International Journal of Geo-Information

MDPI

Article

GroupSeeker: An Applicable Framework for Travel Companion Discovery from Vast Trajectory Data

Ruihong Yao, Fei Wang *, Shuhui Chen and Shuang Zhao

College of Computer Science, National University of Defense Technology, Changsha 410073, China; yaoruihong17@nudt.edu.cn (R.Y.); shchen@nudt.edu.cn (S.C.); zhaoshuang16@nudt.edu.cn (S.Z.)
* Correspondence: wangfei09a@nudt.edu.cn

Received: 5 May 2020; Accepted: 13 June 2020; Published: 20 June 2020

Abstract: The popularity of mobile locate-enabled devices and Location Based Service (LBS) generates massive spatio-temporal data every day. Due to the close relationship between behavior patterns and movement trajectory, trajectory data mining has been applied in numerous fields to find the behavior pattern. Among them, discovering traveling companions is one of the most fundamental techniques in these areas. This paper proposes a flexible framework named GroupSeeker for discovering traveling companions in vast real-world trajectory data. In the real-world data resource, it is significant to avoid the companion candidate omitting problem happening in the time-snapshot-slicing-based method. These methods do not work well with the sparse real-world data, which is caused by the equipment sampling failure or manual intervention. In this paper, a 5-stage framework including Data Preprocessing, Spatio-temporal Clustering, Candidate Voting, Pseudo-companion Filtering, and Group Merging is proposed to discover traveling companions. The framework even works well when there is a long time span during several days. The experiments result on two real-world data sources which offer massive amount of data subsets with different scale and different sampling frequencies show the effective and robustness of this framework. Besides, the proposed framework has a higher-efficiency performing when discovering satisfying companions over a long-term period.

Keywords: traveling companion discovery; spatio-temporal trajectory mining; framework; association analysis; clusteirng; parameter-setting strategy

1. Introduction

According to the statistics of China's 2019 telecommunications business, the number of mobile phone users reached 1.6 billion by the end of 2019 [1]. Due to the development of location techniques and widespread use of smart devices, personal trajectory data has become an important resource for understanding personal or group behaviors, and trajectory data mining has become a hot topic in many of research fields [2]. For instance, Elragal et al. [3] and Shingo Enami et al. [4] used relative technologies in vehicle management. Tian Qin et al. [5] proposed a method to mine spatio-temporal routine of people based on mobile phone data. Huan et al. [6] tried to explore social behaviors on mobile sensors data. Chen et al. [7] made disease predictions based on mobile big data. Xudong Liu et al. [8] used the taxi trajectory data to identify urban functional regions in Chengdu. Besides, trajectory data analysis has applied in some practical applications, such as nearby friend recommendation based on location-based service (LBS) [9] and route navigation in Map Applications, etc.

Discovering accompanying or group behavior pattern is an important branch in mining mobile trajectory data. The pattern is defined as more than one moving objects that travel together for a period of time. Such pattern discovery provides significant supports to a large amount of relative fields, such as control of key personnel, tourism development, accident investigation, group tracking etc.

It has been applied in significant application scenarios. Tang et al. [10] proposed a loose companion discovery for military object monitoring to describe the several members may temporarily leave the group and go back in short time. Meiling Zhu et al. [11] proposed a novel algorithm to find Platoon companion pattern over a special type of spatio-temporal data stream. Zhu et al. used Hainan tourists data to find group movement pattern and classified tourists [12], etc. Thus, mining and analyzing accompanying behavior pattern are necessary for relative applications and academic fields.

Since mobile devices can generate massive amounts of data, one big challenge is brought into accompanying pattern mining, i.e., high performance of algorithms are needed to process massive data in limited time. Another major challenge comes from the optimization of the traveling companion discovering algorithm. Traveling Companion Discovering Algorithm comes from the Clustering-and-Intersection method [13], which defines the companion candidates to describe the similar companions in each time snapshot. Tang et al. [14] optimized the Clustering-and-Intersection algorithm into a smart-and-closed algorithm by combining the buddy structure to improve the effectiveness of the method. In the mean time. Some studies [12,15] use the similar way to discover Traveling Companions or other behavior pattern. However, it is easy to cause an omitting candidate problem with the time-snapshot-slicing-based method, especially when the time period is extremely short-term. Due to the sparsity of the mobile trajectory data, it is a hazard to cluster these trajectory data using the unbefitting time segmentation method. Concretely, some cluster-able trajectory data cannot be clustered possibly and are even filtered as noise. Therefore the approaches based on time-segmented slicing may not always be completely successful.

In this paper, we propose a new companion discovery method based on the clustering algorithm and association analysis algorithm to solve the above problems. In contrast to the time-snapshot-slicing-based methods or models, this method finds the closeness in the location and the closeness in time reflected in the moving-user data from a holistic perspective. In addition, more focus is given to the potential correlation between users. For example, if A and B are a pair of accompanying partners, they are more likely to spend time together in a small region, which can be defined as that B appears when A appears or A appears when B appears.

The proposed algorithm is an extension and optimization of our previous work [16]. On this basis, we improve the algorithm and propose a *5-stage* framework. Firstly, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HBDSCAN) [17] is used to mine similar moving users in a certain geographic area and within the time span. Then, a classical association analysis algorithm Frequent Pattern (FP-growth) is used to predict the internal association relationships among similar users, which takes full use of the characteristics of clustered data with high similarity to find potential accompanying patterns. The following stage involves a filtering strategy which is used to perform the necessary filtering to obtain the recommended travel companions for certain pseudo-companion scenarios. The last stage is designed to merge the results data into groups.

The main contributions proposed in this paper are as follows:

- A framework of traveling companion discovery named GroupSeeker is proposed. Through a five-stage processing flow, GroupSeeker can find potential traveling companions in a huge amount of trajectory data with high performance and accuracy.
- Parameter Setting Strategies are inherently embedded into GroupSeeker. Primary stages can determine their parameters according to the characteristic of datasets, which makes the framework much more practical and applicable.
- A novel Spatio-temporal clustering method is used to deal with trajectory data of long-term time slices and solve the omitting problem of companion candidates caused by improper short-term time segmentation in previous work.
- Experimental results on real-world and simulated datasets show the time cost of GroupSeeker is at a desirable level. Trajectory data for twenty-four hours can be processed within one and a half hours, which means GroupSeeker can be used in all-weather monitoring jobs.

The reminder of the paper is organized as follows. Section 2 introduces the related work; Section 3 gives the problem definition and the methodology, including the framework and methods; Section 4 presents the experimental results; Section 5 concludes this paper and gives some discussion about future work.

2. Related Work

In this section, the related work based on two main categories is introduced, i.e., the trajectory clustering and the companion pattern discovery.

2.1. Trajectory Clustering

For the clustering of similar trajectories based on the time dimension, Agrawal et al. proposed the trajectory similarity measurement based on Euclidean distance [18] in 1993. Faloutsos C et al. and Chan KP et al. used discrete Fourier transform and discrete wavelet transform respectively to preprocess the trajectory similarity measurement based on Euclidean distance [19,20]. Elnekave S et al. improved the expression of MBR by proposing MBB (Minimum Boundary Box) smooth trajectory to deal with the influence of noise better [21].

For clustering based on the similarity of trajectories, the similarity mining of entire trajectory features is focused on reducing the requirement in the time dimension, only requiring the chronological order among the trajectory-recording points, thus, general DTWD is used to deal with such clustering [22]. For local clustering with a single trajectory, Lee et al. presented a framework that divides first and then aggregates to divide into sub-trajectories according to the principle of minimum-description length, using the density clustering method [23]. In addition, several density-based clustering methods were proposed, such as DBSCAN, DENCLUE, OPTICS, etc. DBSCAN [24] is a widely used spatial location clustering algorithm. It has the characteristics of not needing to determine the number of clusters in advance and can find clusters of arbitrary shapes [25]. In 1999, OPTICS was proposed by Ankerst M. et al. Instead of producing clusters of a data set explicitly; however, it creates an augmented ordering of the databaset representing its density-based clustering structure [26]. Hinneburg, A and Gabriel, HH proposed DENCLUE 2.0 to improve the disadvantage of DENCLUE 1.0 [27] that making small steps at first could lead to never converges to the maximum [28]. In 2017, Mclnnes et al. proposed a hierarchical density-based clustering algorithm and released a related codebase as a package in Python to use [17]. Yuqing Yang et al. proposed a trajectory clustering algorithm to extract trajectory Stays based on the density analysis in spatial-temporal trajectory data and achieved higher clustering accuracy in the real-world data sets [29].

For clustering of the trajectory points, Gao Y et al. proposed a constrained k-nearest neighbor queries among trajectories [30]. A Subtrajectory Clustering algorithm based on the Fréchet Distance using GPU was proposed by Gudmundsson J et al. [31] to take advantage of continuous Fréchet Distance as the measurement of similarity among trajectory curves which has obvious performance advantages. Similarly, Deng Z et al. proposed a modified OPTICS algorithm, called Tra-OPTICS, to cluster trajectory. Besides, a GPU-based version is proposed to optimize performance, called G-Tra-OPTICS, which is based on the STR-tree as the indexing structure [32]. Yuan, G et al. summarized these important techniques of trajectory clustering [33].

For semantic trajectory clustering, Xiao X et al. proposed a method for finding similar users using category-based history [34]. Ying JC et al. proposed semantic trajectory clustering based on the location prediction to recommend the user to the next dimension [35]. Liu S et al. presented an approach to achieve recognition of hot spots among trajectories [36]. Andrienko et al. presented generic techniques and visualization guidelines to support movement data analysis, using the trajectory clustering on a real air traffic data-set [37]. Olive, X., and Morio, J. applied the trajectory clustering in the air traffic management and validated the effectiveness of the proposed method on a real-world trajectory set [38].

2.2. Companion Pattern Discovery

Through analyzing the behavioral patterns of mobile-object groups, accident investigation and group tracking based on the spatio-temporal environment can be realized. In a series of research outputs, representative trajectory patterns were defined, mainly including flock [39], convoy [40], swarms [41] and gathering [42]. In 2016, Zhenzhen Wang et al. presented a literature review to summarize the existing travel behavior studies that applied mobile phone data and have discussed the potential of mobile phone data in advancing travel behavior research [43].

