INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(3):428-440, June, 2015.

Group Pattern Mining Algorithm of Moving Objects' Uncertain Trajectories

S. Wang, L. Wu, F. Zhou, C. Zheng, H. Wang

Shuang Wang, Lina Wu, Fuchai Zhou, Cuicui Zheng
Software College
Northeastern University
Shenyang, China
wangsh@mail.neu.edu.cn

Haibo Wang
H. John Heinz III College
Carnegie Mellon University
Pittsburgh, USA
haibowang@cmu.edu

Abstract: Uncertain is inherent in moving object trajectories due to measurement errors or time-discretized sampling. Unfortunately, most previous research on trajectory pattern mining did not consider the uncertainty of trajectory data. This paper focuses on the uncertain group pattern mining, which is to find the moving objects that travel together. A novel concept, uncertain group pattern, is proposed, and then a two-step approach is introduced to deal with it. In the first step, the uncertain objects' similarities are computed according to their expected distances at each timestamp, and then the objects are clustered according to their spatial proximity. In the second step, a new algorithm to efficiently mining the uncertain group patterns is designed which captures the moving objects that move within the same clusters for certain timestamps that are possibly nonconsecutive. However the search space of group pattern is huge. In order to improve the mining efficiency, some pruning strategies are proposed to greatly reduce the search space. Finally, the effectiveness of the proposed concepts and the efficiency of the approaches are validated by extensive experiments based on both real and synthetic trajectory datasets.

Keywords: probabilistic frequent group pattern, uncertain data, trajectory pattern mining, moving object.

1 Introduction

In recent years, with the development of various location-aware devices, such as RFID tags, cell phones, GPS navigation systems, and point of sale terminals, trajectory data has become ubiquitous in various domains [1]. Such data provides the opportunity of discovering usable knowledge about movement behavior, which fosters ranges of novel applications in human mobility understanding [2], smart transportation, urban planning [3], biological studies [4], environmental and sustain ability studies [5]. Ideally, a trajectory is represented as a sequence of locations, and each is being associated with a corresponding time stamp. However, this is a simplification that does not take into account the inherent uncertainties in such trajectories. For example, a position reported in a GPS signal usually implies a location point with an error range rather than an exact location. Moreover, as location privacy has become increasingly a concern, many locations are blurred when they are made public. Though the importance of mining uncertain data has been recognized [6–8], to the best of our knowledge there is little work in the trajectory pattern mining literature that studies its effect in the knowledge discovery process.

A useful data analysis task in movement is to find a group of moving objects that are traveling together for a certain time period. This concept, what we refer to as group patterns, can be defined in both spatial and temporal dimensions: (1) a group of moving objects should be geometrically close to each other; (2) they should be together for at least some minimum time duration. Although mining group patterns has been extensively studied on certain trajectory database, such as flock [9,10], convoy [11,12], swarm [1], traveling companion [13,14], and gathering [15], none of these work deal with the inherent uncertainty in trajectory database.

In this paper, we consider the problem of mining group patterns in the context of uncertain trajectory database. In contrast to previous work that did not model the trajectory with uncertainty, we represent a trajectory of a moving object as a sequence of reported locations at corresponding time stamps and a probability distribution function. The probability distribution function represents the probability distribution of possible locations of the trajectory at a given time instant. In particular, the probability distribution function can be discrete or continuous.

However, discovering the group patterns from uncertain trajectories is not an easy task because of adding a new dimension-probability. The challenges are two-fold: (1) The first challenge is how to design an appropriate distance function to measure the (dis)similarity between two uncertain trajectories. Particularly, an effective similarity metric should be able to conduct measurements in terms of different probability distributions, taking into account spatial distances, temporal intervals, as well as relevant probabilities. (2) Another challenge is how to find group patterns efficiently. Due to adding the probability in uncertain trajectory database, judging frequent group patterns requires to compute the frequent probabilities, and this is not a trivial task. Additionally, group patterns mining solutions may lead to an exponential number of results due to the downward closure property. So it is important to propose some effective pruning rules to reduce the massive search space and the number of probability computations.

Given the afore mentioned challenges, existing works for group patterns mining on exact trajectory data do not tackle the data uncertainty problem well. For instance, due to adding the probability, the concept and similarity distance metric function are not suitable for the uncertain environment. And also the traditional mining algorithms cannot be used directly to solve the uncertain group patterns mining. Our research makes the following contributions: (1) We propose a new problem, mining group patterns over uncertain trajectory data. (2) We introduce a novel and adaptive metric to measure the dissimilarity between two uncertain trajectories. (3) We design an efficient algorithm to discover all uncertain group patterns. In addition, we propose several pruning techniques to reduce search space and avoid redundant computation. (4) Extensive experiments demonstrate that the effectiveness and efficiency of our algorithm.

The remaining of the paper is organized as follows. Section 2 discusses the related work. We introduce the distance similarity function of two uncertain trajectories in Sections 3. In Section 4, we give the definitions of uncertain group patterns, and then introduce our efficient algorithm based on breath-first search strategy. Experiments testing effectiveness and efficiency are shown in Section 5. Finally, our study is concluded in Section 6.

