map Reduce programming model and design trajectory clustering algorithm. The results show that based on the 8-day data of 15,000 taxis in Shenzhen, the characteristic time period is determined. The passenger hot spot area is obtained by clustering the passenger load points in each time period, which verifies the feasibility of the passenger load point recommendation application based on trajectory clustering. Therefore, in the absence of holidays, the number of passenger hotspots tends to be stable. It is reliable to perform cluster analysis. The recommended application has been demonstrated through experiments, and the implementation results show the rationality of the recommended application design and the feasibility of practice.

Keywords: cloud computing, clustering algorithm, taxi data collection, DBSCAN algorithm

1 Introduction

With the rapid development of computer technology and the wide application of Internet technology, the amount of data in all walks of life is increasing rapidly. To analyze these massive data and transform them into easy-to-understand and useful knowledge has become an important issue facing all walks of life. At present, data mining technology has been widely used in various fields [1]. Among them, the density-based DBSCAN algorithm can mine clusters of any shape from the spatial data set containing noise and has been widely used in the field of spatial data mining. The emergence of cloud computing technology has solved the storage and calculation of massive data in data mining. With the powerful storage and computing capabilities provided by cloud computing technology, data mining technology has entered a period of rapid development based on cloud computing [2]. Taxi in the city is a manifestation of the dynamics of the city.
With the rapid development of wireless communication technology, the track record of the taxi has become convenient and fast. Most domestic taxis are already equipped with GPS terminals, which generate a large amount of trajectory data every day. How to obtain useful information for passengers and taxi drivers from the massive trajectory data has become a research hotspot. This paper proposes a trajectory clustering algorithm based on the idea of cloud computing. The traffic dispatch center will have a large amount of taxi trajectory data. To meet the real-time performance of the function, powerful computing power is required [3]. The traditional single-node server can no longer meet the demand. Therefore, this paper designs a data collection system based on cloud computing and proposes recommended application for taxi hotspots. In this paper, cloud computing technology is used through clustering algorithm and density-based DBSCAN algorithm combined with Map Reduce programming model, design trajectory clustering algorithm.

The article is thus organized in the following order. Literature reviews of various techniques and algorithms are detailed in Section 2. Section 3 discusses the system design and parallel design of trajectory clustering algorithm. The offline processing analysis of recommendation system and results is discussed in Section 4. Finally, the manuscript is concluded in Section 5.

2 Literature review

With the development of wireless network positioning, network communication, and data mining technology, the use of GPS trajectories to provide services for taxi drivers has attracted widespread attention in the industry, such as Microsoft Research Asia, Rutgers University, and Wuhan University [4]. Many new methods have been proposed to reduce the empty time of taxis, improve energy efficiency, and maximize the income of taxi drivers. Wu et al. proposed a series of studies based on GPS trajectory [5]. Geo Life is an application system based on GPS data and displays the results on an electronic map. It is not only a tool for managing personal GPS data, but also a platform for sharing GPS data and exchanging experience. T-Drive is an application system based on taxi trajectory data, which can provide services to taxi drivers and can help ease traffic and urban planning. The system extracts the taxi's passenger location and the passenger’s alighting location based on the status of the taxi [6]. Study the trajectory of experienced taxi drivers, calculate the probability that passengers can be picked up at each stop, and then recommend picking locations for taxi drivers and waiting locations for passengers. Clustering algorithm is an important part of data mining. For the parallel processing of clustering algorithm, many scholars have put forward their own ideas. Zhao et al. proposed a clustering method based on MPI (Message Passing Interface), but due to MPI using inter-process communication to coordinate parallel computing will cause the disadvantages of large memory overhead, low parallel efficiency, and intuitiveness. Wan et al. proposed an efficient parallel clustering algorithm in a high-performance cluster environment [7]. Li Jingbin et al. proposed a method to improve the clustering speed through a multicore CPU platform. Mao Jiali and others realized the parallelization of k-means algorithm based on partition based on PVM (Parallel Virtual Machine) system, but the system lacks flexibility due to the limitation of the platform. Zhou et al. proposed a clustering algorithm that processes data and tasks in parallel, but the communication overhead between each node is relatively large, and the parallelization effect is not ideal [8]. In August 2006, at the Search Engine Conference, Google CEO Eric Schmidt first proposed the concept of cloud computing. Cloud computing distributes computing tasks to clusters consisting of a large number of computers to obtain more powerful storage and computing capabilities. Cloud computing combines a variety of technologies and is developing very rapidly. Major companies at home and abroad are vying to join the field of cloud computing, such as foreign IT giants Google, Amazon, IBM, Cisco, and Microsoft [9]. In China, the development of cloud computing is still in its infancy. In terms of open source frameworks, the Apache Foundation’s Hadoop framework is widely used by enterprises and institutions to build cloud computing platforms due to its excellent features. It includes HDFS distributed storage systems, HBase, and Map Reduce programming models. At present, many large IT service providers, such as Yahoo, IMB, etc., are using the Hadoop
framework [10–15]. Figure 1 shows the Graph about the key technologies of Integration of cloud computing and moving objects in video. This paper is also based on the Hadoop framework, using HDFS to store trajectory data in a distributed manner and using Map Reduce programming ideas to parallelize the clustering algorithm [16]. This paper proposes a trajectory clustering algorithm based on the idea of cloud computing. The traffic dispatch center will have a large amount of taxi trajectory data. To meet the real-time performance of the function, powerful computing power is required. The traditional single-node server can no longer meet the demand. Therefore, this paper designs a data mining system based on cloud computing and proposes Recommended apps for taxi passenger hotspots [17–20].