Gudmundsson et al. [39] defined the flocking model which optimizes the early research population patterns by predefining the regional shape and population size. The convoy model defined by Jeung et al. [40] realized trajectory mining with arbitrary shapes based on density clustering, avoiding predefined spatial thresholds, and the model requires a certain number of moving objects to be connected in density over k durations. Further optimization based on the first two models was the Swarm model defined by Li et al. [41]. In their method, the time is not required to be continuous when the moving objects move together for a certain period of time. Zheng et al. [42] defined gathering pattern that simulates group events in trajectories, such as celebrations, parades, protests, etc. In addition, effective index structures and fast patterns based on bit vectors are proposed to improve mining efficiency. Fan Chen et al. proposed a method for detecting group interactions for groups of varying numbers of objects [44]. Zhang et al. [45] used the spatio-temporal graph to retrieve gathering. The researchers presented the CUTis [46] (Clustering Trajectory data stream), which is a processing algorithm for an incremental trajectory data stream. A method for identifying the group movement pattern through mobile phone call detail records (CDRs) based on similarity to discover tourist groups was proposed by Zhu et al. [12]. An algorithm for finding gradual moving objects clusters pattern among trajectory streams was proposed by Yujie Zhang et al. [15]. In order to discover accompanying vehicles, in intelligent transportation system (ITS), a typical application in software engineering technology, Meiling Zhu et al. [47] proposed a method for discovering Traveling Companions through Automatic Number Plate Recognition (ANPR) data stream, using frequent sequence mining with time constraints. Zhang et al. [15] used the sliding window to mining the cluster pattern in trajectory data.

Moreover, the correlation analysis algorithm is used in the trajectory analysis and pattern discovery. Xia Dawen et al. [48] proposed a method using a parallel frequent pattern growth algorithm based on map-reduce to analyze trajectory big data. Hu et al. [49] used OPTICS clustering and association. Based on frequent item-set, Al-badwi et al. [50] proposed a breadth-first and depth-first hybrid distributed approach with Frequent itemset mining (HD-FIM) on Spark to increase the efficiency of discovering companion vehicles.

Regarding to the methods of discovering traveling companions, Puntheeranurak et al. [46] proposed a micro-group-based clustering algorithm to reduce the computational cost and they conducted experiments on a real taxi trajectory data and synetic data. Nevertheless, their research is difficult to avoid Companion Candidate Omitting Problem and the scale of their testing samples are smaller than our work. Besides, Xinning Zhu et al. [12] proposed a threshold-based method and safe semi-supervised support vector machines (S4VMs) to calculate the similarity vectors of tourists and detect their transportation mode for finding the group movement pattern through CDRs. However, this research and the proposed framework are mainly used in special applications such as tourism. Thus, the motivation of their work is actually different from our research. In contrast, our research is closer to the study of the underlying framework in the filed about discovering traveling companion.

3. Materials and Methods

In this section, the problems are illustrated to describe the situation for our methods and problem definitions are presented to facilitate subsequent descriptions. Finally, a framework is proposed, including five stages to discover traveling companion.

3.1. Problem Statement

3.1.1. Companion Candidate Omitting Problem

Traveling Companion is a set of moving objects that move together as a group for a period of time. In terms of spatio-temporal trajectory data, traveling companions are formalized as a set of moving objects, whose spatial positions are density-connected in a cluster within a short-term time span. Previous studies divided continuous time into time snapshots in order to discover traveling companions from spatio-temporal trajectory data and checked each time snapshot for candidate partners. However, since real-time spatio-temporal trajectory data is not always uniformly sampled in the time dimension or the geographic dimension, such time-division operations may lead to the problem of omitting candidates. We will describe the above issues in detail, and give the definitions used in the following work.

After preprocessing the real-world non-intensively sampled trajectory data, two data characteristics are found:

- Signals of real-world positioning data may be blocked during acquisition and transmission. The reason for blocking is because users can actively turn off devices or terminate location service and the transmission of location information may be interfered or blocked by surrounding environments.
- Due to differences in sampling methods and loss of data transmission, trajectory data will be sparse or partially lost during data collection.

Because of the above characteristics, when the conventional accompanying-pattern discovery algorithm uses a time-segment slicing method in a highly sparse trajectory data set, the recording points at the edges of the formed clusters are likely to be filtered as noises due to the inappropriate duration of time slices.

Figure 1 shows a companion candidate omitting examples. There are adjacent time snapshots, i.e., *s1*, *s2* and *s3*. One or more clusters in each snapshot can be obtained after cluster processing, along with several unclustered points such as *A*, *B* and *C*. Because the time segmentation happens to be in the middle of their sampling times, it can be seen that sample *A* and *B* are divided into different time snapshots even though they actually have a tight relationship. If merging *s1*, *s2* and *s3* into one long-time snapshot, a cluster including points *A* and *B* will be founded during clustering and the cluster will be a potential companion candidate for the following processes. In fact, *A* and *B* are traveling buddies, whereas *C* is a real noise point. That is the classic companion candidate omitting problems caused by inappropriate time snapshot boundaries. This problem arises more frequently when trajectory data is more sparse.

The probability of this problem is related to the length of the time segment. For example, if raw data contains trajectory records in a region within one day, researchers would hope to avoid omitting problems as much as possible. There are two choices about whether to slice 24 h according to 5 min, or directly calculate according to the whole 24 h. If choosing time-segment based on short-term slicing, it will lead to an increased possibility of introducing problems. On the contrary, if a method can use 1 day or several hours of trajectory data as input this possibility will be greatly reduced. Obviously, frequent short-time slicing can easily introduce the Companion Candidate Omitting Problem, resulting in non-noisy records being filtered out.

From a holistic perspective, we take records in a longer-term time span as mining targets. Closer geographical similarity and closer temporal dimension features are concerned. The trajectory clustering algorithm is used to mine the similar features in spatio-temporal dimension for these records among users. Meanwhile, the frequent accompanying situation is regarded as the important standard to discover Associated Traveling Companion Candidate (ATCC). Then the characteristics of the accompanying scenarios are combined to specifically confirm the accuracy of the accompanying situation. It will greatly improve the robustness of the method to the degree of data density.

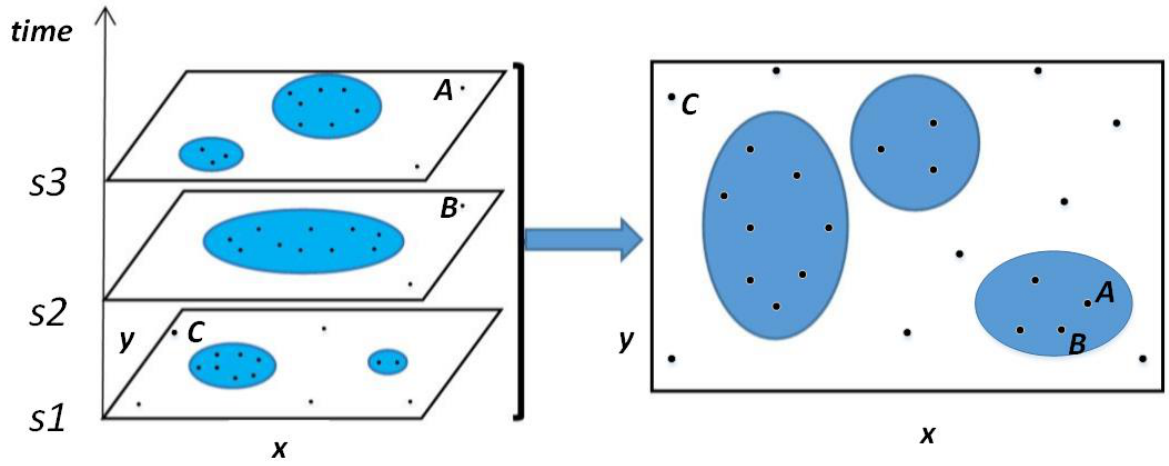


Figure 1. An example of companion candidate omitting problem.

3.1.2. Problem Definition

- **Definition 1 (Snapshot Set):** A time snapshot set $\mathbf{S} = \{\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_n}\}$is a collection of series of short-time snapshots. which can be seen as an extension to a shorter-time snapshot.
- **Definition 2 (Record Group):** A Record Group $\mathbf{R} = \{\mathbf{r_1}, \mathbf{r_2}, ..., \mathbf{r_n}\}$ is a collection of all moving object records in a snapshot set $\mathbf{S} = \{\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_n}\}$, n represents the number of moving objects within the time set. For a moving object o_j , the number of the records $\mathbf{r^j}$ is $\mathbf{k}$, and $\mathbf{r^j} = \{\mathbf{r_1^j}, \mathbf{r_2^j}, ..., \mathbf{r_k^j}\}$
- **Definition 3 (Locational Potential Candidate (LPC)):** A Candidate Set $\mathbf{C} = \{\mathbf{c_1}, \mathbf{c_2}, ..., \mathbf{c_m}\}$ is a set as a set of companion candidates clustered by location information, where m represents the number of clusters. This paper uses the density-based clustering algorithm. Some parameters need to be defined. δ_s is defined as a size threshold of clustering, ε is used as a distance threshold. The default distance formula of several clustering algorithms is based on the Euclidean distance formula, which can provide certain efficiency advantages. However, in order to facilitate the parameter setting of trajectory data mining and improve the accuracy of trajectory data mining results, the distance formula here may be replaced by a distance formula that better meets the needs of the scene. A locational potential candidate set is a cluster set satisfying w.r.t. δ_s and ε.
- **Definition 4 (Time and Location Potential Candidate (TLPC)):** On the basis of potential candidates for position, the clusters of the candidates satisfy clustering based on time to form clusters. The collection of objects in these clusters is regarded as Time and Location Potential Candidate. Among them, δ_s^t is defined as the minimum cluster size. In addition, because *HDBSCAN* is used to weaken another distance parameter, it is not defined here.
- **Definition 5 (Associated Traveling Companion Candidate(ATCC)):** min_sup is the minimum support threshold for the association analysis and min_conf is the minimum confidence threshold. The candidate set $\mathbf{M} = \{\mathbf{m_1}, \mathbf{m_2}, ..., \mathbf{m_q}\}$ satisfies an association rule dictionary $\mathbf{W}$. The key-value pair of the dictionary $\mathbf{W}$ corresponds to the frequent item and its support. $\mathbf{m}$ is a frequent item with its support not less than the minimum support. The key of the association rule is a frequent item $\mathbf{M}$ with its confidence is not less than the minimum confidence.
- **Definition 6 (Pseudo-companion Scenarios):** The Pseudo-companion scenarios refer to scenarios that already have potentially associated companionship while some important features do not fully conform to the accompanying pattern.
- **Definition 7 (Tolerance Strategy):** When performing trajectory data mining in a sparse data set, some parameters cannot be set strictly. Otherwise, it will be difficult to find the research objects that meet the relevant conditions. For this reason, a Tolerance Strategy needs to be considered to discover moving objects.
- **Definition 8 (Traveling Companion (TC)):** $\mathbf{Q} = \{\mathbf{q_1}, \mathbf{q_2}, ..., \mathbf{q_n}\}$ is a set of traveling companion, where a traveling companion group q_i is a group that satisfies the number of records satisfying

