Fast Path Clustering Algorithm Based on Symmetric Neighborhood

Huifang Deng, Xinyan Tang, Caifeng Zou, and Chunhui Deng

***Abstract*—**In this paper, the DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm is analyzed and its limitation is pointed out when it is used in path detection of moving object. Then the FPCSN (the Fast Path Clustering Algorithm based on Symmetric Neighborhood) algorithm is proposed. The concept of density factor is introduced in this algorithm. By searching for the k neighborhood and the reverse k neighborhood of the sub-divided path segments, the density factor is computed, then the sub-divided path segments are clustered, and the query operation of neighborhood is optimized. This algorithm can identify small and dense clusters from large and sparse clusters. It also reduces the query times and improves efficiency.

***Keywords*—**DBSCAN (Density Based Spatial Clustering of Applications with Noise), density factor, FPCSN (the Fast Path Clustering Algorithm based on Symmetric Neighborhood), k neighborhood, moving object, path segments, sub-path.

I. Introduction

CLUSTERING is the process of grouping physical or abstract objects into classes which consist of several similar objects. The objects are called cluster. In the same cluster, the objects have high similarity [1]. In the different clusters, the objects have large difference.

In recent years, with increasing research interest in the moving objects, the clustering of paths becomes an important research subject. The clustering is widely used in traffic management, military management, mobile computing, meteorology and disaster science and so on. For example, in transport system [2], the clustering can apply to paths analysis which can lead to the following applications:

1) Optimization of traffic management. By the clustering analysis of paths which are produced by moving vehicles, the transport network is divided into different sub-regions. Then by macroscopic management and control of the traffic flow, the road network can be optimized and the traffic congestion can be eased.
2) Road construction. Construction of roads is normally with reference to vehicle's travel patterns. Transport network includes the main roads and secondary roads. By the clustering of paths, core sub-paths can be obtained. Core sub-paths represent the paths which vehicles often travel. So core sub-paths can be regarded as the main roads.
3) Traffic navigation. Different segments of roads have different traffic flow. So through the real-time analysis of paths which is generated by vehicles, the traffic flow can be obtained. The user can select the road segment which has relatively less traffic flow which can navigate effectively.

The organization of this paper is as follows: The limitation of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [3] is addressed in Section II. The fast path clustering algorithm based on symmetric neighborhood (FPCSN) is proposed in Section III. The experimental hardware and software environment is given in Section IV along with the ways of preprocessing the moving object path data. The simulation results of FPCSN are presented in Section V. And the last section is the summary.

II. Existing Problems

Density-based clustering method can find clusters of arbitrary shape in general if a proper pre-processing is done. So it can be applied to paths detection of moving objects. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [3] is the most classical algorithm in density-based clustering methods. DBSCAN can find cluster in the data space with noise. But when it is used in moving object paths clustering, it has the following limitations:

1) DBSCAN uses two global parameters which are called ε and MinPts. This method selects parameter according to the most scattered clusters. When the clusters are dense and the neighborhood contain MinPts points, ε will become too large and the adjacent regions of different densities will merge into one cluster. Therefore, DBSCAN can not identify the small, dense regions from the large and sparse regions.
2) Because moving objects have time and space properties [4], the function of the distance between the sub-paths does not meet the triangle inequality and traditional spatial index structures cannot be directly used to support sub-path neighborhood queries. So when DBSCAN algorithm

Huifang Deng is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (corresponding author's phone: +86 18903001886; e-mail: hdeng2008@gmail.com).

Xinyan Tang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (e-mail: xytang2008@gmail.com).

Caifeng Zou is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (e-mail: caifengzou@gmail.com).

Chunhui Deng is with the Department of Computer Engineering, Guangzhou College, South China University of Technology, Guangzhou, China (e-mail: jintiand@126.com).

applies to the paths clustering of moving object, time complexity is $O(n^2)$ where n is the total number of divided paths (segments).

III. The Fast Path Clustering Algorithm based on Symmetric Neighborhood

Zheng [5] proposed outlier detection algorithm based on symmetric neighborhood. This algorithm introduces density factor and isolated factor based on symmetric neighborhood. By computing density factor of each point, the clustering points are removed. Then the isolated factor of each isolated points is computed. In order to improve DBSCAN algorithm, the density factor based on symmetric Neighborhood is introduced and the Fast Path Clustering Algorithm based on Symmetric Neighborhood (FPCSN) is proposed which can apply into path clustering of moving objects.