Figure 1: Graph about the key technologies of integration of cloud computing and moving objects in video.

3 Research methods

3.1 System design

The steps of data mining can be summarized into three main steps, preprocessing of data, data mining, and processing after data mining. Based on this step, this paper proposes a trajectory data mining system based on a three-tier structure [21].
3.1.1 Data layer

Taxi location information is automatically collected by the GPS system equipped with the taxi. The original taxi location data are filtered, classified, and integrated and stored in the data center. Basic road traffic data provide all road information, land use information, traffic district information, and traffic facilities information [22]. Through time-space analysis of basic road traffic data, it is helpful to model the taxi movement pattern. Other data sources include passenger survey results, taxi survey results, etc. Based on this information, mining algorithms can be evaluated and related parameters can be adjusted.

3.1.2 Excavation layer

The mining layer mainly includes parallel data mining algorithms based on Map Reduce. This layer uses spatiotemporal data mining algorithms to mine the trajectory data and transform the original data into useful knowledge. Different types of analysis models can be provided according to different needs to extract, process, and analyze data. First, the raw data collected from mobile sensors are preprocessed and used in the mining stage. Some data received from taxis contain noise and data loss due to GPS unit or transmission problems. Data preprocessing must identify and remove these data to ensure the quality of the data set [23]. On the other hand, traffic forecasting models in different regions are different, and working days and rest days may have different patterns [24–28]. In order to divide the data into different data sets, it is also necessary to classify the data. Next, data mining, association analysis, cluster analysis, and data classification are needed to create different analysis modes to accurately describe the trajectory pattern of taxis.

3.1.3 Application layer

The application layer mainly uses the knowledge obtained by data mining to provide services, including direct access to data interfaces and end-user-oriented application services. Taxi trajectory data are widely used. Passenger waiting time model and taxi empty rate model are important indicators to evaluate taxi service quality, service capability, and passenger satisfaction. From the perspective of a taxi driver, how much he hopes to be able to provide him with a suggestion that his taxi will never be empty [29]. From the perspective of a taxi passenger, he hopes that when he wants to take a taxi, the system can efficiently and timely provide him with an empty taxi to his location. Considering the efficiency and profitability of taxi operations, taxi operators need to learn the relationship between taxi demand and supply and balance the benefits of these two stakeholder groups. The trajectory mining system can provide them with this help. Efficient taxi management is an important means to provide passengers with quality services and achieve corporate profit growth. The mining system can provide analysis results of various types of taxi business, which helps managers make correct decisions [29].

