IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 24, 2017, accepted November 13, 2017, date of publication December 15, 2017,
date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2778690

Trajectory Mining Using Uncertain Sensor Data

MUHAMMAD MUZAMMAL[1,2], MONEEB GOHAR[2], ARIF UR RAHMAN[2,3], QIANG QU[1,4], AWAIS AHMAD[5], AND GWANGGIL JEON[6]

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518118, China
[2]Department of Computer Science, Bahria University, Islamabad 44000, Pakistan
[3]Faculty of Computer Science, Free University of Bolzano, 39100 Bolzano, Italy
[4]MOE Key Laboratory of Machine Perception, Peking University, Beijing 100080, China
[5]Department of Information and Communication Engineering, Yeungnam University, Gyeongbuk 38541, South Korea
[6]Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, South Korea

Corresponding authors: Qiang Qu (qiang@siat.ac.cn) and Awais Ahmad (aahmad.marwat@gmail.com)

This work was supported in part by the CAS Pioneer Hundred Talents Program, in part by the MOE Key Laboratory of Machine Perception at Peking University under Grant K-2017-02, in part by the Business for R&D funded The Korea Ministry of SMEs and Startups in 2017 under Grant C0540726010047432O, and in part by NRF Grant funded by the Korean Government under Grant 015R1D1A1A01058171.

ABSTRACT Trajectory mining is an interesting data mining problem. Traditionally, it is either assumed that the time-ordered location data recorded as trajectories are either deterministic or that the uncertainty, e.g., due to equipment or technological limitations, is removed by incorporating some pre-processing routines. Thus, the trajectories are processed as deterministic paths of mobile object location data. However, it is important to understand that the transformation from uncertain to deterministic trajectory data may result in the loss of information about the level of confidence in the recorded events. Probabilistic databases offer ways to model uncertainties using possible world semantics. In this paper, we consider uncertain sensor data and transform this to probabilistic trajectory data using pre-processing routines. Next, we model this data as tuple level uncertain data and propose dynamic programming-based algorithms to mine interesting trajectories. A comprehensive empirical study is performed to evaluate the effectiveness of the approach. The results show that the trajectories could be modeled and worked as probabilistic data and that the results could be computed efficiently using dynamic programming.

INDEX TERMS Trajectory mining, sensor data, IoT.

I. INTRODUCTION

Trajectory mining is an interesting data mining problem that has been studied in the context of smart cities and the Internet of Things (IoT) [3], [6]. Smart cities and the IoT are indeed the way to the future as trillions of IoT devices, ranging from coffee machines to mobile objects which may or may not be inter-connected, generate enormous amounts of data which need to be modelled and processed effectively to improve daily life [5]. For example, to optimize the commuting time to work, many sources of information including the intended route, calendar, city traffic, weather, etc. need to come together to determine a route which would be the most convenient and therefore, smart data collection, preparation and fast algorithms are needed which can work with the incoming data and propose solutions in real time.

One of the key issues in future smart cities is the incorporation of intelligence into the cities using mobile intelligence [19]. The primary source of mobile intelligence is the mobility data collected through the Internet of Things. The data is obtained from a variety of sources, e.g. moving individuals or devices, which are constantly providing location data, along with a time stamp, to some central repository. Once such data is processed, interesting information could be revealed [4], for instance, which areas in the city are witnessing an increase in activity [6], [9], the location of any traffic anomalies [7], which person or group of people are moving [8], what the popular stay points are [5], etc.

A trajectory is a time-ordered record of a moving object obtained at pre-defined discrete time intervals. However, the 'exact' location of a moving object during these intervals could be uncertain. A lot of research has focussed on trajectory uncertainties with an aim to enhance the utility of trajectories. Probabilistic databases offer ways to model uncertainties using possible world semantics [1]. The uncertainties in the trajectories could be at the event level, which is the uncertainty associated with the location of the object, or at the trajectory level, which is the uncertainty associated with the path recorded as compared to the path taken, or others [11]. An interesting solution in this regard is to record the individual mobile object readings and then create complex events using probabilistic event extraction [3]. Many such systems have been proposed which work with trajectory

uncertainty, however Khoussainova *et al.* [3], proposed creating the events along with confidence values, and this is one of the motivations for the current study.