A. Related Concepts of FPCSN

Definition 3.1: Sub-path. If a random path in the space is divided into several sections, each section is called sub-path.

Let T be the set of sub-paths, T={L1,L2, ⋯ , Ln}, sub-path Li($1 \le i \le n$) and Lj($1 \le j \le n$) are two random paths in the set of sub-paths. We use similarity as the measurement of Li and Lj which is denoted as Si,j.

Definition 3.2: k neighborhood of sub-path. If L is a sub-path, k neighborhood of L can be denoted as kNB(L) in brief. kNB(L) is the set of sub-paths which is most similar to L.

Definition 3.3: k neighborhood radius r of sub-path. For all the sub-paths which are included in kNB(L), the distance between the sub-paths which have the smallest similarity is called k neighborhood radius r.

Definition 3.4: Reverse k neighborhood of sub-path. If L is a sub-path, the reverse k neighborhood of L can be denoted as RkNB(L) in brief. If kNB(L') contains L, the set of L' is RkNB(L).

Definition 3.5: Density factor of sub-path. If L is a sub-path, the density factor of L can be denoted as LDF(L) in brief. LDF(L)=|RkNB(L)| / |kNB(L)| where |RkNB(L)| denotes the magnitude of RkNB(L) and |kNB(L)| denotes the magnitude of kNB(L).

Definition 3.6 Core sub-path. If Li($1 \le i \le n$) is a sub-path and LDF(Li) is at least no less than the threshold minLdf where minLdf⩾1, Li is called core sub-path.

B. The Process of FPCSN

The basic idea of the algorithm can be described as follows: Selecting the first unlabelled sub-path, computing density factor of the sub-path and then determining if this sub-path is core sub-path. If this sub-path is core sub-path, merging its k neighborhood with the overlapping paths according to the merging principles of the greatest connected sub-graph.

The algorithm flow chart is shown as Figure 1.

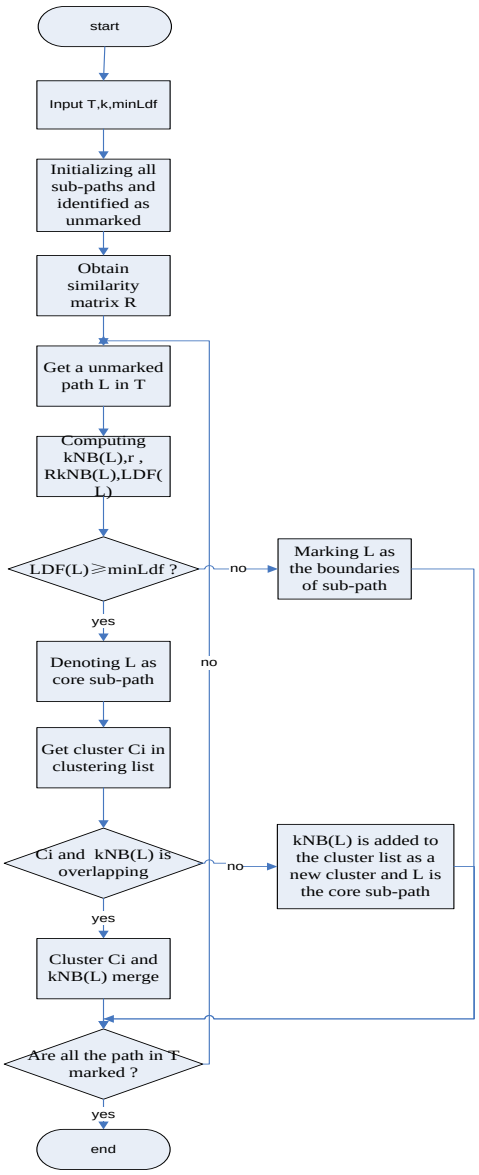

Fig. 1 The flow chart of FPCSN

The FPCSN algorithm is implemented as follows:

Input: The set of sub-paths which is called T, parameter k , parameter minLdf
Output: The cluster Ci of sub-paths
Begin
1: Initializing all sub-paths and identified as unmarked
2: Calculating the similarity between all sub-paths and forming a similarity matrix R
3: for(each sub-path L which is denoted as unmarked in T)
4: Computing kNB(L), r and RkNB(L)
5: Computing LDF(L)=|RkNB(L)| / |kNB(L)|
6: if (LDF(L)⩾minLdf)
7: Denoting L as core sub-path
8: for(each cluster Ci in clustering list)
9: if(Ci and kNB(L) is overlapping)//If the distance between L and core sub-path of Ci no more than 2r, Ci and kNB(L) is overlapping

10：Cluster Ci and kNB(L) merge;//merging the sub-paths in cluster Ci and kNB(L), then the result is saved in newCi . L and the core sub-paths of Ci are added to the corresponding core sub-path of newCi, then newCi instead of the current cluster.

11：else

12：kNB(L) is added to the cluster list as a new cluster and L is the core sub-path.

13：else

14：Marking L as the boundaries of sub-path.

15：end

The advantages of FPCSN can be described as followed:

1) Using the longest common sub-sequence as the measure of similarity between two sub-paths which can avoid the differences between time, space, speed and length of the path which is caused by Euclidean distance.
2) FPCSN can identify the small, dense regions from the large, sparse regions. FPCSN finds cluster according to the following procedure: FPCSN firstly determines a core sub-path based on neighborhood which is in orbital cluster, and then orbital cluster expansion. For the sub-path L, FPCSN determines whether L is the core sub-path based on neighborhood according to whether there is "LDF (L) ≥ minLdf" and this process reflects local feature.
3) Merging kNB(L) and the overlapping sub-paths can reduce the time of neighborhood queries.

IV. Simulation Setting and Data Pre-processing

All experiments are conducted under following configurations: CPU: Intel(R) Core(TM) i5 3.2G; Memory：4G; Operating System: Windows 7; Development Tools: Microsoft Visual Studio 2010 and Microsoft SQL Server 2008.

Pre-processing is applied to all experimental data, i.e., path data of moving objects. Three ways are used to divide or sub-divide the path of moving object: (1) by position, (2) by time, and (3) by pattern cycle, The x and y co-coordinators extracted from the traffic path data generated by the simulator of Thomas Brinkhoff are used as the initial path data set in the experiment, while in the real situation, the temporal property of data set is ignored and only the spatial property is considered.

V. Emulation of FPCSN

In this paper, we use the paths produced by Thomas Brinkhoff which is a traffic generator. The path information includes the location information of vehicles at a specific period of time. We extract the x and y coordinates as the original data set.

A. Quality Assessment

Here, we use DB index to describe the quality of clustering. DB index [6][7] is the comprehensive evaluation index for clustering quality. By selecting six groups of data sets and executing DBSCAN and FPCSN, DB index is computed and shown in Figure 2.

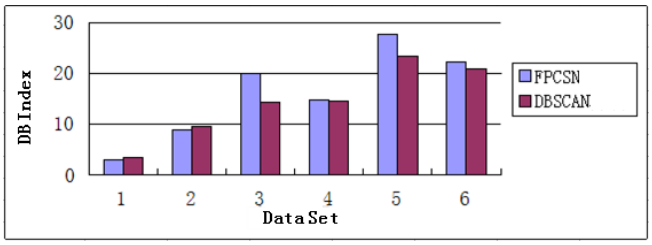

Fig. 2 The comparison of DB index

From Figure 2, we can see that for data set 3, 4, 5 and 6, the DB index for FPCSN is larger than for DBSCAN. So FPCSN has better clustering results. This is because these four data sets are the ones of multi-level (i.e., non-uniformly distributed) density. But for data sets 1 and 2, the DB index for FPCSN is smaller than for DBSCAN, this implies that DBSCAN is more suitable for the uniformly distributed data sets. While in reality majority of data sets are non-uniformly distributed, i.e., of multi-level density (Figure 3).

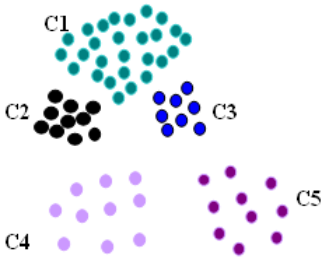

Fig. 3 An example of a data set of multi-level density

B. The Influence of Execution Time by Parameter k

When the number of paths is constant, the influence of execution time by parameter k can be shown as Figure 4.