the potential accompany situation is greater than the frequency threshold δ_f, and the proportion of the records satisfying is greater than the percentage threshold δ_r within the time period S.

3.2. Methodology

3.2.1. Framework

Raw trajectory data generated from different sensing sources has different data formats and positional accuracies. A flexible framework named GroupSeeker is proposed to discover traveling companions from those divers trajectory data. The framework primarily includes a five-stage processing flow, which is composed of Data Preprocessing, Spatio-temporal Clustering, Candidate Voting, Pseudo-companion Filtering and Group Merging. Then a series of parameter-setting strategies throughout the whole processing flow are proposed to deal with different scenarios. The entire processing is shown in Figure 2. The various categories of sampling methods could bring several different characteristics of trajectory data and this paper focuses on two sampling methods, i.e., GPS and CDRs, which have the characteristics of collecting easily and having huge scales.

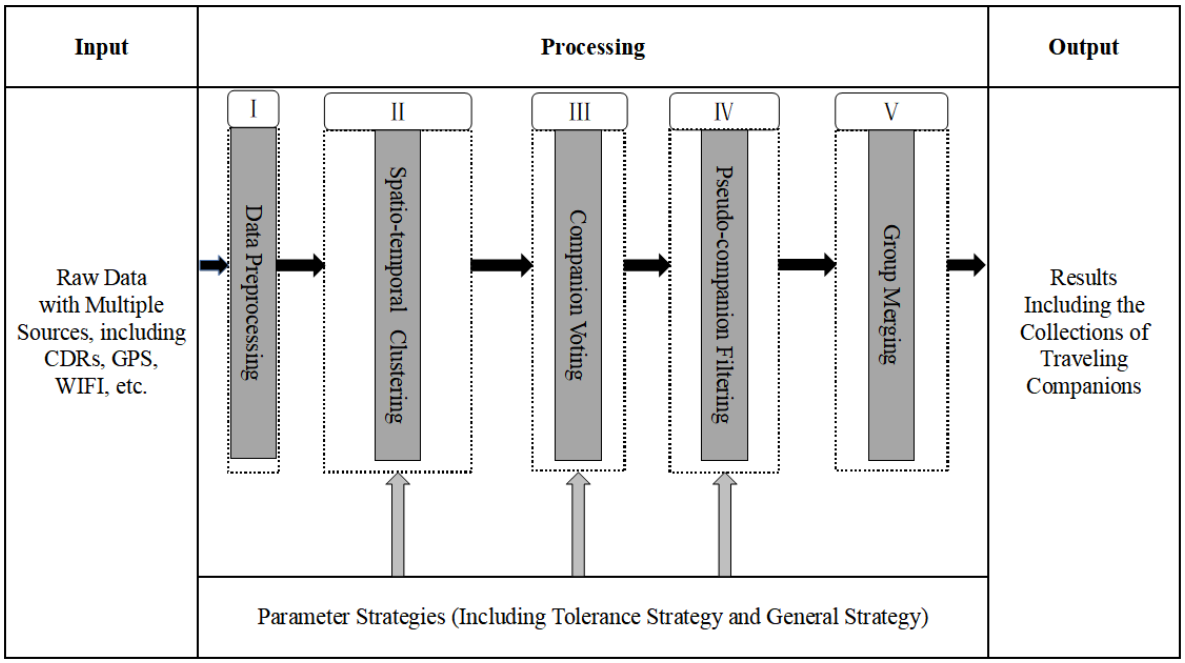


Figure 2. The framework of the entire processing.

Data Preprocessing removes unnecessary fields in raw trajectory data and filters noise and redundant data in remaining fields. Then the entire trajectory data is split into many sub-trajectory data sets to reduce the computational overhead. In the Stage II, Spatio-temporal Clustering, trajectory data is clustered in the spatial dimensional to discover Location Potential Candidate (LPC). Through clustering these LPC, Time and Location Potential Candidate (TLPC) can be discovered from the temporal dimensional. In addition, Candidate Voting stage focuses on the accompanying frequency between each pair of users in TLPC to discover the Associated Traveling Companion Candidate (ATCC). Subsequent Stage IV is Pseudo-companion Filtering that aims to offer some rules to filter some confusing pseudo-companions from ATCC. At the last stage, called Group Merging, it is to merge the companion sets with the same moving objects to make them as an accompanying group with multiple objects. As a semi-supervised framework, parameter-setting strategies could offer some significant strategies to guide these methods in Stage II, stage III and Stage IV to set relatively appropriate parameters.

3.2.2. Data Preprocessing

Stage I, Trajectory Data Preprocessing, aims to standardize the raw trajectory data, no matter what type of data source it comes from. A set of standardized sample data is shown in Table 1. Since many redundant fields are included in the raw data, such as acquisition-action number, base-station number, cell number, operator code, administrative-area code, and altitude, acquisition action number, base station number, cell number, operator code, administrative area code, etc., it is necessary to use various preprocessing methods in this stage including noise filtering (trajectory cleaning), trajectory segmentation, map-matching.

Table 1. Data pre-processed form (from two data sources).

Data_Source	Time	User_ID	Latitude	Longitude
Traveling User Dataset	2014-11-16 14:53:56	XXXXXXXXXX	4X.XXX627	8X.XXX317
GeolifeV1.3	2009-04-09 18:28:10	u3	39.999966	116.327415

Firstly, these redundant fields are abandoned and the remaining fields get uniform naming, such as *Time, User ID, Longitude, Latitude and an index number*. When cleaning these trajectories, some obvious noise points should be filtered, such as records containing error data type corresponding to a certain filed, records including wrong longitude and latitude in the range of known geographic area, and records containing timestamp that does not match the actual sample time. Besides, median filtering is used to deal with the single-noise point and Kalman filtering is used to deal with continuous-noise points. Using a stay point detection method through these filtered data, stay points in these trajectories could be found, which could be used to guide the further-patterns discovery. To reduce the computing scale for trajectory clustering and mining as much as possible about the behavior patterns among the sub-trajectory segments, the trajectory-segment operation is executed to divide the whole trajectory records into several sub-trajectories. We split a sparse trajectory data set into 18 sub-datasets and split Geolife trajectory data set into 19 sub-data set according to the number of records. A part of trajectory data is selected for map-matching to briefly verify the reliably of the trajectory data. In addition, the filtered data basically conforms to the map and there is no big drift.

3.2.3. Spatio-Temporal Clustering

To find representative sub-trajectories or public propensity behaviour through different moving users, trajectory clustering plays an important role by clustering similar trajectories. Generally, a feature vector is used to represent a trajectory. The similarities of two trajectories can be measured by calculating the distance between their feature vectors. The input of clustering algorithms in previous companion discovery is the data in a time segmentation. Because of the difficulty in collecting complete data and the data sparsity, it is a hazard to cluster these trajectory data using the unbefitting time segmentation method. Concretely, some cluster-able trajectory data cannot be clustered possibly and are even filtered as noise data. Therefore the approaches based on time-segmented slicing may not always be completely successful. Thus, a Spatio-temporal clustering for location-and-time dimension is proposed to solve these problems that cause omitting traveling companions.

Figure 3 shows the clustering process in detail. In this processing, HDBSCAN is used to discover Location Potential Candidate (LPC) and Time and Location Potential Candidate (TLPC). LPC shows the similarity in location attributes. On this basis, TLPC requires the similarity in the time dimension more strictly. Figure 4 shows an example of a specific process for combining data fields. In Figure 4a, a set of data samples is presented that several fields (User ID, Latitude, Longitude, Time) are the remained fields after preprocessing and the Fill field is added for these records as LPC. To discover LPC, two parameters are used to limit the minimum size of the cluster and the neighborhood-distance threshold, which make HDBSCAN get the steady and effective results to discover LPC and to filter some noise which cannot be clustered. A Fill field is increased into the collection of LPC to increase the dimension to meet

the requirement. The value of FILL field is set to 1 to simplify the calculation. In each cluster of LPC, HDBSCAN is executed once to find TLPC, including similar time-and-location characteristics, and to filter some noise records. The Figure 4b. illustrates this process visually. In Algorithm 1, steps 4–8 show the stage from the algorithm level. Notably, the number of these filtered records could influence the promotion of satisfying records. For different research purposes, they are valued differently.

Algorithm 1: Spatio-temporal Clusteirng and Companion Voting Algorithm.