3.2 Parallel design of trajectory clustering algorithm

3.2.1 Algorithm preprocessing

When processing massive taxi trajectory data, the entire data area is the map of Shenzhen. This area is linear and can be divided into equal areas. However, after statistics on part of the original data, it is found that a large number of taxi pick-up points are concentrated in some areas, and most of the areas are sparsely distributed or even nonexistent. Therefore, this article uses a density-based partitioning method to ensure that too much The multi-passenger point data are distributed in a region, which prevents the data from consuming too much performance or even node operation errors when they are distributed to a data
node for processing, which affects the results of the entire clustering process and the stability of system operation. In order to facilitate data processing, when the data are specifically partitioned, the data are partitioned into a rectangular area, as shown in Figure 2. Through the analysis of the characteristics of the taxi trajectory data, the daily distribution law is similar, and there will not be much change. The specific results refer to the experimental chapter. Therefore, when dealing with massive taxi trajectory data, first analyze and process one day’s data to get a rough partition, and then partition all data. After the data are partitioned, it can provide a basis for the parallel processing of the entire algorithm, and the task of processing massive data can be assigned to a large-scale computer cluster for separate processing. After the data are partitioned, they provide a basis for the task allocation of the cloud computing platform Hadoop, which can allocate small data areas to each data node for processing.

Figure 2: Schematic diagram of data partition.

### 3.2.2 Parallel implementation of the algorithm

The realization of parallel algorithm based on Map Reduce is mainly based on the algorithm idea of parallelization to write Map function and reduce function. The Map function takes the form of key-value pairs as input parameters. This article uses the partition number of the passenger point data as the key value, and all data objects in the partition as the data value. The following is the pseudo-code implementation of the Map function (Table 1).

| Table 1: Map function |
|-----------------------|
| Mark all objects in the dataset as unvisited; |
| Initialize the cluster identifier Cluster Id; |
| for data object a in data object set |
| if object a is not visited |
| if object a the number of objects in the Eps neighborhood $\leq$ Min Pits |
| Mark object a as noise object |
| else if object a is at the partition boundary |
| Mark object a as boundary noise object |
| else |
| new set P; |
| Mark object a as a cluster object; |
| Let P store all objects in the neighborhood where the radius of object a is Eps; |
| for object point b in set P |
| Mark b as visited; |
| Recursively process all objects in object b; |
| Cluster Id ++; |
| End for |
The Reduce function merges the obtained local clustering results. The input data are also key-value pairs. The content in Table 2 is the pseudo-code implementation of the Reduce function.

Table 2: Reduce function

| For key-value pair in partial result set |
|------------------------------------------|
| if data object a is a noise object |
|   if data object a is not a boundary noise object |
|     Delete data object a; |
| else |
|     Processing boundary objects a; |
| else |
|   if cluster C is a boundary cluster |
|     Pint class cluster C; |
| else |
|     Perform boundary data processing; |
| End if |
| End for |

4 Research results

4.1 Data source

4.1.1 Data introduction

The data source used in this article is the official data from the D problem in the 2020 National College Student Mathematical Modeling Summer Camp. The data collected the trajectories of 15,000 taxis in Shenzhen from April 18, 2020 to April 26, 2021.

4.1.1.1 Data format

The data provider is Beijing Qihua Communication Co., Ltd. The data content mainly includes real-time information related to vehicles and passengers generated by the GPS of the taxi assembly [30]. The original database table includes the taxi license plate number field, the data collection time field, the longitude and latitude fields where the taxi is located, the instantaneous speed field of the taxi, the empty and loaded state field of the taxi, and the driving direction field. The meaning of each field in the data table is as follows:

name: Taxi license plate number, the unique number of the taxi in the traffic control department.
time: GPS collection time point, time information format is YYYYMMDD hh:mm:ss.
jd and wd: longitude and latitude information, output in the form of geodetic coordinate system.
status: vehicle status, 0 means nonmetering, that is, no-load state; 1 means metered, heavy-loading state.
v: The instantaneous speed of the vehicle, which requires unit conversion.
Angle: the direction the vehicle is traveling, 0 means east, 1 means southeast, 2 means south, 3 means southwest, 4 means west, 5 means northwest, 6 means north, and 7 means northeast.