2 Related work

Group pattern mining over exact trajectory data. Group pattern mining, which is to discover a group of objects that move together for a certain time period, is an important data analysis task for moving object trajectories. The research mainly included flock, convoy, and swarm pattern mining. The concepts of group patterns can be distinguished based on how the group is defined and whether they require the time periods to be consecutive. Specifically, a flock [9,10] is a group of objects that travel together within a disc of some user-specified size for at least k consecutive time stamps. A major drawback is that a circular shape may not reflect the natural group in reality, which may result in the so-called lossy-flock problem [11]. To avoid rigid restrictions on the sizes and shapes of the group patterns, the convoy is proposed to capture generic trajectory pattern of any shape and extent by employing the density-based clustering. Instead of using a disc, a convoy requires a group of objects to be density-connected to each other during k consecutive time points. While both flock and convoy have strict requirement on consecutive time period, the rigid definition of flock and convoy sometimes makes it not practical to find potentially interesting patterns. In contrast to flock, convoy, Li et al [1] proposed a more general type of trajectory pattern, called swarm, which is a cluster of objects lasting for at least k (possibly non-consecutive) timestamps. Because this is more realistic as different people may temporarily leave the cluster at some snapshots, in this paper our uncertain group pattern definition is based on swarm and is extended to uncertain data model. Although there are a lot of works on mining group patterns, all these algorithms are designed for exact trajectory data and cannot be extended to uncertain trajectory data directly.

Pattern mining on uncertain data. Another set of researches related with our work are pattern mining over uncertain data. Existing work on mining frequent itemsets from uncertain databases falls into two categories based on the definition of a frequent itemset: expected support-based frequent itemset and probabilistic frequent itemset. Both definitions consider the frequency (support) of an itemset as a discrete random variable. The former employs the expectation of the support as the measurement. That is, an itemset is frequent only if the expected support of the itemset is no less than a specified minimum expected support. The latter uses the frequentness probability as the measurement, which is the probability that an itemset appears no less than a specified minimum support times. Then, an itemset is frequent only if its frequentness probability is no less than a specified minimum probability threshold. However, the use of expected support may lead to the loss of important patterns. Thus, the use of a probabilistic frequentness measure has been more popular recently. A recent survey for comparing these two measures and analyzing their relationships is given in [16].For the problem of uncertain sequence pattern mining, some initial research has been undertaken. For example, the expected support-based frequent sequential pattern mining has been studied in [17]. In contrast, Zhao et al. [18] proposed to mine probabilistic frequent sequential patterns according to the frequentness probability. However, all of these works only considered the simple value-based data type, and are not suitable for the complex data type-trajectory data, which contains both spatial and temporal information. Our paper is the first work to solve the group patterns mining problem on uncertain trajectory data.

3 Similarity metric of two uncertain locations

In this section, we present our method of computing the spatial proximity of objects with uncertain locations. We formalize the model of uncertain trajectories in Section 3.1 and give the similarity distance function in Section 3.2. Frequently used symbols throughout this paper are summarized in Table 1.

3.1 Uncertain trajectory model

Let $T_{DB} = \{t_1, t_2, \cdots t_n\}$ be a linearly ordered list of n timestamps. Let O_{DB}=$\{o_1, o_2, \cdots, o_m\}$ be a collection of m moving objects that appear in T_{DB}. The locations of objects o observed at timestamps T_{DB} are uncertain. We first give the definition of a certain trajectory.

Definition 1. (Certain Trajectory). A certain trajectory T_r is represented as a sequence of points $\{(x_1, y_1, t_1), (x_2, y_2, t_2), \cdots, (x_n, y_n, t_n)\}$ $(t_1 < t_2 < \cdots < t_n)$, where n is the number of points in the trajectory and (x_i, y_i) are the coordinates of the ith point at timestamp t_i.

In some uncertain trajectories' studies [6,7], they commonly used probability density function (*pdf*) to represent the uncertainty. Unfortunately, the exact probability distribution is not easily computed. So in this paper, we adopt the expectation and variance to model the uncertain trajectory. We can use the Evolving Density Estimator [8] to compute the mean value u and standard deviation v of an object's position at each time. Based on the definition of certain trajectories, we have the definition of an uncertain trajectory as follows:

Definition 2. (Uncertain Trajectory). An uncertain trajectory UTr is a sequence of random variables, and all the random variables at different timestamps are assumed to be independent. An Utr is represented as $\{(x_1, y_1, u_1, v_1, t_1), (x_2, y_2, u_2, v_2, t_2), ..., (x_n, y_n, u_n, v_n, t_n)\}$ $(t_1 < t_2 < ... < t_n)$, where u_i and v_i are the expectation and variance of Utr at timestamp t_i.

Table 1: Summary of the use of notations

Symbol	Illustration
O_{DB}	Moving object set, $O_{DB} = \{o_1, o_2, o_3, ..., o_m\}$
o_j	The jth object
O	Objects subset,$O \subseteq O_{DB}$
T_{DB}	Timestamp set, $T_{DB} = \{t_1, t_2, ..., t_n\}$
t_i	Timestamp t_i
T	Time subset,$T \subseteq T_{DB}$
UTr_i	The ith object's uncertain trajectory
$UTr_i(j)$	The jth sample point of the ith uncertain trajectory
$dist_i(UTr_a, UTr_b)$	The distance of the two objects at time t_i
C_{DB}	The database of clusters using FCM algorithm
C_{t_i}	The clusters at time t_i
c_{t_ij}	The jth cluster of C_{t_i}
$C_{t_i}(o_j)$	The set of clusters that object o_j is in at timestamp t_i
$p(o_j \in c_{t_ij})$	The probability of object o_j belonging to an cluster c_{t_ij}
$p(O \in c_{t_ij})$	The probability of objects O belonging to an cluster c_{t_ij}
$p(O \in C_{t_i})$	The probability of objects O belonging to the clusters C_{t_i}
min_o	The minimum objects threshold
min_t	The minimum timestamps threshold
$minprob$	The minimum probability threshold
$\Pr(\text{support}(O, T) \geq min_t)$	The probability that objects of O are in the same cluster for at least min_t timestamps

3.2 Expected distance function

Before giving the definition of the uncertain group pattern, we introduce a novel and adaptive metric EE-distance (expected Euclidean distance) for measuring the similarity between two uncertain trajectories.