Input: Trajectory records set R, a period time S; Distance thershold ε, size threshold δ_s for location clustering, the size threshold δ_s^t for time clusteirng; support threshold min_sup and confident threshold min_conf

Output: Assocaited Users frequencey-itemset Set M

1 location potential candidate set $C_l = \Phi$;
2 location and time potential candidate set $C_l^t = \Phi$;
3 User ID set $U_{ID} = \Phi$;
4 $C_l \leftarrow$cluster R with ε and δ_s during S;
5 **foreach** *cluster* $c_i \in C_l$ **do**
6 $\quad C_l^t \leftarrow$cluster c_i with δ_s^t;
7 $\quad$ **foreach** *cluster* $c_j^t \in C_l^t$ **do**
8 $\quad\quad A_{ID} \leftarrow$ the account ID of c_j^t;
9 associated companion frequent-itemset set $M = \Phi$;
10 associated rule dictionary $W = \Phi$;
11 frequent itemset set $F = \Phi$;
12 Using FP-growth algorithm $F \leftarrow$ find frequent itemsets in U_{ID} ;
13 $W \leftarrow$ get (f_i, con_i) by finding items f_i in F and its confidence $con_i \geq min\_con$;
14 **for** $(f_i, con_i) \in W$ **do**
15 $\quad M \leftarrow$ get the user ID u_i from f_i
16 return M;

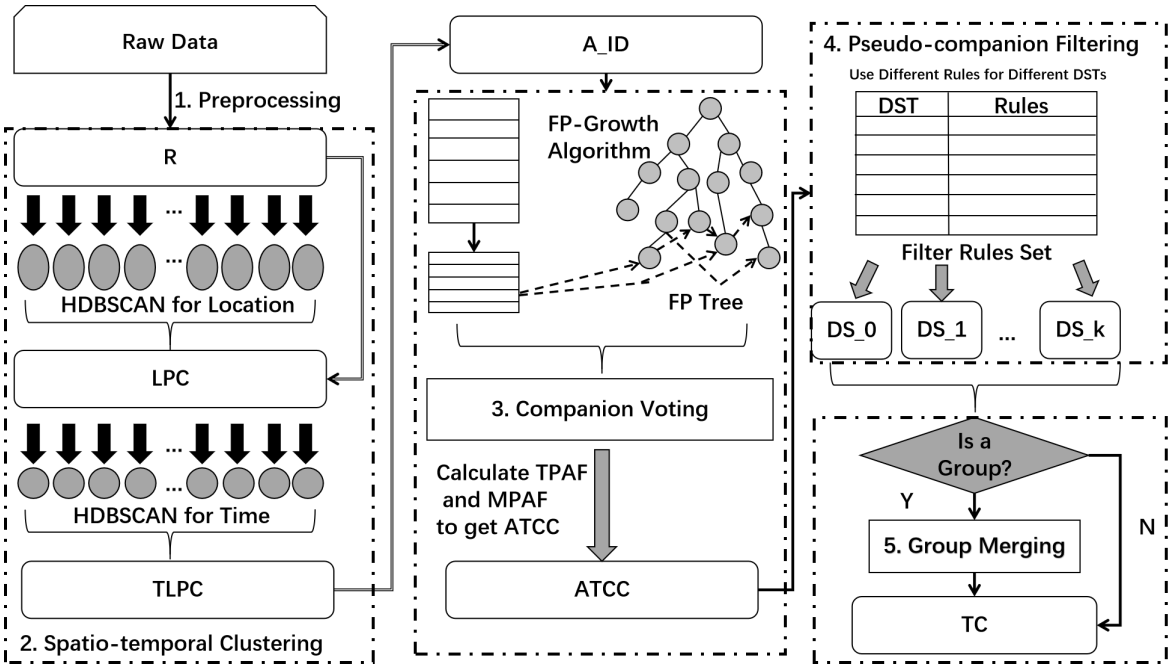
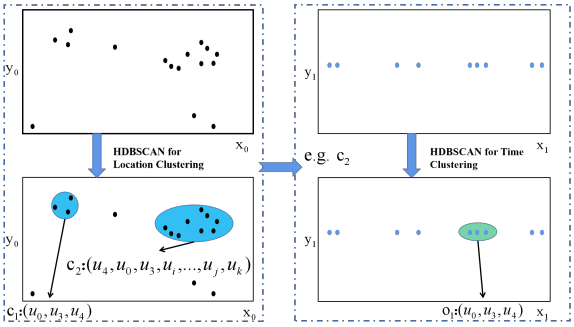


Figure 3. The Detailed Process Example of Methodology.

User ID	Latitude (x_0)	Longitude (y_0)	Time (x_1)	Fill (y_1)
u0	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 11:50:58*	*1*
u3	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 11:50:59*	*1*
u4	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 11:51:04*	*1*
u30	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 11:51:30*	*1*
...	...	...	...	...
u4	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 12:33:30*	*1*
u0	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 12:33:32*	*1*
u3	*xx.xxxxxx*	*xx.xxxxxx*	*2009-03-05 12:33:35*	*1*
...	...	...	...	...

(**a**) data samples.



(**b**) clustering processes.

Figure 4. Trajectory Clustering for Discovering Time and Location Potential Candiate.

3.2.4. Companion Voting

Mining the frequent pattern is to discover the frequent-temporal mode from the extensive trajectory data, which could mine the rules of publicity or frequent paths in public trajectories. In this method, information such as location, time and semantic information could be combined to mining the characteristics of moving objects.

Stage III takes advantage of the FP-growth algorithm to discover Associated Traveling Companion Candidate (ATCC) and FP-growth algorithm is a tree-based method using the frequent items. A technique is used to shorten the time for this algorithm to search through the suffix tree. Specifically, because the FP-growth searches from the frequent single-item set to the frequent n-item set, the potential accompanying situation between two users will be focused on and the convergence time will be reduced greatly, if the length of the suffix is set to 2. In order to further mine users with accompanying patterns, the occurrence of associations between users is worth noting. In addition, a collection including all of the user set from each TLPC is regarded as the target to find ATCC. *The Total Proportion of Accompanying Frequency (TPAF)* between two users in this collection is calculated by Equation (1), which will be compared with a parameter and be used to vote for ATCC. The Mutual Promotion of Accompanying Frequency (MPAF) is calculated using Equation (2) to judge the occurrence of the accompanying pattern between two users. Furthermore, it will be compared with a threshold to decide whether to vote for these two users meeting ATCC. In Algorithm 1, steps 12–15 show how to discover ATCC using FP-growth. Figure 3 shows the Companion Voting process in detail, such as building FP-tree, calculating TPAF and TPAF to get ATCC.

If two records of user X and Y that want to analyze the TPAF, the corresponding *TPAF* is:

$$TPAF(X,Y) = P(XY) = \frac{number(XY)}{number(All\_Samples)} \tag{1}$$

For X and Y, the MPAF is obtained as:

$$MPAF(X \Leftarrow Y) = P(X|Y) = P(XY)/P(Y) \tag{2}$$

3.2.5. Pseudo-Companion Filtering

Pseudo-companion Filtering aims to offer significant rules for filtering some confusing Pseudo-companion scenarios. These pseudo-companion situations in discovering Traveling Companion Patterns from the ATCC are from the intermediate results in previous processes. Figure 3 shows the Pseudo-companion Filtering process in detail and the process need to use the Filter Rules Set for Different Data Sources (DTS).

Due to the diversity of trajectory data, there is no recognized method of confirmation to evaluate the sparseness of trajectory data. Combined with the analysis of experimental data, the sparseness of the trajectory data discussed in this article can be regarded as follows. It is the average value of the

number of individual user records per unit time as a standard. When this value is larger, the data set is denser, otherwise the data set is sparse. Generally, positioning and sampling are used to quickly determine the sparseness of data sources. According to the sparseness of the data source, different data types at this stage will correspond to different rule sets, which will affect the time cost of this stage but improve the accuracy of the results.

The sparseness of the data affects the judgment of such pseudo-complications. Therefore, it is necessary to distinguish between pseudo-companion scenarios in different data sources. Through the visualization of the intermediate results of the experiment and the situation of mobile data sources, we briefly distinguished the pseudo-companion scenarios in the two types of data types. In order to prevent these scenes from interfering with the real accompanying results, the necessary filtering rules are proposed. Table 2 shows these scenarios, scenario descriptions and corresponding rules.

For different data sources, there are some differences in the factors that distinguish Pseudo-companion scenarios. For example, in a long-term period, users from ATCC may not have many accompanying records for sparse sample sets. Meanwhile, it can be determined that they have accompanying circumstances. Certainly, it may be identified as a short-term encounter.

In Algorithm 2, steps 1–20 show this stage. Among them, steps 1–2 are short pseudo codes of this stage. Steps 4–20 clearly show one of the rule sets, which is a scenario of filtering brief contact in the intermediate result set from the sparse data source.

Table 2. Pseudo-companion Filtering: Scenario Names, Description and Rule Sets.

Data Source Type	Scenarios Names	Scenarios Description	Filtering Rules
Traveling Users Data	Breif Contact	There is a small amount of close contact in the total record of A or B within a small area.	The number of pseudo-accompanying records is small and the total number of records is relatively small. If either is less than the relative threshold, the two objects are filtered.
	No-contact	There is almost no close contact in the total record of A or B within a small area.	Pseudo-accompanying cases account for a so small proportion but the distance between the central geographic location of two objects is within the signal strength range of a base station.
Geolife	Breif Encounter	A and B have frequent contacts in a small area within a short-term period within a small area.	The time span of the accompanying records is short-term. The directions of these moving objects change after these records. There is no accompanying record for a long-term period.

3.2.6. Group Merging

The purpose of Stage V is to discover Traveling Companions including multiple users rather than only a pair containing two users. The set including multiple users is regarded as a group. Figure 3 shows the Group Merging process in detail. It is necessary to identify whether there is a group accompanying situation and decide to merge them. In the Stage III for discovering Associated Traveling Companion Candidate (ATCC), a trick is used to optimize computational overhead to reduce the convergence time, which leads the research scenarios to be discussed between two users. However, the virtual number of accompanying users may be multiple, such as tourist groups, participants in group activities of a family of three in shopping, etc. From the perspective of the designer, it is necessary to mine further the final stage for the Multiple-User situation that may exist among the traveling companion candidate previously discovered. If these users are filtered in Pseudo-companion Filtering and are stayed, they should be merged using existing common sub-sets. For instance, for the set $\{\{u_0, u_3\}, \{u_3, u_4\}\}$, because two of the sub-items contain a common sub-set $\{u_3\}$, we merge the two sub-items and remove the other true subsets. Finally, the set changes to $\{\{u_0, u_3, u_4\}\}$. In Algorithm 2, steps 19–25 show this process in the last stage.