4.1.2 Data preprocessing

Although the data set used in this article is relatively standard, the amount of original data is still relatively large and needs to be filtered accordingly. Attribute filtering is to filter out some attributes that are not used
in the data mining process. The taxi pick-up point is the moment when the taxi status changes from 0 to 1, so attributes that are not related to this can be filtered out, such as vehicle speed and vehicle driving direction [31]. In addition to some data that are useless for data mining, there are still some error messages in the original data set. Because taxis shuttle in the city, GPS signals are more susceptible to the influence of surrounding buildings, which will produce some incorrect data. In addition, when passing through some areas such as tunnels, there will be a short interruption of signals, taxis are always in motion, and it is normal for GPS equipment to have accuracy differences, which may lead to different forms of noise data in the data set [32].

4.2 Trajectory clustering algorithm based on map reduce

In order to verify the timeliness of the trajectory clustering algorithm based on the Map Reduce platform, this paper processes the original data and extracts 10,000, 50,000, 80,000, 150,000, and 300,000 data for experiments [33]. The experimental platform uses a stand-alone system and a Hadoop-based cloud computing system. Figure 2 shows the final running results after statistics. It can be seen from Figure 3 that before 80,000 items, the processing times of the stand-alone system and the Hadoop cloud platform system are not much different, and when the amount of data doubles again, the time required for the stand-alone system shows a geometric increase. The increase in running time of algorithms based on the Hadoop platform is still not great. Experiments show that the DBSCAN algorithm based on Map Reduce has higher timeliness than a stand-alone system and is more suitable for large-scale data set mining tasks.

![Figure 3: Comparison of DBSCAN algorithm operation under stand-alone system and cloud platform.](image)

4.3 Offline processing analysis of recommendation system

4.3.1 Analysis of passenger loading points

The taxi’s driving is restricted by time and space conditions, showing the regularity of changing with time. After preprocessing the data and extracting the passenger load points, the load points from the 18th to the 25th are counted. As shown in Figure 4. In Figure 4, the 18th to the 22nd are Monday to Friday, which are normal working days, and the 23rd and 24th are Saturdays and Sundays, which are the rest days. Figure 5
selects the 19th and 20th, 23rd and 24th to represent working days and rest days, respectively, to make statistics on the passenger load in each time period.

It can be seen from the passenger load statistics for each time period in the four days in Figure 4 that the areas with the smallest number of passengers in the four days are distributed between 0:00 and 6:00. During this period, most residents are resting at night. From the trend of the four curves, it can be seen that the three peak hours are 7:00–12:00, 12:00–16:00, and 18:00–22:00, which are in the three periods with the largest crowd flow, such as commuting to get off work, school, and so on. It can also be seen from the figure that the 19th and 20th working days are almost the same as the 23rd and 24th rest days. The number of passengers on working days is generally higher than that on rest days of the low valley and peak area.

Through the analysis of the taxi passenger situation, according to the travel pattern of the crowd, the time of day is divided into different characteristic time periods, specifically 0:00–7:00, 7:00–12:00, and 12:00–16. In the five time periods of 00, 16:00–20:00, and 20:00–24:00, the next hotspot clustering of taxi passengers will be carried out according to the above characteristic time periods.
4.3.2 Clustering parameters

For DBSCAN algorithm, the difference of Eps and Min Pts parameter settings will cause the difference of clustering results. The judgment of the core object is based on Eps and Min Pts. Two parameters determine whether the current object is a core object or a noise object and determine the final clustering result. Offline processing mainly explores the hotspot area of taxi passengers, that is, within the radius of Eps, the number of taxi passengers exceeds Min Pts Area. After experimenting with one day’s data, this paper defines the taxi passenger hotspot area in a circular area with a radius of 400M, and the area where the number of taxi passenger points exceeds 12.