Trajectory similarity is commonly estimated using trajectory distance measures, such as the Euclidean distance, the dynamic time warping (DTW) distance, the principal component analysis (PCA) distance, the edit distance with real penalty (ERP), and the longest common subsequence (LCSS) distance. However, there is no trajectory similarity measure that can beat all the others in every circumstance as introduced in [19]. In this paper we adopt the Euclidean distance due to its simplicity in implementation and low computation complexity. Next, we will show how to compute the uncertain instant distance between two trajectories based on the expected Euclidean distance.

Definition 3. (Uncertain Instant Distance). We can treat the distance between two uncertain trajectories at timestamp t_i as a square sum of the sample points, as shown in equation(1), $UTr_a(i)$ is the sample point of uncertain trajectory UTr_a at timestamp t_i, the same as $UTr_b(i)$.

$$dist_i(UTr_a, UTr_b) = (UTr_a(i) - UTr_b(i))^2 \tag{1}$$

$UTr_a(i)$ and $UTr_b(i)$ are the independent random variables, so $dist_i(UTr_a, UTr_b)$ is also the random variable. The expectation of the random variable $dist_i(UTr_a, UTr_b)$ can be computed in equation (2).

$$\begin{aligned} E((UTr_a(i) - UTr_b(i))^2) &= E(UTr_a(i))^2 + var(UTr_a(i)) \\ &+E(UTr_b(i))^2 + var(UTr_b(i)) - 2E(UTr_a(i)) \cdot E(UTr_b(i)) \end{aligned} \tag{2}$$

According to the equation (2), we can easily compute the expected distance of two uncertain locations in $O(1)$ time complexity. Unlike other works requiring exact probability distribution function, our distance formulation is statistically sound and only requires knowledge of the general characteristics of the data distribution, namely, its mean and variance.

4 Mining Probabilistic Frequent Group Patterns

In this section, we first provide definitions of our probabilistic frequent group patterns, and then show how to find all probabilistic frequent group patterns in an uncertain trajectory database.

4.1 Probabilistic frequent group patterns definition

Definition 4. (frequent group pattern)[1]. A pair (O, T) is said to be a group pattern if all objects in O are in the same cluster at any timestamp in T. Specifically, given two minimum thresholds min_o and min_t, for (O, T) to be a frequent group pattern, where $O = \{o_{i_1}, o_{i_2}, ..., o_{i_l}\} \subseteq O_{DB}$ and $T \subseteq T_{DB}$, it needs to satisfy two requirements: (1) there should be at least min_o objects; (2) objects in O are in the same cluster for at least min_t timestamps.

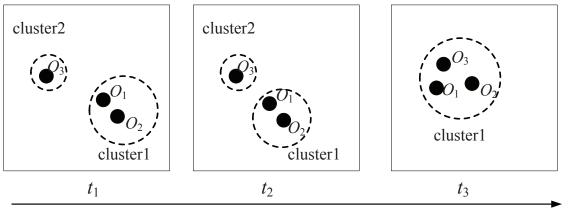


Figure 1: Object clusters at each timestamp in certain database

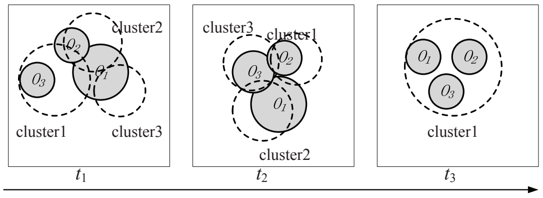


Figure 2: Object clusters at each timestamp in uncertain database

Fig.1 shows an example. There are 3 objects and 3 timestamps, $O_{DB} = \{o_1, o_2, o_3\}$, $T_{DB} = \{t_1, t_2, t_3\}$. Each sub-figure is a snapshot of object clusters at each timestamp. It is easy to see that o_1 and o_2 travel together for most of the time. Given min_o=2 and $min_t = 2$, there are only one frequent group pattern:$(\{o_1, o_2\}, \{t_1, t_2, t_3\})$.

As shown in Fig.2, in the uncertain scenario, the object's location is not an exact position, but an uncertain range. Thus, it is more realistic to cluster the object in different clusters than to a single cluster. Based on this point, we can apply a fuzzy clustering algorithm (e.g.FCM [20,21]) to create the cluster at timestamp t_i. Fuzzy clustering would assign each object a Ą°degree of belongingnessĄą(belongingness probability) for each cluster. A simple example of clusters is given in table 2.

Table 2: An example of uncertain clusters.