Algorithm 2: Pseudo-companion Filtering and Group Merging Algorithm.

Input: associated companion frequent-itemset set M, a frequency threshold δ_f; Trajectory records set R; distance threshold δ_d, time difference threshold δ_t records number threshold δ_r;

Output: Traveling Companion Set Q

```
foreach m_i∈M do
    execute the corresponding rule sets;
    #e.g. in Step 4 to 20#;
    num_1 ← 0 ;
    num_2 ← 0 ;
    D1, D2 ← R_{m_i^1}, R_{m_i^2};
    N_1 ← The number of records in D1;
    N_2 ← The number of records in D2;
    foreach D1_j∈D1 do
        foreach D2_k∈D2 do
            T1 ← T[D1_j];
            T2 ← T[D2_k];
            if (△T(T1, T2)≥δ_t) and (d(D1_j, D2_k) ≥δ_d) then
                num_1←num_1 + 1;
                num_2←num_1 + 1;
                break;
    if (num_1≤δ_f) or (num_2≤δ_f) then
        remove m_i from M ;
    if (num_1 / N_1≤δ_r) or (num_2 / N_2≤δ_r) then
        remove m_i from M ;
foreach m_i∈M do
    foreach m_j∈M do
        if i is j then
            continue;
        else
            m_i←m_i∪m_j
remove all dupicate subsets in M;
Traveling Companion Set Q = Φ;
for each m_i∈M do
    q_i← records r_{m_i} in R during S;
    Q← all of pi;
return Q;
```

3.2.7. Parameter Setting Strategy

Since many factors need to be considered in the scenario of discovering traveling companion patterns by combining with real-world data samples, the algorithms related to parameter settings are used in three significant stages of this framework (Spatio-temporal Clustering, Companion Voting, Pseudo-companion Filtering). Some of these algorithms have obvious semi-supervised methods. Although we have reduced the number of parameters and simplified the complexity of using those as much as possible during the implantation of important algorithms. For example, we no longer consider using DBSCAN but use HDBSCAN as a clustering algorithm, it is inevitable to think about

optimization of existing parameters and establishment of a set of strategies. The necessary parameter strategy will boost the effectiveness and efficiency of the method, which could reduce the learning cost for this method. In addition, all parameter notations are archived in Table 3.

- General Strategy: The general strategy is explained here in order to highlight the tolerance strategy. First, the haversine formula is a formula especially calculated to the distance between two points through their latitudes and longitudes. Many clustering algorithms include a parameter called "metric", which can be set as "haversine". Secondly, for discovering Traveling Companions, the minimal clustered number for clustering should be larger than 3 to reduce the number of clusters. Moreover, for the support-and-confidence setting, Table 4 shows a preliminary correspondence between participation and confidence level. We hope to guarantee a higher confidence level, so the default confidence value set in this study is 0.6. For the support level, we will focus on the frequency of the target object at the same time and not necessarily require to get a ratio. Finally, it is important for the consistency of the results of a data set to ensure the distance threshold parameter. For instance, for ε and δ_d, they are set to the same value in consideration of sampling accuracy at different stages. Absolutely, if the purpose of applications requires stricter filtering, it needs to set the latter parameter smaller.
- Tolerance Strategy: Compared with the strictness of the general strategy, the tolerance strategy provides good support for the data sets from some special data sources, such as CDRs. Besides, it is difficult to give a clear value range for some parameters for various data sets, while the proposed tolerance strategy can guide users to weaken some parameter setting ideas from the purpose of mining. The original intention of this strategy is that for data samples with higher sparseness, strict threshold constraints are bound to make the result set as small as possible. In fact, the setting of this strategy comes more from the practicality of the results. In this field, the sparseness of trajectory data has always been a major challenge. At the same time, it is difficult for some specific data sources to collect data information of all users in a specific geographic area within a long period of time. This results in the sparseness of real-world data that is reasonable and unavoidable. For this reason, researchers should hope to make full use of each recorded information (except obvious noise). Specifically, for some important scenarios, such as mining the behavior patterns of specific groups and specific individuals to discover the traveling companion pattern, sometimes various factors disturb the collecting process so that these data are caused to be sparse. In this case, the tolerance strategy can better prevent some records from being strictly filtered out, which is more likely to find other related moving objects. In our study, it is important for δ_f and δ_r in data source D1 to consider tolerance. These two parameters can be set to larger values to limit the confusion scenarios, such as only a small number of records are related and most of the records are far apart, or the number of records of an object is so small that it should be filtered out.

Table 3. Description of Parameter Notations.

Parameters	Description	Parameters	Description
ε	the distance threshold in HDBSCAN	δ_s	the minimal clustered number for location clustering
δ_s^t	the minimal clustered number for time clustering	***min_sup***	the minimal support threshold for FP-growth
min_conf	the minimal confident threshold for FP-growth	δ_f	the minimal frequency threshold for the records number
δ_d	the minimal distance threshold between two records	δ_t	the maximal time span threshold between two records
δ_r	minimal records promotion threshold	**metric**	distance formula

Table 4. Correspondence Table of **min_conf** Value and Confidence Level.

min_conf	[0, 0.2]	[0.2, 0.4]	[0.4, 0.6]	[0.6, 0.8]	[0.8, 1.0]
Confidence Level	low	relatively low	medium	relatively high	high

4. Experiment and Results

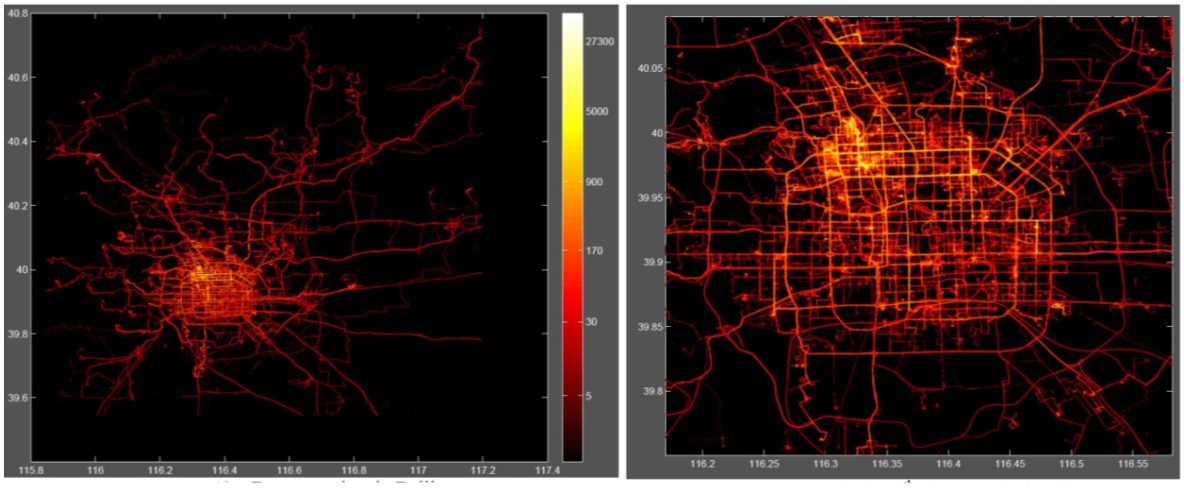
All the algorithms are implemented in python 3.8.2 on PyCharm and are performed on computers with Intel Core i7-8550U CPU 1.80 GHz, 16.0 GB RAM and windows 10.

4.1. Data Sets

Based on the two real-world data sets, various sample sets are extracted based on different criteria.. The criteria are shown as follows:

- The Sampling Frequency
- The Number of Records for Individuals
- Effective Duration
- Data Collection Period

- **D1 (Traveling Users Dataset):** This dataset is collected from real users in a certain region of China between 16 November 2014 and 18 November 2014, which was provided by a communication provider in China. The locations are from the cell-sites which are connected with many phones. The raw spatial trajectory data mainly includes the latitude and longitude coordinates, time-stamp and user information. When we got this dataset, personal-sensitive information in the dataset was anonymized and the coordinate information was re-adjusted by this provider for privacy protection.
- **D2 (Geolife Trajectory):** This dataset was collected in (Microsoft Research Asia) Geolife project from 182 users between *April 2007 and August 2012* [51–53]. A GPS trajectory from that set is represented by a sequence of time-stamped points containing information on latitude, longitude and altitude. **91**.5% of the tracks are in a dense representation, e.g., every 1–5 s or every 5–10 m per point, the overview of this data set shown in Figure 5:



(a) Data overview in Beijing (b) Within the 5th Ring Road of Beijing

Figure 5. (**a**,**b**) Overview of D2 [53].

It is essential to choose suitable data sets. For D1 and D2, after data preprocessing, they are divided into many subsets according to the number of records. For instance, we split Geolife data set into 19 subsets according to the amount of 800,000 records. In these subsets, we choose 5 subsets

from D1 and D2 relatively, which are shown in Table 4. Notably, a simulated data set called Sim1 is generated based on a subset from D2. Sim1 is regarded as a subset from the real-world and simulated sources D3.

Despite the fact that Sim1 has a small size, it contains two companion simulation users we added for a particular user, which could quickly verify the effectiveness of the algorithm. The generation of the two simulation data comes from understanding the trajectory of a real user, especially to be able to have a simple understanding of its state changes, the most basic of which is its direction change in the two dimensions of latitude and longitude. By recording a state-change matrix, the basic state changes can be learned from the simulation data and hence the traveling companion's behavior can be simulated.

Except for Sim1 and Set5, other subsets have similar data size and a similar number of records. 10 samples subsets (Set1–Set5 and Geo1–Geo5) are randomly selected from D1 and D2 respectively in order to compare with the impact of the sparseness and density of the dataset in the real-world scene on the algorithm results. Set5, whose size is about half of the remaining 9 sample sets, is used to show the effect of data size on the method. Certainly, for dealing with the scale of 800,000 records, our experimental environment can be close to the its memory limit.