The possible world semantics [2] have been used to model uncertainties. The possible worlds are essentially the all possible combinations of the worlds where an event may or may not be present. However, the idea of using possible world semantics is inherently difficult due to the explosion in the number of possible worlds. Techniques such as dynamic programming [12] have been proposed in the literature which, although giving results which are the same as the possible worlds approach, are significantly fast. Further, many approximation schemes have been proposed in the literature that give comparable approximate solutions [10]. It should also be noted that in the literature, many simplifying assumptions are made to compute the solution using possible worlds. Such assumptions do not depict real-world situations and thus provide results which may not be very useful. Likewise, relatively complex models tend to be computationally intractable and have been shown to be #-P Complete or NP-Complete [11].

There is a need to develop solutions which model and extract the events in the uncertain trajectory data along with the confidence values and then compute the results based on the confidence values and this is the focus of this work.

In this work, we develop a framework that works by collecting the trajectory data obtained from the sensors. The data is stored and processed in a way that helps in identifying events such as key activity areas, evolving activity, etc. thus helping to attain better insight into the work habits of the population.

We work on the data obtained from the sensors attached to the office cards. The first step is to pre-process the data and model this as uncertain trajectory data. This is rather challenging considering the types of uncertainties involved, e.g. at the attribute level, i.e. source which generates information, at the tuple level, i.e. the location of the object, or at the trajectory level. Next, the processing that could be performed on the data, which in turn affects the quality of results that could be obtained from the data. Another important aspect is the choice of interestingness measures that is used to compute the results. Many such measures have been proposed in the literature; however, whilst the measures which are simple can be computed in a reasonable time, more detailed measures have been shown to be extremely computing extensive or even computationally intractable [11]. Thus, having a reasonably complex model that can capture the underlying uncertainties in the data as well as a measure which captures the essence of the information in the data is a challenging task which still needs to be investigated at large.

Once the probabilistic models have been developed, a processing framework is required which can be used to answer questions such as, which are the most frequent paths taken by the people in a locality? Or, which are the active regions in a locality over time? etc. The above questions and similar need to be answered in real time and, therefore, efficient mechanisms need to be developed which can work with uncertain trajectory data to yield the live state of the current happenings in a smart environment. This is extremely challenging as, when the live trajectories are being processed, they should be dealt with as a data stream and while many useful traditional data mining algorithms work under the assumption that the data are stored and could be re-accessed if desired, it becomes extremely challenging to produce similar kinds of results on-the-fly.

Our Contributions. We now give our contributions.

1) We propose working with the trajectory data which contains both the recorded event an also the event confidence and thus model uncertain sensor data as probabilistic events. These probabilistic events are used directly by the mining algorithm to find the interesting trajectories.

2) We present a dynamic programming based algorithm that computes interesting trajectories efficiently.

3) We give the proof of concept by presenting a case study of generating and processing sensor data in a building environment and give promising directions for future work.

The rest of this paper is organized as follows. In Section II, we discuss trajectory data extraction and mining. We present the probabilistic trajectory mining framework in Section III. An extensive experimental study is presented in Section IV and some related works are discussed in Section V. Section VI concludes this work.

II. TRAJECTORY DATA EXTRACTION AND MINING

In this section, we first discuss trajectory generation from the sensor data and then the trajectory mining. A sample trajectory data collection environment is presented in Fig. 1.

A. TRAJECTORY DATA

A trajectory is a collection of location data points ordered by a time stamp. A data point is a triple of the form *(eid, sid, e)* where *eid* is the time stamp, *sid* is the source identifier, and *e* is the event. The length of a trajectory t is the number of data points it contains, $\sum |t|$. For example, $t = \langle a_1, a_2, a_3 \rangle$, is a trajectory of three data points ordered in time. A trajectory $s = \langle a_1\ a_2 \cdots a_n \rangle$ is called a sub-trajectory of a trajectory, $\mathrm{t} = \langle b_1\ b_2 \cdots b_n \rangle$, if $s_i = t_j$, for $i, j \in [1, n], i \leq j$. In other words, we say that the trajectory t *contains* s. Given the location readings obtained from the sensors, a trajectory is obtained by combining in order all the location points recorded for a single object. The trajectory database contains the trajectories for all the sources. The support of a trajectory t is the number of source trajectories that contain the trajectory t.