Time	Uncertain clusters
1	$c_{11} = \{o_1 : 0.3; o_2 : 0.5; o_3 : 1.0\}, c_{12} = \{o_1 : 0.5; o_2 : 0.5\}, c_{13} = \{o_1 : 0.2\}$
2	$c_{21} = \{o_1 : 0.3, o_2 : 0.8\}, c_{22} = \{o_1 : 0.7, o_3 : 0.2\}, c_{23} = \{o_2 : 0.2, o_3 : 0.8\}$
3	$c_{31} = \{o_1 : 1.0, o_2 : 1.0, o_3 : 1.0\}$

An object $o_j \in O_{DB}$ with an uncertain location at timestamp $t \in T_{DB}$ has a belongingness probability of cluster $c \in C_{DB}$, the probability is denoted as $p(o_j \in c)$, where $p(o_j \in c) \in [0, 1]$. At timestamp t_i, an object could belong to more than one cluster, we use $C_{t_i}(o_j)$ to denote the set of clusters that object o_j is in, the total belongingness probability of object o_j in different clusters is one $p(C_{t_i}(o_j)) = \sum_{c \in C_{t_i}} p(o_j \in c) = 1$. In addition, for a given objectset O, we write $C_{t_i}(O) = \bigcap_{o_j \in O} C_{t_i}(o_j)$, which means O occurs at timestamp t_i in the same cluster. We assume that different objects and different timestamps are mutually independent, i.e., the belongingness probability of an object has no effect on those of other objects. So, the probability of the object set O belonging to a cluster c could be computed in equation (3), the probability of object set O in the same clusters at timestamp t_i could be computed in equation (4). To make our framework more general, we take clustering as a preprocessing step. The details are given in example 1.

$$p(O \in c_{t_ij}) = \prod\nolimits_{o_j \in O} p(o_j \in c_{t_ij}) \tag{3}$$

$$p(C_{t_i}(O)) = \sum\nolimits_{c_{t_ij} \in C_{t_i}} p(c_{t_ij} \in C_{t_i}) \tag{4}$$

Example 1: For $T_{DB} = \{t_1, t_2\}, O_{DB} = \{o_1, o_2, o_3\}$ in table 2, at timestamp $t_1 : c_{11} = \{o_1 : 0.3; o_2 : 0.5; o_3 : 0.8\}$; $c_{12} = \{o_1 : 0.5; o_2 : 0.5; o_3 : 0.2\}$; $c_{13} = \{o_1 : 0.2\}$, $O = \{o_1; o_2\}$, then $p(o_1 \in c_{11}) = 0.3$, $p(o_2 \in c_{11}) = 0.5, p(O \in c_{11}) = 0.3 * 0.5 = 0.15$,and $p(C_{t_1}(O)) = p(O \in c_{11}) + p(O \in c_{12})$=0.15 + 0.25 = 0.4. In the same way, at timestamp t_2 and t_3, $p(C_{t_2}(O))$=$p(O \in c_{21})$=0.24, $p(C_{t_3}(O)) = 1.0$.

In the uncertain scenario, the number of timestamps that objects O in the same cluster at the timestamps T, denoted as *support*(O, T), is no longer certain. Instead co-occurrence is described by a discrete probability distribution function. As shown in example 1, the probability of the objects $O = \{o_1; o_2\}$ occurring in the same cluster at timestamp t_1 , t_2 and t_3 is 0.4, 0.24 and 1.0 respectively. The frequency distribution is described in table 3. For example, the probability that O occurring in the same cluster at least two timestamps is 0.054. The definition of probabilistic frequent group pattern should consider this uncertainty.

Table 3: Frequency distribution of $O = \{o_1; o_2\}$ occurring at timestamps $T = \{t_1, t_2, t_3\}$

Frequency of timestamp	≥ 0	≥ 1	≥ 2	≥ 3
Probability	1.0	1.0	0.054	0.096

Definition 5. (Probabilistic frequent group pattern). A pair (O, T) is an probabilistic frequent group pattern $iff(O, T)$ is a frequent group pattern and Pr(support$(O, T) \geq min_t) \geq$ *minprob*, *minprob* is the probability threshold.

For a given group pattern (O,T), the frequentness probability, $\Pr(\text{support}(O,T) \geq min_t)$, which is interpreted as the probability that objects of O are in the same cluster for at least min_t timestamps. Under the definition of the probabilistic frequent group pattern, it is critical to compute the frequent probability of a group pattern efficiently. The frequentness probability $\Pr(\text{support}(O,T) \geq \min_t)$, could be computed by means of the paradigm of dynamic programming shown in equation(5) [22].

$$P_{\geq i,j}^{(O,T)} = \begin{cases} p_{\geq i-1,j-1}^{(O,T)} * p(O \in C_{t_j}) + p_{\geq i,j-1}^{(O,T)} * (1 - p(O \in C_{t_j})) & O \in C_{t_j} \\ p_{\geq i,j-1}^{(O,T)} & O \notin C_{t_j} \end{cases} \tag{5}$$

For the sake of the following discussion, we define that $P_{\geq i,j}^{(O,T)}$ denotes the probability that objects O appears at least i timestamps among the first j timestamps in the given time set T. The dynamic programming approach is to split the problem of computing $P_{\geq i,j}^{(O,T)}$ at the first j timestamps into sub-problems of computing the frequentness probabilities at the first $j-1$ timestamps. Our goal is to find all probabilistic frequent group patterns. Note that even though our problem is defined in the similar form of uncertain frequent pattern mining [22], none of previous work in uncertain frequent pattern mining area can solve exactly our problem. Because FP mining problem takes transactions as input, group pattern discovery takes clusters at each timestamp as input. If we treat each timestamp as one transaction, each "transaction" is a collection of "itemsets"rather than just one itemset.Therefore, there is no trivial transformation of FP mining problem to group pattern mining problem. The difference demands new techniques to specifically solve our problem.