4.2. Pseudo-Companion Scenarios Filtering Display

Some typical intermediate results are visualized in Figures 6 and 7, which are the situations that need to be filtered out. In order to facilitate the display, we select a data type of rule set to use. It is the case of the two-types scenarios in the sparse data set.

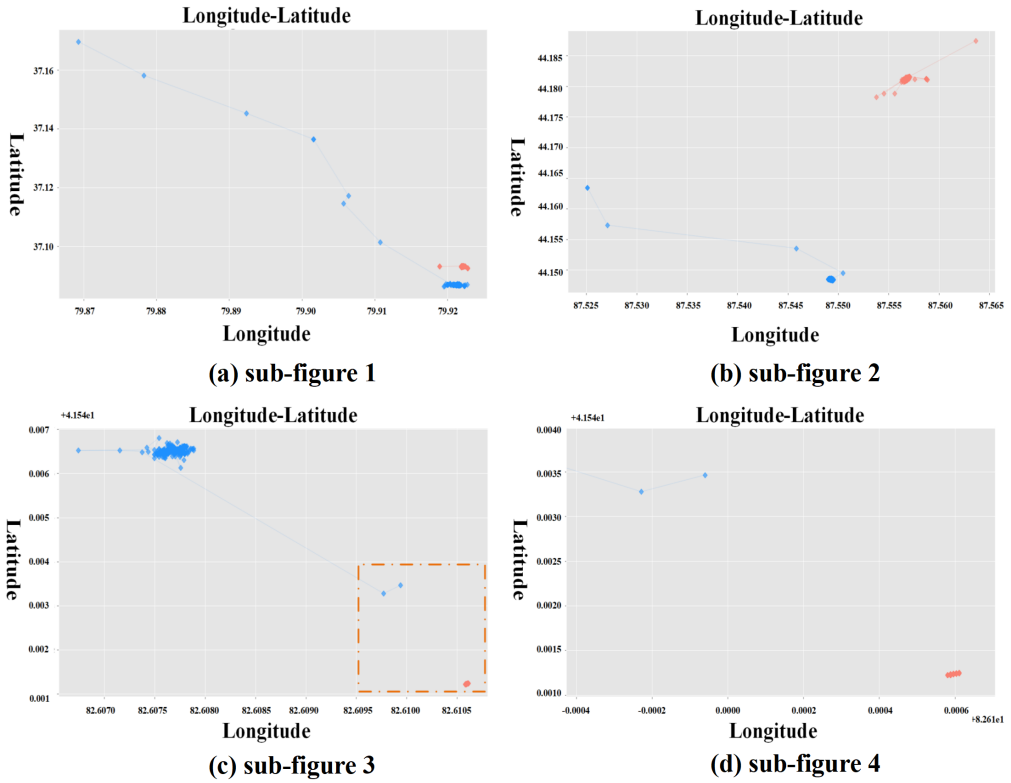


Figure 6. (**a**–**d**) Brief Contact and No-contact.

In sub-figures of Figure 6, although there is brief contact between two users. For one of two users, the number of records representing the contact processing does not stand at a big proportion of the total number of records. Hence, they are filtered by the rule sets. For the Figure 6b, they could be regarded as the no-contact scenario because they have few records presenting close contact. Finally,

the Figure 6d. is a partial enlargement of Figure 6c. and the close-contact records between two users still account for too few, so they are not considered to be real companions satisfying the proportion of records. The sub-figures in Figure 7 show the cases of satisfying the filtering rules. Among them, the Figure 7a. is the result of *Sim1* including three users. These users move together in a small area. In addition, the Figure 7d. is the partial enlargement of Figure 7c.

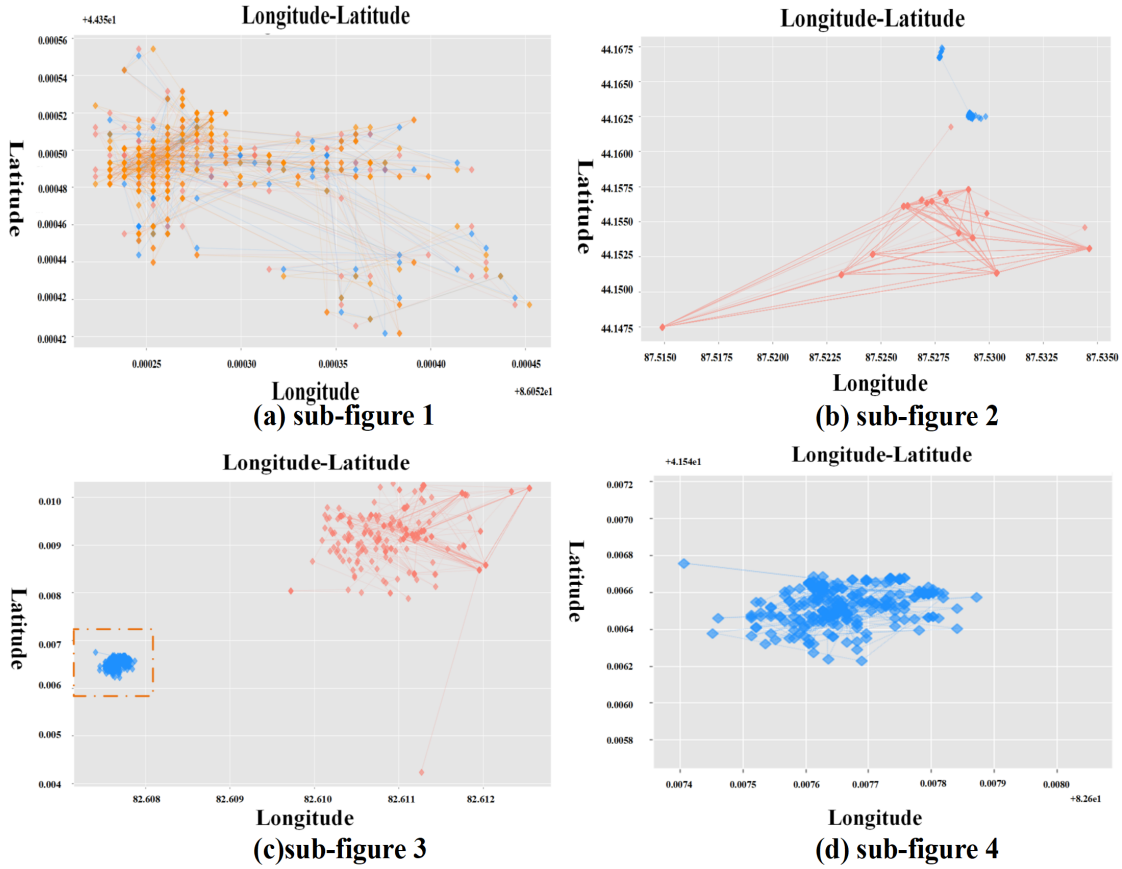


Figure 7. (**a**–**d**) Filtered Results.

4.3. The Results of Traveling Companion Discovery and Validation

4.3.1. Measuring Time Overhead

Table 5 highlights the time overhead of 10 data subsets in the framework, which is illustrated in Stage II to Stage V. It is evident that Stage II is the largest time-overhead stage in these 10 data subsets and has great differences between D1 and D2. The time overhead in Stage III is affected by the scale of the data subset. In Stage V, when the result of the previous stage leads to the absence of multiple targets, its time overhead will be 0. In addition, we use the average number of users' records to show the sparseness of each data subset. Obviously, D1 is more spare than D2. Thus, the parameter setting should not be too strict for D1. Otherwise, it would be difficult to discover TC. In practice, the filter rules for this result is to filter the brief contact for D1. Since this rule set is not used in D2 with a dense sampling effect, so the time cost is 0 in Stage IV for D2. Finally, it is worth mentioning that the time overhead in Group Merging is too shorter than other stages to be negligible in this scale of data set. Therefore, the overhead in Stage V is not shown in Table 6. Related parameter settings are shown in Table 7. The distance are measured in meters, and the time threshold are measured in seconds.

Table 5. Time Overhead in Main Stages and The Number of Results.

Source	Data Set	Duration2	Duration3	Duration4	Total Duration	Average Duration for D1/D2	TC#	Total TC Users#
	Set1	5724.4	2.1	50.3	5776.8		7	14
	Set2	5572	2.1	55.5	5629.6		2	6
D1	Set3	5421.9	1.7	1.5	5425.1	5664.78	1	2
	Set4	5381.5	2.8	145.8	5530.1		1	2
	Set5	5961.8	0.5	0	5962.3		0	0
	Geo1	3192.2	0.1	0	3192.3		0	0
	Geo2	2428	0.2	0	2428.2		1	2
D2	Geo3	2702.8	0.2	0	2703	2847.5	1	4
	Geo4	2890.3	0.2	0	2890.5		1	2
	Geo5	3023.3	0.2	0	3023.5		1	3
D3	Sim1	413.4	0.2	11.8	425.4	425.4	1	3

Table 6. Data Sets Information.