The trajectory mining problem is defined as follows. Given a trajectory database D, find all interesting trajectories, i.e. trajectories whose support is at least a user-specified support threshold θ.

Trajectory mining is a multi-stage process which primarily involves pre-processing and pattern mining. We first discuss trajectory data pre-processing.

1) DATA CLEANING

Trajectory data is obtained from a variety of sources, e.g. sensors or other mobile devices, and is not entirely correct mainly due to equipment and technological limitations. The errors in trajectory data could be, e.g. (a) a location reading falling out of the motion track (or path) (b) or a moving object recorded at more than one distinct locations, simultaneously. In all such situations, data has to be cleansed. For example, the value of the location attribute is fixed using techniques such as mean/median filters, etc.

2) DATA COMPRESSION

Trajectory data is recorded at pre-defined discrete time intervals, e.g. a reading every few seconds by each sensor, and most of the points reported are usually repetitive and carry no significant information. However, keeping all the recorded data results in a notable increase in the computational complexity of the problem. It is a common practice to pre-process the data and reduce the number of exact locations recorded, i.e. attain some speed-up at the cost of losing some information which may not be worth the computation effort needed to process the information. Data compression could be offline or on the go and is performed using techniques such as computing the distance metric or similar.

3) TRAJECTORY CREATION

The final step is the trajectory creation which typically involves creating a trajectory of time-ordered location data points for each source. Once, the trajectories are created, data mining or other tasks could be performed. Another, important aspect is trajectory data management, however in this work we only focus on trajectory mining.

B. TRAJECTORY MINING

Once the trajectories have been created, trajectory patterns are discovered which could be one of the following.

1) TOGETHERNESS PATTERNS

These patterns are aimed at answering questions such as which objects move together. This could help in identifying an emerging activity in a locality or similar. The local administration can use such information in better managing the city's resources, e.g. traffic signals.

2) COMMON PATH PATTERNS

These patterns are the most frequent paths taken by the moving objects. The techniques used for finding such patterns include sequence mining, association mining, etc. These patterns generally help in predicting the next probable location of a moving object.

3) GROUP PATTERNS

Similar trajectories are grouped together to find groups of people who move similarly at the same points in time. This is not a trivial task as a feature vector has to be generated which is used to compute the distance between two trajectories. These group patterns show the group mobility trends and could be very useful when dealing with law and order situations, for instance.

4) CYCLIC PATTERNS

Moving objects usually have a similar mobility behaviour over time, i.e. the same activity is performed in cycles, going to work for example. If such personal information is known, it could be used to improve the commuting experience of the individuals, e.g. by warning them of slow moving traffic and then suggesting an alternate route, etc.

C. UNCERTAINTY IN TRAJECTORIES

Uncertainty in trajectories is a major concern in many situations as the trajectory data recorded is only a sample of the actual movement. Further, the exact location of a moving object at a specific point in time may not be known. A lot of research has focused on working with trajectory uncertainties. An interesting aspect is to record the object locations along with the confidence values, i.e. along with the location, also record the confidence in an object being at that location. This is a novel idea as in literature the uncertainties associated to the object's location are removed using some threshold based methods and the final trajectory has only deterministic time-ordered data points.

This work is focused on working with the uncertainty when dealing with trajectories. In the next section, we discuss the generation of probabilistic events from trajectory data and then present the probabilistic trajectory mining techniques to extract 'interesting' trajectories from the uncertain trajectory data.

III. TRAJECTORY MINING USING UNCERTAIN EVENTS

We first discuss uncertain event extraction from the recorded sensor data. The data recorded by the sensors is simple location data and the accuracy of such data is usually low. Thus, the events extracted from such data are uncertain, as issue which is discussed below.

A. UNCERTAIN EVENTS

The most primitive type of events are *presence* events. A *presence* event is of the form *(eid, sid, e, prob)*. For example, a sample reading looks like (t_1, s_2, l_3, *0.7)* which means that at time t_1, a source s_2 was spotted at location l_3 with probability *0.7*. The reason for having the probability is as follows. For example, a source s_1 enters a room which has three (03) antennae installed to detect an object in the room. If two of the three antennae report the presence of the source in the room, there is only a *66.6%* chance that the source was in the room at that time. It is also interesting that each antenna records the detection of an object with certainty, i.e. the reading from a single sensor looks like (t_1, s_2, l_3, *1.0)* which means that the source s_2 was sighted at location l_3 at time t_1 with probability *1.0*. This makes sense as an antenna only reports an event which is detected and there is no uncertainty

in this simple event. However, it is the readings from the neighbouring antennae which contribute to the belief in the presence of a source at a specific location. The system takes the readings from multiple antennae and only then decides the confidence in a *presence* event.