4.2 Uncertain trajectory data preprocessing

When mining probabilistic frequent group patterns, we assume that each moving object has a reported location at each timestamp. However, in most real cases, the raw data collected is not as ideal as we expected. The sampling timestamps for different moving objects are usually not synchronized. Even though many complicated interpolation methods could be used to fill in the missing data with higher precision, any interpolation is only a guessing of real positions. In this paper, we use linear interpolation to obtain possible position and statistical value at an arbitrary time between two consecutive sample times. We define the $o(t) = \{x_t, y_t, u_t, v_t\}$ as an uncertain object between two consecutive sample timestamp t_i and t_{i+1} , the value of $o(t)$ can be computed in equation(6).

$$\begin{aligned} x_t = x_i + (x_{i+1} - x_i) \cdot \frac{t - t_i}{t_{i+1} - t_i}, y_t = y_i + (y_{i+1} - y_i) \cdot \frac{t - t_i}{t_{i+1} - t_i} \\ u_t = u_i + (u_{i+1} - u_i) \cdot \frac{t - t_i}{t_{i+1} - t_i}, v_t = v_i + (v_{i+1} - v_i) \cdot \frac{t - t_i}{t_{i+1} - t_i} \end{aligned} \tag{6}$$

In order to make our approach has extensibility, we treat the computation of the spatial proximity and construction the clusters of objects with uncertain locations as a preprocessing step. In this way, spatial proximity of objects and clustering methods can be flexible and application-dependent. We only require a set of clusters as input for each timestamp, where each object is associated with a belongingness probability that specifies the confidence the object is in a cluster at a given timestamp. The preprocessing algorithm is as follows.

```
Algorithm Preprocessing.Preprocessing algorithm
input: uncertain trajectory database $UTD$
output: uncertain clusters database $C_{DB}$
1.For each object $o_i$
2.   For each timestamp $t_j$
3.      if (no sample value)
4.         interpolate the value;
5.      compute the expected distance $dist_j(UTr_i, UTr_k)$ for each object $o_k$ in $O_{DB}$;
6.use FCM cluster algorithm based on the expected similarity;
7.output uncertain clusters database $C_{DB}$;
```

For each object o_i, we compute the expected distance of every object o_k at time t_j (line 1-5), then we uses the FCM mining algorithm to cluster the uncertain objects at each timestamps(line 6). Finally, we get the uncertain clusters database C_{DB}(line 7).

4.3 Pruning Techniques

Although we use the dynamic programming to compute the frequent probability of a group pattern, the cost of computation is still high, the time complexity is $O(n^2)$, n is the number of timestamps in T. Fortunately, we find two efficient pruning rules based on properties of the frequent probability.

Pruning rule 1(count prune): Given a group pattern (O, T), if $cnt(O, T) < \min_t$, then (O, T) is not a probabilistic frequent group pattern,$cnt(O, T)$ is the numbers of timestamps in T that O in the same clusters.

Pruning rule 2 (expected prune): Given a group pattern (O, T), $e\sup(O, T)$ is the expected support of a group pattern (O, T), is defined as the sum of the probabilities that objects O occurring in the same cluster in each of the timestamps in T, $e\sup(O, T) = \sum\limits_{t_j \in T} p(O \in C_{t_i})$. If $\lambda^- <\ minprob$, then (O, T) is not a probabilistic frequent group pattern.

$$\lambda^- = \begin{cases} \dfrac{-(\min_t - 1 - e\sup(O,T)^2)}{4e\sup(O,T)} & 0 < \dfrac{\min_t - 1 - e\sup(O,T)}{e\sup(O,T)} < 2e - 1 \\ 2^{1+e\sup(O,T)-\min_t} & \dfrac{\min_t - 1 - e\sup(O,T)}{e\sup(O,T)} \geq 2e - 1 \\ \dfrac{e\sup(O,T)}{\min_t} & other \end{cases} \tag{7}$$

Pruning rules could be used to prune infrequent group pattern before computed the exact frequent group probability. The running time of computing the $cnt(O, T)$ and $e\sup(O, T)$ is $O(n)$, but $O(n^2)$ for the exact probability $\Pr(\text{support}(O, T) \geq min_t)$. So pruning rules can significantly improve the running time of mining algorithm.

4.4 Mining algorithm of probabilistic frequent group pattern

Traditional frequent itemset mining is based on support pruning by exploiting the anti-monotonic property of support. In uncertain databases, recall that support is defined by a probability distribution and that we mine group patterns according to their frequentness probability. It turns out that the frequentness probability is also anti-monotonic.

Theorem 6. $\forall O \subseteq O(prime), \Pr(\text{support}(O, T) \geq min_t) \geq \Pr(\text{support}(O(prime), T) \geq min_t)$, *all subsets of a probabilistic frequent group patterns are also probabilistic frequent group patterns.*

We can use the Theorem 1 to prune the search space for probabilistic frequent group pattern. That is, if a group pattern (O, T) is not a probabilistic frequent group pattern,i.e.$\Pr(\text{support}(O, T) \geq min_t) < minprob$, then all patterns $O(prime) \supset O$ cannot be probabilistic frequent group patterns.