Source	Data Set	Data Size	Number of Records	Duration
	Set1	51,256 KB	800,000	24H
	Set2	51,246 KB	800,000	24H
D1	Set3	50,880 KB	800,000	10H
	Set4	50,930 KB	800,000	8H
	Set5	29,469 KB	464,746	5H
	Geo1	41,080 KB	800,000	17D
	Geo2	41,496 KB	800,000	30D
D2	Geo3	41,874 KB	800,000	17D
	Geo4	41,927 KB	800,000	22D
	Geo5	41,927 KB	800,000	31D
D3	Sim1	7848 KB	125,367	80M

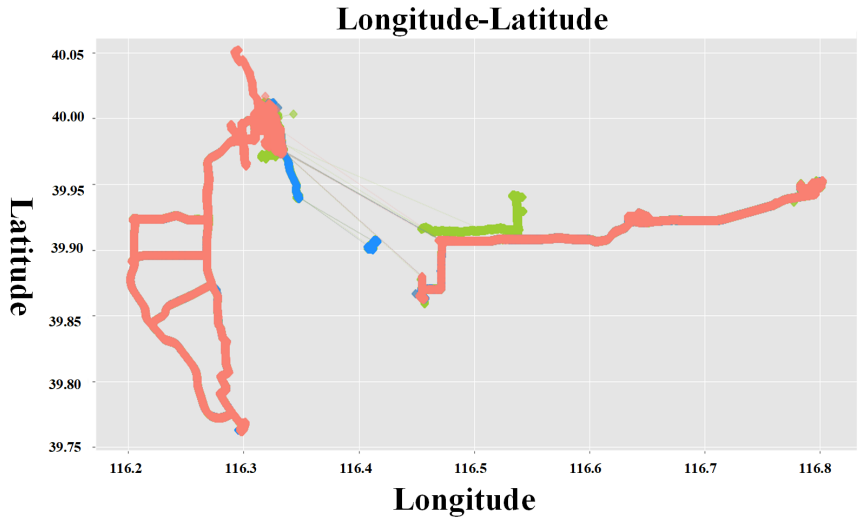
D: day(s); H: hour(s): M: minute(s).

Table 7. Parameter-setting table corresponding to this experiment result.

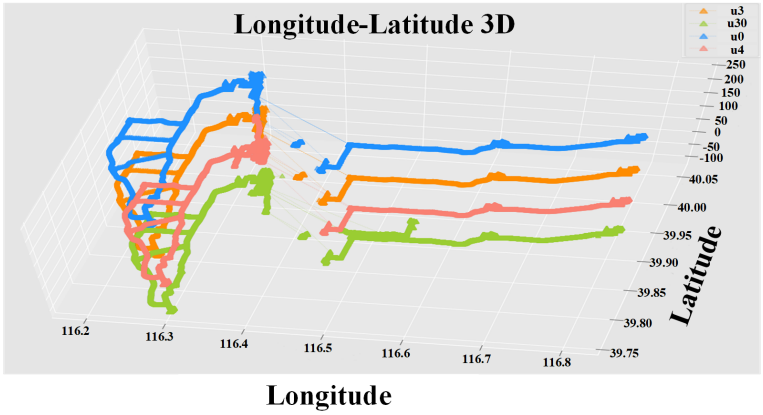
Source	ε	δ_s	δ_s^t	δ_d	δ_t	min_sup	min_conf	δ_r
D1	500	4	4	200	120	0.6	0.8	0.5
D2	5	4	4	5	5	0.6	0.8	-
D3	500	4	4	200	120	0.6	0.8	0.5

4.3.2. Significant Result Analysis

The number of TC in each data subset is shown in Table 5. Although some subsets produce a few or no results, it matches the real-world data scenarios with no accompanying pattern. In the following, some special and meaningful TC are presented by visualizing the experimental results. For instance, u0, u3, u4, and u30 are recommended from Geo3 as a TC. In the this long-term period of Geo3, all of them move through the road network in this geographic area within a close period of time. Therefore, their trajectories, which are shown in Figure 8, are very similar and the coverage rate among them is so high. The main difference is shown in Figure 8a. that is a small part of the trajectory difference exists, which may result from a short-term separation or a certain amount of data loss caused by a difference in the positioning signal. On the other hand, u0 and u3 have the same records within a long-term period. We further checked the undivided D2 dataset to verify this situation. It has been found that their records appeared same from 0:52 on 30 March 2009 to 2:58 on 5 July 2009. Therefore, it is reasonable to guess that this is likely to be the case of an individual carrying two mobile devices, which could offer positive support in the management of special objects, such as focusing on individuals or groups with sensitive behaviors. The sample data is shown in Table 8.



(**a**) sub-figure 1.



(**b**) sub-figure 2.

Figure 8. (**a**,**b**) Typical results in *D2*.

Table 8. Partial Data of Experimental Results.

Data Source	Time	User_ID	Latitude	Longitude
D2	2009-04-09 18:28:25	u0	39.999912	116.32751
	2009-04-09 18:28:25	u3	39.999912	116.32751
	2009-04-09 18:28:27	u30	40.000008	116.327446
	2009-04-09 18:28:28	u4	39.99983	116.32712
	2009-04-09 18:28:29	u30	40.000008	116.32754
	2009-04-09 18:28:30	u30	39.999996	116.32745
	2009-04-09 18:28:30	u3	39.999924	116.327484
	2009-04-09 18:28:30	u0	39.999924	116.327484
	2009-04-09 18:28:31	u30	39.99999	116.32748
	2009-04-09 18:28:33	u4	39.99989	116.32744
	2009-04-09 18:28:35	u0	39.9999	116.32745

5. Conclusions and Discussion

At present, mobile positioning devices represented by navigation devices, smart wearable devices, and smart infrastructures are increasingly popular in daily life. LBS has become an important element, and which is not available to most people. Locatable devices and LBS provide sufficient conditions for

generating a massive amount of mobile trajectory data. The trajectory traveling companion discovery algorithm is widely used as an important method for discovering accompanying behavior patterns. However, it is necessary to improve the applicability and efficiency of the method as much as possible under the premise of current information explosion and diverse sampling methods.

Thus, as one basic support technology of many trajectory data mining applications, this paper proposes an applicable framework GroupSeeker to discover traveling companions in vast spatial-temporal data. The framework includes a five-stage processing flow and the core algorithms lie in the following three stages, Spatio-temporal Clustering, Companion Voting, and Pseudo-companion Filtering. GroupSeeker successfully avoids the problem that useful clusters are considered to be noise due to bad time segmentation. Besides, considering the different sparseness of data sources, the parameter setting strategies are proposed to improve the reliability of the framework and reduce the learning cost. Moreover, a set of imperfect but indeed effective methods for filtering confusing scenarios is proposed. In practice, parameters in GroupSeeker could be set according to the purpose of mining and specific scenarios. Finally, the framework is evaluated on several real-world datasets with different sparsity and data sizes. The experimental results show practically efficiency and stability.

In the future, more focus can be given to how effectively extract the features in the Pseudo-companion scenarios. Besides, it is necessary for the framework to further reduce the number of parameters and to simplify the parameter-setting strategies. In addition, if the entire framework can be upgraded in combination with a high-performance parallel and distributed computing solution to reduce the overhead time in Clustering Stage, the efficiency of the whole framework will be better optimized. Moreover, we plan to use a large amount of labeled accompanying trajectory data combined with machine learning methods to conduct more detailed rule formulation and algorithm design for the Pseudo-companion Filtering stage in our future work.

Author Contributions: Conceptualization, Ruihong Yao, Fei Wang and Shuang Zhao; methodology, Ruihong Yao; software, Ruihong Yao; validation, Ruihong Yao and Fei Wang; formal analysis, Ruihong Yao; investigation, Ruihong Yao and Fei Wang; resources, Shuhui Chen and Fei Wang; data curation, Ruihong Yao; writing–original draft preparation, Ruihong Yao and Fei Wang; writing–review and editing, Fei Wang and Shuang Zhao; visualization, Ruihong Yao; supervision, Fei Wang and Shuhui Chen; project administration, Shuhui Chen; funding acquisition, Shuhui Chen. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANPR	Automatic Number Plate Recognition
ATCC	Associated Traveling Companion Candidate
CDRs	call detial records
DBSCAN	density-based spatical clustiny of application with noise
DENCLUE	density-based clustering
DTS	Different Data Source
DTWD	Dynamic Time Warping Distance
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
HD-FIM	a breadth-first and depth-first hybrid distributed approach with Frequent itemset mining
ITs	intelligent system
LBS	Location-Based Service
LPC	Locational Potential Candidate
MBB	Minimal Bounding Box

MBR	Minimum Bounding Rectangle
MPAF	The Mutual Promotion of Accompanying Frequency
OPTICS	Ordering points to identify the clustering structure
TLPC	Time and Location Potential Candidate
TC	Traveling Companion
TPAF	The Total Proportion of Accompanying Frequency

References

1. National Bureau Statistics of China, Statistical Communiqué of the People's Republic of China on National Economic and Social Development in 2019. Available online: http://www.stats.gov.cn/tjsj/zxfb/202002/t20200228_1728913.html (accessed on 28 February 2020).
2. Gao, Q.; Zhang, F.L.; Wang, R.J.; Zhou, F. Trajectory Big Data: A Review of Key Techologies in Data Processing. *Ruan Jian Xue Bao/J. Softw.* **2017**, *28*, 959–992. (In Chinese)
3. Elragal, A. Analysis of trajectory data in support of traffic management. *Lect. Notes Comput. Sci.* **2015**, *8557*, 174–188.
4. Enami, S.; Shiomoto, K. Spatio-temporal human mobility prediction based on trajectory data mining for resource management in mobile communication networks. In Proceedings of the IEEE International Conference on High Performance Switching and Routing, Xi'An, China, 26–29 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
5. Qin, T.; Shangguan, W.; Song, G.; Tang, J. Spatio-Temporal Routine Mining on Mobile Phone Data. *ACM Trans. Knowl. Discov. Data* **2018**, *12*, 56.1–56.24. [CrossRef]
6. Li, H.; Gou, Y. Mining Mobile Sensor Data for Social Behaviors. In Proceedings of the 2nd International Workshop on Social Sensing, Pittsburgh, PA, USA, 21 April 2017.
7. Chen, Y.; Crespi, N.; Ortiz, A.M.; Shu, L. Reality mining: A prediction algorithm for disease dynamics based on mobile big data. *Inf. Sci. Int. J.* **2017**, *379*, 82–93. [CrossRef]
8. Liu, X.; Tian, Y.; Zhang, X.; Wan, Z. Identification of Urban Functional Regions in Chengdu Based on Taxi Trajectory Time Series Data. *Int. J. Geo-Inf.* **2020**, *9*, 158. [CrossRef]
9. Zheng, Y. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 29:1–29:41. [CrossRef]
10. Tang, L.; Zheng, Y.; Yuan, J.; Han, J.; Leung, A.; Peng, W.; Porta, T.F.L. A framework of traveling companion discovery on trajectory data streams. *ACM Trans. Intell. Syst. Technol.* **2013**, *5*, 3:1–3:34. [CrossRef]
11. Zhu, M.L.; Liu, C.; Wang, X.-B.; Han, Y.-B. Approach to discover companion pattern based on anpr data stream. *Ruan Jian Xue Bao/J. Softw.* **2017**. (In Chinese)
12. Zhu, X.; Sun, T.; Yuan, H.; Hu, Z.; Miao, J. Exploring Group Movement Pattern through Cellular Data: A Case Study of Tourists in Hainan. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 74. [CrossRef]
13. Gudmundsson, J.; van Kreveld, M.J. Computing longest duration flocks in trajectory data. In Proceedings of the 14th ACM International Symposium on Geographic Information Systems, ACM-GIS 2006, Arlington, VA, USA, 10–11 November 2006; de By, R.A., Nittel, S., Eds.; ACM: New York, NY, USA, 2006; pp. 35–42.
14. Tang, L.; Zheng, Y.; Yuan, J.; Han, J.; Leung, A.; Hung, C.; Peng, W. On Discovery of Traveling Companions from Streaming Trajectories. In Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA, 1–5 April 2012; Kementsietsidis, A., Salles, M.A.V., Eds.; IEEE Computer Society: Washington, DC, USA, 2012; pp. 186–197.
15. Zhang, Y.; Ji, G.; Zhao, B.; Zhang, B. An Algorithm for Mining Gradual Moving Object Clusters Pattern From Trajectory Streams. *CMC-Comput. Mater. Contin.* **2019**, *59*, 885–901. [CrossRef]
16. Yao, R.; Wang, F.; Chen, S. TCoD: A Traveling Companion Discovery Method Based on Clustering and Association Analysis. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–7.
17. Mcinnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Sour. Softw.* **2017**, *2*, 205. [CrossRef]
18. Agrawal, R.; Faloutsos, C.; Swami, A.N. *Efficient Similarity Search in Sequence Databases*; Springer: Berlin/Heidelberg, Germany, 1993.