In a subsequent formulation, suppose a source s_1 also enters the room and one out of the three antennae detect s_1. What is the probability of the event that both s_1 and s_2 are in the room, together? An event of this form could be that the sources s_1 and s_2 are at location l_3 with probability *0.18*.

However, the detection of the sources s_1 and s_2 at location l_3 may not be sufficient to establish that both s_1 and s_2 are at location l_3. There is other information which should also be considered whilst creating such events, e.g. the ownership of the location l_3. If one of the two sources, is the owner of this location, it is probable that the two are together, for example for a meeting. However, if neither of the two owns the location, the detection of these two at the same place with low confidence, i.e. probability *0.18,* could be an error. Therefore, the detection and other information are also needed to establish such complex events and for establishing the truth value of a true occurrence.

Further, the sensors are not entirely accurate due to technological limitations, and for example, if a sensor has an error rate of 20% and there are a total of three (03) sensors at a point, then a presence event has a low accuracy.

B. UNCERTAIN DATA MODEL

From the previous section, we know that the uncertain events generated by the sensors are of the form *(eid, sid, e, prob)* which corresponds to tuple-level uncertainty, i.e. a tuple has an existential probability of occurrence. A sample probabilistic trajectory database is shown in Table 1. We now define the possible world semantics for an uncertain trajectory database D'.

TABLE 1. A sample probabilistic trajectory database.

time-stamp	*sid*	*eid*	*prob*
1	t_1	u	0.4
2	t_2	w	0.6
3	t_2	v	0.7
4	t_1	v	0.8

1) POSSIBLE WORLDS SEMANTICS

The possible world semantics are as follows. Given an uncertain trajectory database D', for each event e in a trajectory there are two kinds of worlds: one in which the event is present and the other where it is not. For each source trajectory t_i, the set of possible worlds is obtained by taking all possible combinations in which an event is present in the world or otherwise. The complete set of possible worlds is obtained by taking all such combinations. The probability of an event occurring is the cumulative probability of occurrence

TABLE 2. The trajectory database of table 1 transformed to probabilistic trajectories.

trajectory-id	*trajectory*
t_1	$(u: 0.4)(v: 0.8)$
t_2	$(w: 0.6)(v: 0.7)$

TABLE 3. The set of possible worlds for source t_1 from table 2.

World	*Probability*
$t_{1,1}$	$<> = (1 - 0.4) \times (1 - 0.8) = 0.12$
$t_{1,2}$	$\{u\} = 0.4 \times (1 - 0.8) = 0.08$
$t_{1,3}$	$\{v\} = (1 - 0.4) \times 0.8 = 0.48$
$t_{1,4}$	$\{u, v\} = 0.4 \times 0.8 = 0.32$

of the worlds where this event is present. As in the literature, we assume that the events across possible worlds occur independently of each other. An example of possible world computation is shown in Tables 2-4 for the sample database of Table 1 transformed to a trajectory database in Table 2.

TABLE 4. Complete set of possible worlds for the trajectory database of table 2.

World	t_1	t_2	$Pr(D'_*)$
D'_1	$<> = 0.12$	$<> = 0.12$	$= 0.12 \times 0.12$
D'_2	$<> = 0.08$	$\{v\} = 0.18$	$= 0.08 \times 0.18$
...	...	...	...
D'_{16}	$\{u, v\} = 0.32$	$\{w, v\} = 0.42$	$= 0.32 \times 0.42$

2) INTERESTINGNESS MEASURE

Using the possible world semantics, an event that occurs in a significant number of worlds with high probability is considered an interesting event. The interestingness measure, the expected support of an event, is defined in terms of the expectation of the event occurring in all the possible worlds, i.e. for a trajectory t,

$$ExpSup\left(t, D'\right) = \sum_{D^* \in PW(D')} \Pr\left[D^*\right] * Sup(t, D^*). \quad (1)$$

For example, the expected support of a trajectory $\{u,v\}$ in the sample database of Table 1 is computed by taking the sum of the probabilities of all the possible worlds which contain $\{u,v\}$, i.e. worlds $D'_{12} - D'_{16}$, as shown in Table 4.