We propose a probabilistic frequent group patterns mining approach based on the Apriori algorithm. Like Apriori, our method iteratively generates the probabilistic frequent group patterns using a bottom-up strategy. Each iteration is performed in two steps, a join step for generating new candidates and a pruning step for calculating the frequentness probabilities and extracting the probabilistic frequent group patterns from the candidates. The pruned candidates are, in turn, used to generate candidates in the next iteration. Theorem 1 is exploited in the join step to limit the candidates generated and in the pruning step to remove group patterns that need not be expanded. In the join step, our algorithm adopts the breadth-first implementation. Because the depth-first strategy does not fully use the downward closure of the probabilistic support,this is due to the fact that the depth-first implementation does not know all frequent k-objectset before considering the $(k + 1)$-objectset. This may lead to a bigger search space. The detailed steps of our algorithm to compute the probabilistic frequent group pattern is listed in Algorithm PFGPM.

We first select frequent 1-object sets (line 1), and then recursively generate candidate $(k+1)$-objectset from k-objectset (line 2-9). At each iteration, only the frequent k-objectsets are extended (Apriori property, line 3-4). We first scan the database to calculate the expected support of each candidate, and use the pruning rules to prune candidate (line 5), if not be pruned, compute the frequentness probability(line 6-8). Next, we output those patterns satisfying $|O| \geq min_o$ and probabilistic support (line 9).

Algorithm PFGPM. Probabilistic frequent group pattern mining algorithm

```
input: uncertain clusters database $C_{DB}$
output: probabilistic frequent group patterns
1.Apply the pruning rules first, then calculate the frequentness probability for each 1-objectset,find all the frequent 1-objectset called $Cand$, and sort them in alphabetic order;
2.$K = 2$
3.For each $O$ in $Cand$
4.    extend $O$ using a breadth-first search like strategy to its supersets with $O$ as prefix,
      denoted $O(prime)$;
5.    Using pruning rule 1 and 2 for $O(prime)$;
6.    if($O(prime)$ not pruned)
7.        Compute the frequent group probability($fgp$) as shown in equation(5);
8.        If($fgp \geq minprob$)
9.            OUTPUT $(O(prime), fgp)$, if$|O(prime)| \geq min_o$
10.$K = K + 1$;
11.go to 3;
```

5 Experiments

5.1 Datasets

Real data. A truck dataset (http://www.chorochronos.org/) recording 50 trucks delivering concrete to construction sites around Athens over 33 days and consisting of 276 trajectories. To increase the size of moving objects, we considered each distinct trajectory as the ID of an object, yielding 276 trucks with 2449 timestamps. In our experiments, we consider only the first 128 positions of each trajectory. We normalize the positions of trajectories into a unit space. The probabilities of each trajectory were assigned according to two different distributions: (1) Each certain position p was assigned a probability according to a uniform distribution in the range of (0.5, 1.0]. (2) Each position was assigned a normal distribution N(0.5,0.2) in the range of [0, 1.0].

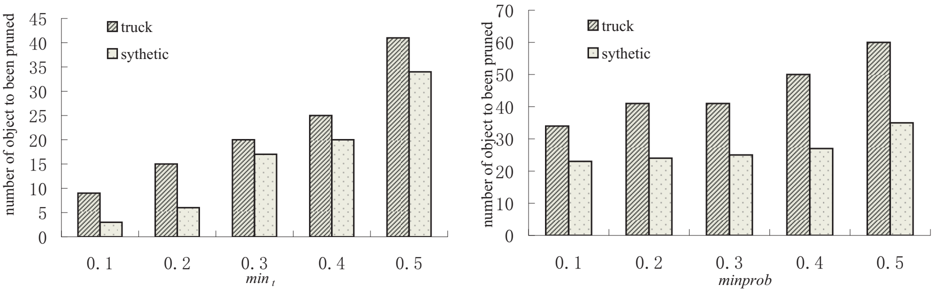
Synthetic data. We first generate two certain trajectories based on a custom GSTD data generator [23]. Each dataset contains 10,000 uncertain trajectories with the same length 128. We then convert these certain trajectories to uncertain trajectories in the way described above.

The fuzzy clusters of teach timestamp are obtained by the fuzzy c-means clustering algorithm [20] with $m = 2$ and $EPS = 0.01$, where m is the weighting exponent and EPS is the termination criterion. Each object is assigned a belongingness probability by the fuzzy clustering algorithm. The default values of min_o=10, min_t=0.5,(half of the overlapping time span), $minprob$=0.3.

5.2 Performance evaluation

No previous technique addresses probabilistic frequent group patterns mining for uncertain trajectories. We compare our PFGPM approach with an alternative, PFGPM-NP, that does not use any pruning rules.We present the experimental results in this section. All the experiments are run on a desktop PC with a 2.66GHz CPU and 4GB RAM.

Effect of punning rules: We use two dataset with varying parameter settings to test the performance of the pruning rules. (1) Synthetic dataset. We extract the first 274 trajectories of synthetic dataset; (2) Truck datasets. Each position p in two datasets was assigned a probability according to a uniform distribution approach.As shown in Fig.3, the pruning works well for skewed dataset (truck dataset). The reason is straightforward: the more skewed the data, the higher the number of objects is infrequent and, thus, cannot be pruned. Regarding the effect of parameters, the larger min_t and $minprob$ are, the more objects will be pruned. However,min_t has a more significant influence than $minprob$, that is, pruning rule 1 has more strong power than pruning rule 2.