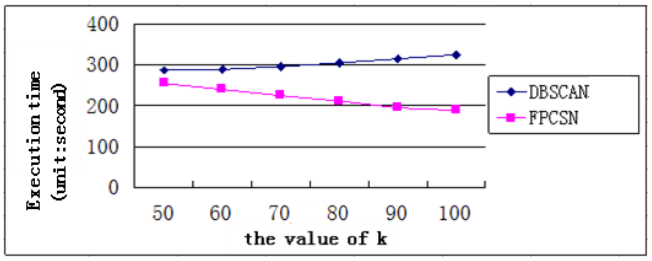

Fig. 4 The influence of execution time by parameter k

It can be seen from the figure, with the increase of k, the execution time of DBSCAN increases, while the execution time of FPCSN algorithm decreases. That is because if the k increases, the k neighborhood of sub-paths for DBSCAN increases and the execution time increases. For FPCSN, the k neighborhood of sub-paths also increases, but there is a process of merger. So increase in k means more merger of sub-paths which in return reduces the operation time and the execution time.

C. The Influence of Execution Time by the Size of Paths

Figure 5 shows the influence of execution time by the size of paths when the parameters are kept constant. After reading the sub-paths from the database, we can obtain the number of sub-paths are 5000, 8000, 11000, 14000, 17000, 20000. Then algorithms of DBSCAN and FPCSN are executed and the execution time can be obtained.

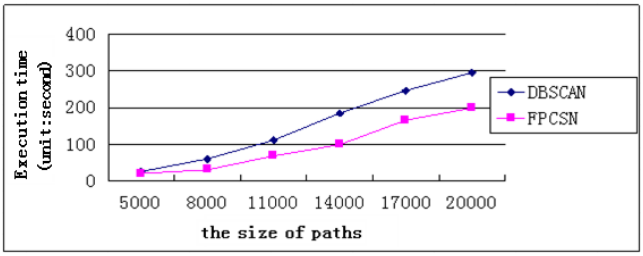

Fig. 5 The influence of execution time by the size of paths

It can be seen that with the increase of the size of sub-paths, the execution time of DBSCAN and FPCSN increases, but the increase for FPCSN is slower than that for DBSCAN. It means that FPCSN outperforms DBSCAN in efficiency.

VI. Conclusion

Cluster analysis of the path with moving objects is a hot topic. In this paper, the concept of the density factor is introduced and the query of the neighborhood is optimized which leads to the rapid clustering with different density data sets. Then FPCSN algorithm is proposed. The experimental results indicates that the FPCSN algorithm is able to identify small and dense clusters from large and sparse clusters of the paths of moving objects and shows a good quality in terms of DB index. Also the performance of FPCSN algorithm is improved to some extent.

At present, in this paper, we only considered the spatial property in analyzing the path clustering of moving objects, while ignored the time feature which we will take into account in the future work. Because the data for paths of moving objects is huge, we need to find a more efficient method for neighborhood search in the future as most of the computing time in FPCSN spends on the k neighborhood search.

References

[1] J. Han, M. Kamber, Data Mining:Concepts and Techniques, Beijing: China Machine Press, 2007.

[2] S. Niu, Research on Urban Traffic Incident Detection Based on Floating Car, Beijing: Beijing Jiaotong University, 2008.

[3] M. Ester, H. P. Kriegel, J. Sander, et al., "A density-based algorithm for discovering clusters in large spatial databases," in Proc. of the 2nd Int'l Conference on Knowledge Discovery and Data Mining, Portland, AAAI Press, 1996, vol. C, pp. 226-231.

[4] J. R. Hwang, H. Y. Kang, and K. J. Li, "Spatio-Temporal Similarity Analysis Between Trajectories on Road Networks," Perspectives in Conceptual Modeling, ER Workshops, Springer Berlin Heidelberg, Lecture Notes in Computer Science, vol. 3770, pp. 280-289, 2005.

[5] J. Zheng, Research and Implementation of Clustering and Outlier Detection Algorithms, Nanjing: Nanjing University of Aeronautics and Astronautics, 2007.

[6] D. L. Davis, and D. W. Bouldin, "A Cluster separation measure," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 4, pp. 224-227, 2009.

[7] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Clustering Validity checking methods," Part Ⅱ, ACM SIGMOD Record Archive, vol. 31, no. 3, pp. 19-27, 2002.