3) UNCERTAIN PATTERN MINING

The uncertain pattern mining problem is defined as follows. Given a trajectory database, find all frequent patterns whose expected support is at least a user-defined support threshold θ.

Note that the number of possible worlds is exponential in nature and computing the expected support using possible worlds becomes computationally intractable. We now present a dynamic programming approach to compute the expected support of a trajectory.

C. EXPECTED SUPPORT COMPUTATION

Given a trajectory and a source trajectory, we create a dynamic programming matrix M, $(q+1)\times(p+1)$, where q is the number of elements in the trajectory and p is the number of elements in the source trajectory, and initialize all elements in the top row equal to *1* and all elements in the first column (except the top entry) equal to *0*. Next, we compute the other values row-by-row by using the following relation:

$$\mathrm{M}\left[\mathrm{i},\mathrm{j}\right] = \left(1\mathrm{c}_{\mathrm{ij}}\right) \times \mathrm{M}\left[\mathrm{i},\mathrm{j}-1\right] + \mathrm{c}_{\mathrm{ij}} \times \mathrm{M}\left[\mathrm{i}-1,\mathrm{j}-1\right] \quad (2)$$

An example of this computation is shown in Table 5. The right bottom cell in the table gives the expected support of the trajectory *{u,v}*.

TABLE 5. Computing expected support using dynamic programming.

		{u:0.4}	*{v:0.8}*
		1	1
{u}	0	0.4 × 1 + (1 – 0.4) × 1 = 0.4	0.4
{u, v}	0	0	0.4 × 0.8 + (1 – 0.8) × 0 = 0.32

The expected support of a trajectory t in the trajectory database D' is computed by summing the expected support of t across all trajectories.

Algorithm 1 An Outline of the Trajectory Mining Algorithm

1: **Given:** A Trajectory Database D', An Expected Support threshold Θ
2: **Required:** All frequent trajectories

3: $i = 2$
4: F_1 =: Compute all simple events
5: **while** F_{i-1} is not *null*
6: C_i =: join F_{i-1} with itself
7: Prune C_i
8: **for** all trajectories in C_i
9: Compute Expected Support
10: **end for**
11: F_i =: all frequent C_i
12: i =: i+1
13: **end while**
14: Output the frequent trajectories in F

D. UNCERTAIN TRAJECTORY MINING

The uncertain trajectory mining algorithm is similar to the uncertain apriori algorithm [12] and we give a few details here. An outline of the algorithm is given in Algorithm 1.

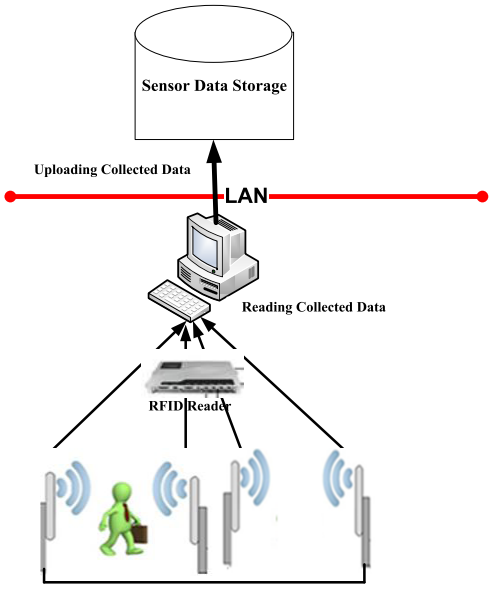

FIGURE 1. A typical sensor-based mobility data collection environment.

1) FREQUENT SIMPLE EVENTS

A scan of the trajectory database D' extracts all simple events in the database D' which have support at least equal to the threshold. The expected support of all the events is computed. Once the database has been scanned, all the frequent simple events have been found and these form the candidates for the next phase, i.e. frequent pair computation.