19. Faloutsos, C.; Ranganathan, M.; Manolopoulos, Y. Fast Subsequence Matching in Time-Series Databases. In Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, MN, USA, 24–27 May 1994; pp. 419–429.
20. Chan, K.; Fu, A.W. Efficient time series matching by wavelets. In Proceedings of the 15th International Conference on Data Engineering (Cat. No.99CB36337), Sydney, Australia, 23–26 March 1999; pp. 126–133.
21. Elnekave, S.; Last, M.; Maimon, O. Incremental Clustering of Mobile Objects. In Proceedings of the IEEE International Conference on Data Engineering Workshop, ICDE 2007, Istanbul, Turkey, 15–20 April 2007; pp. 585–592.
22. De Vries, G.K.D.; Van Someren, M. Clustering Vessel Trajectories with Alignment Kernels under Trajectory Compression. In Proceedings of the Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, 20–24 September 2010; Volume 6321, pp. 296–311.
23. Hightower, J.; Borriello, G. Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In Proceedings of the UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, 7–10 September 2004; Volume 3205, pp. 88–106.
24. Ester, M.; Kriegel, H.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial Databases with Noise. *Kdd* **1996**, *96*, 226–231.
25. Peipei, Z.; Qinghai, D.; Haibo, L.; Xinglin, H. Trajectory outlier detection based on DBSCAN clustering algorithm. *Infrared Laser Eng.* **2017**, *46*, 528001. [CrossRef]
26. Ankerst, M.; Breunig, M.M.; Kriegel, H.; Sander, J. OPTICS: Ordering points to identify the clustering structure. SIGMOD 1999. In Proceedings of the Proceedings ACM SIGMOD International Conference on Management of Data, Philadelphia, PA, USA, 1–3 June 1999; pp. 49–60.
27. Hinneburg, A.; Keim, D.A. A general approach to clustering in large databases with noise. *Knowl. Inf. Syst.* **2003**, *5*, 387–415. [CrossRef]
28. Hinneburg, A.; Gabriel, H.H. DENCLUE 2.0: Fast Clustering Based on Kernel Density Estimation. In Proceedings of the Advances in Intelligent Data Analysis VII, 7th International Symposium on Intelligent Data Analysis, IDA 2007, Ljubljana, Slovenia, 6–8 September 2007; Volume 4723, pp. 70–80.
29. Yang, Y.; Cai, J.; Yang, H.; Zhang, J.; Zhao, X. TAD: A trajectory clustering algorithm based on spatial-temporal density analysis. *Expert Syst. Appl.* **2020**, *139*, 112846. [CrossRef]
30. Gao, Y.; Zheng, B.; Chen, G.; Li, Q. Algorithms for constrainedk-nearest neighbor queries over moving object trajectories. *Geoinformatica* **2010**, *14*, 241–276. [CrossRef]
31. Gudmundsson, J.; Valladares, N. A GPU approach to subtrajectory clustering using the Fréchet distance. In Proceedings of the SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, 7–9 November 2012; pp. 259–268.
32. Deng, Z.; Hu, Y.; Zhu, M.; Huang, X.; Du, B. A scalable and fast OPTICS for clustering trajectory big data. *Clust. Comput.* **2015**, *18*, 549–562. [CrossRef]
33. Yuan, G.; Sun, P.; Zhao, J.; Li, D.; Wang, C. A review of moving object trajectory clustering algorithms. *Artif. Intell. Rev.* **2017**, *47*, 123–144. [CrossRef]
34. Xiao, X.; Zheng, Y.; Luo, Q.; Xie, X. Finding similar users using category-based location history. In Proceedings of the 18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2010, San Jose, CA, USA, 3–5 November 2010; pp. 442–445.
35. Ying, J.J.C.; Lee, W.C.; Weng, T.C.; Tseng, V.S. Semantic trajectory mining for location prediction. In Proceedings of the 19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011, Chicago, IL, USA, 1–4 November 2011; pp. 34–43.
36. Liu, S.L.Y.; Ni, L.M. Towards Mobility-based Clustering. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 25–28 July 2010; pp. 919–928.
37. Andrienko, G.; Andrienko, N.; Fuchs, G.; Garcia, J.M.C. Clustering Trajectories by Relevant Parts for Air Traffic Analysis. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 34–44. [CrossRef]
38. Olive, X.; Morio, J. Trajectory Clustering of Air Traffic Flows around Airports. *Aerosp. Sci. Technol.* **2019**, *84*, 776–781. [CrossRef]

39. Gudmundsson, J.; Kreveld, M.J.V. Computing longest duration flocks in trajectory data. In Proceedings of the 14th ACM International Symposium on Geographic Information Systems, ACM-GIS 2006, Arlington, VA, USA, 10–11 November 2006.
40. Jeung, H.; Yiu, M.L.; Zhou, X.; Jensen, C.S.; Shen, H.T. Discovery of convoys in trajectory databases. *Proc. VLDB Endow.* **2008**, *1*, 1068–1080. [CrossRef]
41. Zhenhui, L.; Bolin, D.; Han, J. Swarm: Mining relaxed temporal moving object clusters. *Proc. VLDB Endow.* **2010**, *3*, 723–734.
42. Kai, Z.; Yu, Z.; Yuan, N.J.; Shang, S. On Discovery of Gathering Patterns from Trajectories. In Proceedings of the 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, 8–12 April 2013; pp. 242–253.
43. Wang, Z.; He, S.Y.; Leung, Y. Applying mobile phone data to travel behaviour research: A literature review. *Travel Behav. Soc.* **2017**, *11*, 141–155. [CrossRef]
44. Fan, C.; Cavallaro, A. Detecting Group Interactions by Online Association of Trajectory Data. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 1754–1758.
45. Zhang, J.; Li, J.; Wang, S.; Liu, Z.; Yuan, Q.; Yang, F. On Retrieving Moving Objects Gathering Patterns from Trajectory Data via Spatio-temporal Graph. In Proceedings of the 2014 IEEE International Congress on Big Data, Anchorage, AK, USA, 27 June–2 July 2014; pp. 390–397.
46. Puntheeranurak, S.; Shein, T.T.; Imamura, M. Efficient Discovery of Traveling Companion from Evolving Trajectory Data Stream. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference, Tokyo, Japan, 23–27 July 2018; Volume 1, pp. 448–453.
47. Zhu, M.; Chen, L.; Wang, J.; Wang, X.; Han, Y. A Service-Friendly Approach to Discover Traveling Companions Based on ANPR Data Stream. In Proceedings of IEEE the International Conference on Services Computing, SCC 2016, San Francisco, CA, USA, 27 June–2 July 2016; pp. 171–178.
48. Xia, D.; Lu, X.; Li, H.; Wang, W.; Li, Y.; Zhang, Z. A MapReduce-Based Parallel Frequent Pattern Growth Algorithm for Spatiotemporal Association Analysis of Mobile Trajectory Big Data. *Complexity* **2018**, *2018*, 2818251:1–2818251:16. [CrossRef]
49. Wen-Bo, H.U.; Huang, W.; Guo-Chao, H.U. Trajectory Adjoint Pattern Analysis Based on OPTICS Clustering and Association Analysis. *Comput. Mod.* **2017**. (In Chinese) [CrossRef]
50. Albadwi, A.; Long, Z.; Zhang, Z.; Alhabib, M.; Alsabahi, K. A Novel Integrated Approach for Companion Vehicle Discovery Based on Frequent Itemset Mining on Spark. *Arab. J. Sci. Eng.* **2019**, *44*, 9517–9527. [CrossRef]
51. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of the International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 791–800.
52. Zheng, Y.; Li, Q.; Chen, Y.; Xie, X.; Ma, W. Understanding mobility based on GPS data. In Proceedings of the Ubicomp: Ubiquitous Computing, International Conference, Ubicomp, Seoul, Korea, 21–24 September 2008; pp. 312–321.
53. Zheng, Y.; Xie, X.; Ma, W. GeoLife: A Collaborative Social Networking Service among User, location and trajectory. *IEEE Data(base) Eng. Bull.* **2010**, *33*, 32–39.