(a) varying min_t (b) varying $minprob$

Figure 3: Pruning power

Effect of $|O_{DB}|$ and $|T_{DB}|$. We ran out our algorithm in synthetic data set. Fig.4 and Fig.5 depict the running time when varying $|O_{DB}|$ and $|T_{DB}|$ respectively. In both figures, PFGPM-NP is much slower than PFGPM. Furthermore, PFGPM-NP is usually 5 times slower than PFGPM. Comparing Fig.4 and Fig.5, we can see that PFGPM is more sensitive to the change of $|O_{DB}|$. This is because its search space is enlarged with larger $|O_{DB}|$, whereas the change of $|T_{DB}|$ increases the computing time of frequent probability, which does not directly affect the running time of PFGPM.

Effect of min_o. Fig.6 shows the running time w.r.t. min_o. With the increasing of min_o, the running time will decrease because less group patterns would meet the requirement. Besides, it is

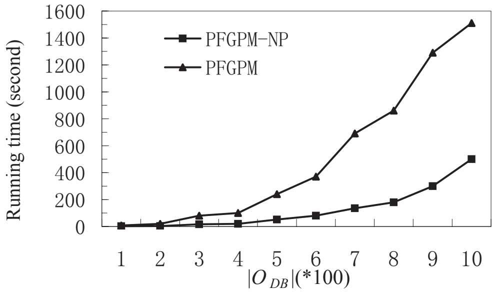


Figure 4: Running time with varying $|O_{DB}|$

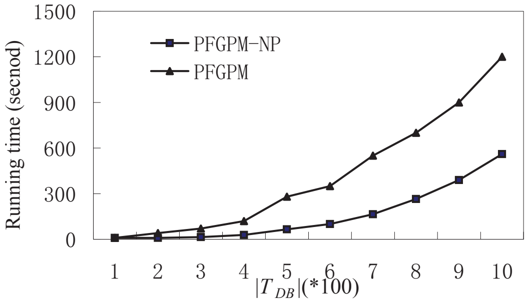
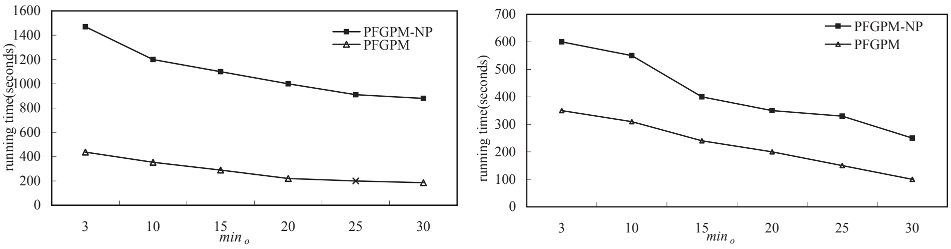


Figure 5: Running time with varying $|T_{DB}|$

obvious that PFGPM-NP takes much longer time than PFGPM. The reason is that PFGPM-NP does not use any pruning rules to find the frequent group patterns, this leads to more redundant computation for frequent probability and larger search space.

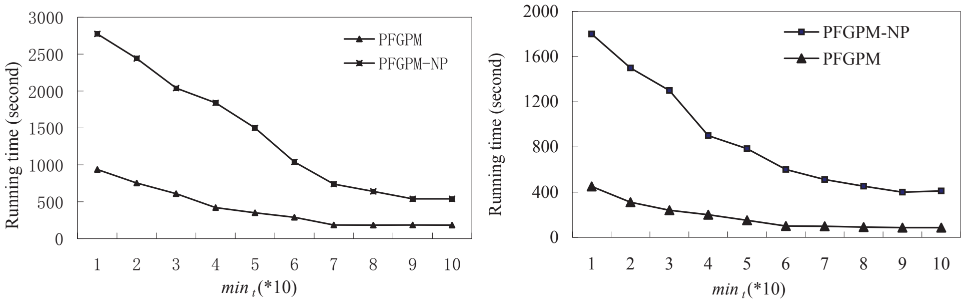


(a) synthetic data set

(b) truck data set

Figure 6: Object cluster at each timestamp in certain trajectory database

Effect of min_t. Fig.7 shows the influence of min_t on the runtime when using different data sets. When min_t decreases, we observe that the running time of all the algorithms goes up due to the number of probabilistic frequent group patterns increases. However, we find that the growth speed of PFGPM is low due to all pruning methods it employed.

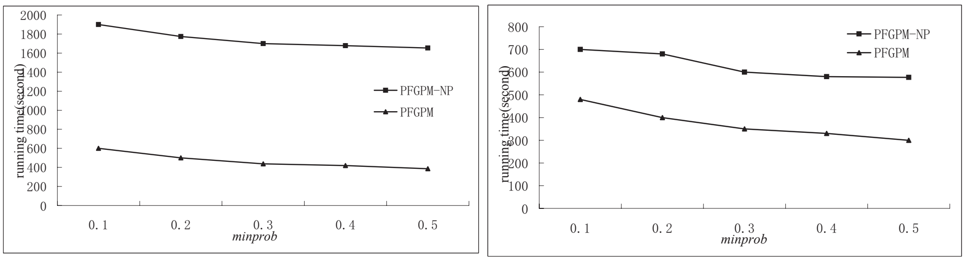


(a) synthetic data set

(b) truck data set

Figure 7: Running time with varying min_t

Effect of *minprob*. Finally, we test the running time of two compared algorithms with varying the probabilistic frequent threshold, *minprob* in two different datasets. In Fig.8, we can obverse that PFGPM is always faster than PFGPM-NP algorithm. With regards to the change of *minprob*, the running time of all algorithms remains approximately the same. Thus, we can discover that the influence of probabilistic frequent threshold will be smaller than that of min_t to the total running time.