2) FREQUENT PAIRS

Once the frequent simple events have been computed, all possible pairs of events are generated which are then tested for being frequent. Note that only the frequent simple events are used to generate candidate frequent pairs. This is due to the apriori property which is anti-monotonic and states that for any pair event to be frequent, both the simple events in the pair have to be frequent. For example, for *{u, v}* to be frequent, both *{u}* and *{v}* need to be frequent. Only then should the support computation test be performed.

3) FREQUENT TRAJECTORIES

The frequent pairs discovered during the previous phase are used to generate candidate trajectories by appending the frequent simple events to the frequent pairs. The idea is that a candidate trajectory can be extended by appending a simple event to a frequent trajectory which has already been discovered. This step continues until no more candidate trajectories can be created or all the frequent trajectories have been discovered.

An outline of the process discussed in Section III is shown in Fig. 2. As shown in the figure, it is a two-stage process. In the first stage, the trajectories are extracted which are then mined to extract frequent trajectories in the second stage.

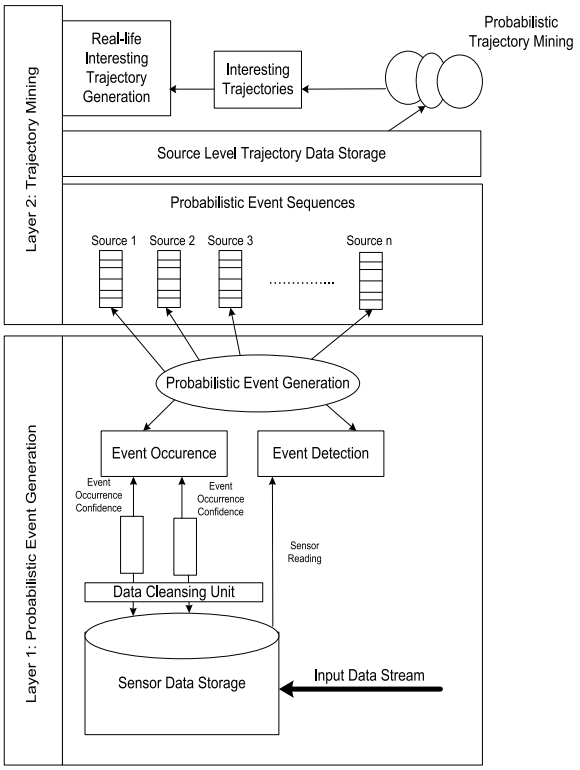

FIGURE 2. An overview of trajectory mining using uncertain sensor data. Layer 1 deals with the probabilistic event generation whilst trajectory mining is performed at layer 2.

IV. EMPIRICAL EVALUATION

We now discuss empirical evaluation and a brief account of the results. We first give the dataset, then the data preparation, after which we discuss the mining results.

A. DATASET

We used twenty participants in this study who were monitored using RFID sensors attached to their office cards. The antennae were placed on two floors, i.e. in the corridors, towards the café, and inside the offices. The location data were recorded every three seconds and the participants were told to take their usual tea break at 10:30 and their lunch break at 12:30. The participants were monitored for four consecutive hours, i.e. from 9:00 to 1:00 in the afternoon. A total of 96,000 readings were reported to the system.

B. DATA PREPARATION

The data processing and cleaning step detected a total of 594 *presence* events, i.e. there were only 594 events where there was a change in location. We used 20 participants for the study however, in the experiments, we have duplicated the participants to increase the number to 50, i.e. 10 randomly selected participants out of 20 have been replicated to form 30 and replicating every participant forms 40, and so on. The idea was to test the scalability of the proposed approach by duplicating the available data. The precision and recall values were tested using the provided data and we aimed

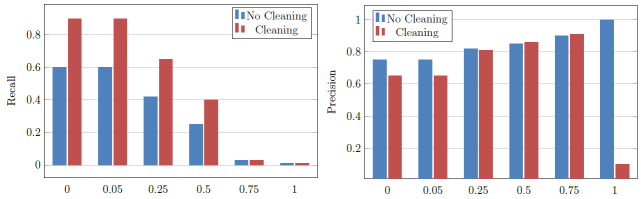

FIGURE 3. Recall and precision for event extraction with and without cleaning (results reproduced from [3]).

for maximum recall at the cost of some precision. Further, in the case of recall, data cleaning significantly improved the recall and we set the probability threshold to 0.05 which gave around 90% recall and with more than 60% precision. The extraction process was performed as in [3] and the results reported are similar as shown in Fig. 3.