4.3.3 Clustering results

Based on Hadoop’s DBSCAN algorithm, with parameters Eps = 400, Min Pts = 12, cluster analysis of daily data is performed. Figure 5 shows the number of passenger hotspots from the 18th to the 25th. It can be seen from the Figure 6 that the daily passenger load points are between 200 and 250. Therefore, in the absence of holidays, the number of passenger hotspots tends to be stable. It is reliable to perform cluster analysis.

![Figure 6: Statistics of passenger hot spots from 18th to 25th.](image)

5 Conclusion

This article introduces the format and preprocessing of the data source, verifies the timeliness of the trajectory clustering algorithm based on the cloud platform, and conducts statistical analysis based on the 8-day data of 13,798 taxis in Shenzhen and determines the characteristic time. By clustering the passenger load points in each time period to get the passenger hot spots, map matching with Baidu map, combined with the recommendation degree to display, it verifies the feasibility of the passenger load point recommendation application based on trajectory clustering. The trajectory clustering algorithm needs to be improved. In this article, when using the DBSCAN algorithm for clustering, it is still necessary to enter the corresponding Eps and Min Pts parameters, but the difference of the parameters will lead to differences in the clustering results. There are still shortcomings in the research on parameter selection, and it will be necessary in the future. We also summarize the problems and the challenges existed in moving object data mining, which mainly includes the following aspects:
• Most of the current trajectory clustering algorithms cannot fully combine time and space dimensions, and they just regard time as the additional dimension of trajectory object.
• When clustering results are converted to knowledge, some problems lead to a considerable part of them being either too complex to be intuitively understandable or too simple, close to common sense, which deviates from the target results.
• When the trajectory data used in trajectory clustering are too much, they often lead to the efficiency of trajectory clustering algorithms being low.
• The general applicability of trajectory clustering algorithm is low.
• Most trajectory clustering algorithm cannot take the overall features and local features of trajectories into an overall consideration.

Conflict of interest: Authors state no conflict of interest.

References

[1] Wan J, Zhang D, Sun Y, Lin K, Zou C, Cai H. VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing. Mobile Netw Appl. 2014;19(2):153–60.
[2] Sajid A, Abbas H, Saleem K. Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges. IEEE Access. 2016;4:1375–84.
[3] Lopez J, Rubio JE. Access control for cyber-physical systems interconnected to the cloud. Comput Netw. 2018 Apr;134:46–54.
[4] Gravina R, Ma C, Pace P, Aloi G, Russo W, Li W, et al. Cloud-based activity-aaservice cyber-physical framework for human activity monitoring in mobility. Future Gener Comput Syst. 2017 Oct;75:158–71.
[5] Wu D, Rosen DW, Wang L, Schaefer D. Cloud-based design and manufacturing: a new paradigm in digital manufacturing and design innovation. Comput Aided Des. 2015 Feb;59:1–14.
[6] Wan J, Zhang D, Zhao S, Yang LT, Lloret J. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. IEEE Commun Mag. 2014 Aug;52(8):106–13.
[7] Wan J, Zhang D, Sun Y, Lin K, Zou C, Cai H. VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing. Mobile Netw Appl. 2014;19(2):153–60.
[8] Elshawi R, Sakr S, Talia D, Trufnio P. Big data systems meet machine learning challenges: towards big data science as a service. Big Data Res. 2018;14:1–11.
[9] Agarwal D, Cheah YW, Fay D, Fay J, Guo D, Hey T, et al. Data-intensive science: the Terapixel and MODIS/Azure projects. Int J High Perform Comput Appl. 2011;25(3):304–16.
[10] Evangelinos C, Hill C. Cloud computing for parallel scientific hpc applications: feasibility of running coupled atmosphere-ocean climate models on amazons ec2. Ratio. 2008;2(2.40):2–34.
[11] Santos LW. Data warehouse with big data technology for higher education. Proc Comput Sci. 2017;124:93–9.
[12] Kwon T, Yoo WG, Lee W-J, Kim W, Kim D-W. Next generation sequencing data analysis on cloud computing. Genes Genomics. 2015;37(6):489–501.
[13] Mohammed EA, Far BH, Naugler C. Applications of the mapreduce programming framework to clinical big data analysis: current landscape and future trends. BioData Mining. 2014;7(1):22.
[14] Gonzalez NM, de Brito Carvalho TCM, Miers CC. Cloud resource management: towards efficient execution of large-scale scientific applications and workflows on complex infrastructures. J Cloud Comput. 2017;6(1):13.
[15] Li X, Song J, Huang B. A scientific workflow management system architecture and its scheduling based on cloud service platform for manufacturing big data analytics. Int J Adv Manuf Technol. 2016;84(1–4):119–31.
[16] Afgan E, Baker D, Batut B, Van Den Beek M, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46(W1):W537–44.
[17] Cardone G, Corradi A, Foschini L, Ianniello R. Participact: a large-scale crowdsensing platform. IEEE Trans Emerg Topics Comput. 2016 Jan/Mar;4(1):21–32.
[18] Tokosi TO, Scholtz BM. A classification framework of mobile health crowdsensing research: a scoping review. Proc South Afr Inst Comput Sci Inf Technol. 2019;2019:1–12.
[19] Radhakrishnan S, Pan R, Vahdat A, Varghese G. Netshare and stochastic netshare: predictable bandwidth allocation for data centers. ACM SIGCOMM Comput Commun Rev. 2012;42(3):5–11.
[20] Yu W, Liang F, He X, Hatcher WG, Lu C, Lin J, et al. A survey on the edge computing for the internet of things. IEEE Access. 2017;6:6900–19.
[21] Gautam P, Ansari MD, Sharma SK. Enhanced security for electronic health care information using obfuscation and RSA algorithm in cloud computing. Int J Inf Secur Priv. 2019;13(1):59–69.