(a) Synthetic data set

(b) truck data set

Figure 8: running time with varying *minprob*

6 Conclusion

In this paper, we have formulated and studied the problem of mining probabilistic frequent group patterns in uncertain trajectory database. We introduce a novel notion expected distance to measure the dissimilarity between uncertain locations. In order to mining such group patterns efficiently, we proposed several pruning techniques to reduce search space and to avoid many complicated computations. We further designed a Apriori-based algorithms using breadth-first implementations for efficient enumeration of all probabilistic frequent group patterns from uncertain data. Extensive experimental results show the effectiveness and efficiency of the mining algorithm. In our further study, we aim to extend our current approach to be able to handle more complex patterns for the trajectory data.

Acknowledgement

This research is partially supported by Fundamental Research Funds for the Central Universities (Grant No. 130317003), the National Natural Science Foundation of China (Grant No. 61440014, Grant No. 61300196), the Natural Science and Technology Major Project (Grant No. 2103ZX03002006), and the Liaoning Province Science and Technology Project (Grant No. 2013217004).

Bibliography

[1] Z. Li, B. Ding, et al, Swarm: Mining relaxed temporal moving object clusters, *the VLDB Endowment*, 3(1-2):723-734.

[2] D. Wegener, D. Hecker, et al, Parallelization of R-programs with GridR in a GPS-trajectory mining application, *1st Ubiquitous Knowledge Discovery Workshop on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, (2008).

[3] X. Li, J. Han, et al, Traffic density-based discovery of hot routes in road networks, *the 10th International Symposium on Spatial and Temporal Databases*, 441-459, (2007).

[4] Z.Li, J.G. Lee, et al, Incremental Clustering for Trajectories, *the 15th Database Systems for Advanced Applications*, 32-46, (2010).

[5] X. Li, J. Han, et al, Motion-alert: automatic anomaly detection in massive moving objects, *the 4th IEEE International Conference on Intelligence and Security Informatics*, 166-177.

[6] N. Pelekis, I. Kopanakis, et al, Clustering uncertain trajectories, *Knowledge and Information Systems*, 28(1): 117-147.

[7] M. Chunyang, L. Hua , et al, KSQ: Top-k Similarity Query on Uncertain Trajectories,*Knowledge and Data Engineering, IEEE Transactions*, 25(9): 2049-2062.

[8] J. Hoyoung, Managing Evolving Uncertainty in Trajectory Databases, *IEEE Transactions on Knowledge and Data Engineering*, 26(7): 1692-1705.

[9] J. Gudmundsson, M. V. Kreveld, Computing longest duration flocks in trajectory data, *the 14th annual ACM international symposium on Advances in geographic information systems*, 35-42, (2006).

[10] J. Gudmundsson, M. V. Kreveld, et al, Efficient detection of motion patterns in spatio-temporal data sets, *the 12th annual ACM international symposium on Advances in geographic information systems*, 250-257, (2004).

[11] H. Jeune, M. Yiu, et al, Discovery of convoys in trajectory databases, *the VLDB Endowment*, 1(1):1068-1080.

[12] H. Jeune, H. Shen, et al, Convoy queries in spatio-temporal databases, *the 24th International Conference on Data Engineering*, 1457-1459, (2008).

[13] L.A. Tang, Y. Zheng, et al, A Framework of Traveling Companion Discovery on Trajectory Data Streams, *ACM Transaction on Intelligent Systems and Technology*, 5(1):3.

[14] L.A. Tang, Y. Zheng, et al, Discovery of Traveling Companions from Streaming Trajectories, *the 28th IEEE International conference on Data Engineering*, 186-197, (2012).

[15] K. Zheng, Y. Zheng, et al, Online Discovery of Gathering Patterns over Trajectories, *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1974-1988.

[16] Y. Tong, L. Chen, et al., Mining frequent itemsets over uncertain databases, *VLDB Endowment*, 5(11): 1650-1661.

[17] M. Muzammal, R. Raman, Mining sequential patterns from probabilistic databases, *the 15th Pacific-Asia conference*, 210-221, (2011).

[18] Z. Zhao, D. Yan, et al, Mining probabilistically frequent sequential patterns in uncertain databases, *the 15th International Conference on Extending Database Technology*, 74-85,(2012).

[19] H. Wang, H. Su, K. et al, An Effectiveness Study on Trajectory Similarity Measures, *the 24th Australasian Database Conference*, 13-22, (2013).

[20] J. Bezdek, R. Ehrlich, et al, FCM: The fuzzy c-means clustering algorithm, *Computers Geosciences*, 10(2):191-203.

[21] C. Hwang, F. C.-H. Rhee, Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means, *Fuzzy Systems*, 15(1):107-120.

[22] T. Bernecker, H.-P. Kriegel, et al, Probabilistic frequent itemset mining in uncertain databases, *the 15th ACM SIGKDD on Knowledge discovery and data mining*, 119-128,(2009).

[23] Y. Theodoridis, J. R. O. Silva, et al, On the generation of spatiotemporal datasets, *the 6th International Symposium on Advances in Spatial Databases*, 147-164, (1999).