[22] Zahid F, Gran EG, Bogdański B, Johnsen BD, Skeie T. Efficient network isolation and load balancing in multi-tenant HPC clusters. Future Gener Comput Syst. 2017 Jul;72:145–62.

[23] Kirschnick J, Calero JMA, Wilcock L, Edwards N. Toward an architecture for the automated provisioning of cloud services. IEEE Commun Mag. 2010 Dec;48(12):124–31.

[24] Liu X, Chen X. Tracking technique for active vision based on extended Kalman filter. CAE. 2007;2:10–8.

[25] Wang W, Sun Q, Zhao X, Yang F. An improved particle swarm optimization algorithm for QoS-aware web service selection in service oriented communication. Int J Comput Intell Syst. 2010;3(1):18–30.

[26] Sotomayor B, Montero RS, Llorente IM, Foster I. Virtual infrastructure management in private and hybrid clouds. IEEE Internet Comput. 2009 Oct;13(5):14–22.

[27] Milani AS, Navimipour NJ. Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends. J Netw Comput Appl. 2016;71:86–98.

[28] Fu X, Zhou C. Virtual machine selection and placement for dynamic consolidation in cloud computing environment. Front Comput Sci. 2015;9(2):322–30.

[29] Kokilavani T, Amalarethinam DG. Load balanced min-min algorithm for static meta-task scheduling in grid computing. Int J Comput Appl. 2011;20(3):42–8.

[30] Zhang G, Yang Z, Xie H, Liu W. A secure authorized deduplication scheme for cloud data based on blockchain. Inf Process Manag. 2021;58(3):102510.

[31] Babu LDD, Krishna PV. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl Soft Comput. 2013 May;13(5):2292–303.

[32] Ragmani A, El Omri A, Abghour N, Moussaid K, Rida M. A performed load balancing algorithm for public cloud computing using ant colony optimization. Recent Pat Comput Sci. 2018;11(3):179–95.

[33] Paya A, Marinescu DC. Energy-aware load balancing and application scaling for the cloud ecosystem. IEEE Trans Cloud Comput. 2017 Mar;5(1):15–27.