C. TRAJECTORY MINING

In the next step, the trajectories were input to the mining algorithm which reported the frequent trajectories. The running time recorded in seconds and the number of interesting trajectories reported are given in Fig. 4. The support threshold was varied from 10% to 40%, the trajectory length changed every hour and increased to 8, 14, 17, 27 during hours 1 – 4, and the interesting trajectories were reported every hour.

We now discuss the results.

1) SUPPORT THRESHOLD

The support threshold was changed from 10% to 40% and the number of sources were changed from 10 to 50. As can be seen in Fig. 4, at a higher support threshold, e.g. 30% or 40%, only a few candidate trajectories pass the threshold and therefore the algorithm runs quickly. However, when the support threshold is low, for example at 10%, many trajectories pass the support threshold and therefore it takes significantly longer to complete the run. The running times regards to number of sources scale well and doubling the number of sources takes slightly more than double the time, a fact that is understandable due to increased I/O costs.

2) TRAJECTORY LENGTH

As a general observation, a linear increase in the trajectory length has a significant impact on the running time and the number of frequent trajectories reported. For example, during hour 1, considering 40 sources and a support value of 10%, it takes 800+ seconds to complete the run. However, after hour 4, and for the same number of sources and the same support threshold, the running time has increased beyond 15,000 seconds. This is a concern as it may become even more difficult to compute the results beyond four hours or alternatively, interesting trajectories for the full day may not be possible to be computed using the current settings. This is understandable as probabilistic trajectory mining uses the dynamic programming routine for expected support calculation and with an increase in the length of the trajectories,

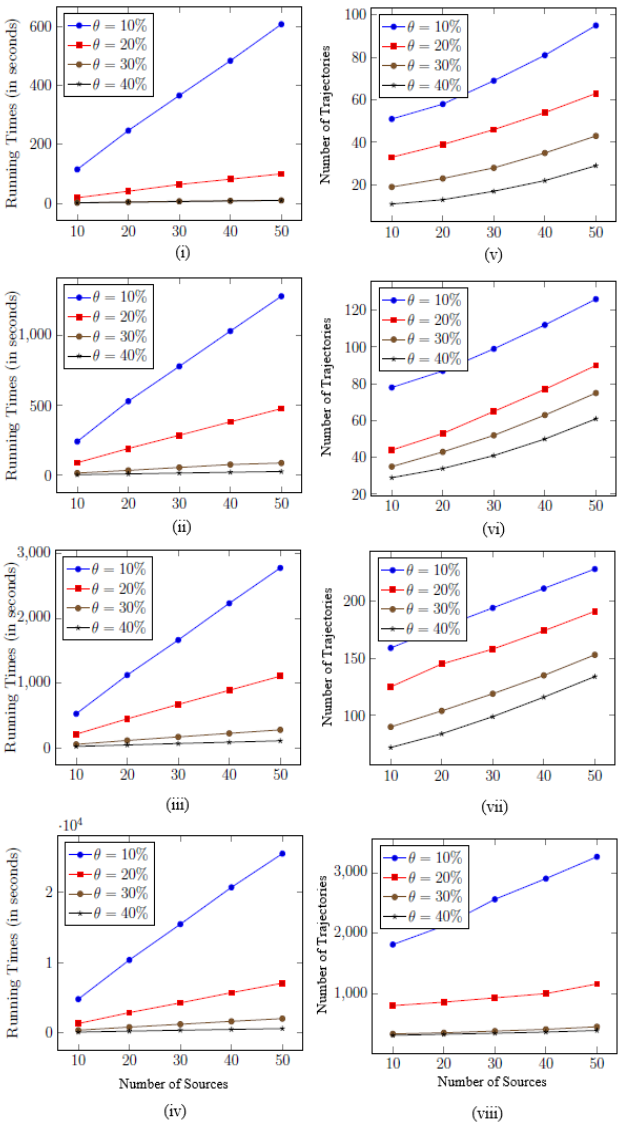

FIGURE 4. Trajectory mining results, i.e. running times and number of frequent trajectories reported for hours 1 – 4. The graphs on the left show the running times recorded in seconds whereas the graphs on the right show the number of frequent trajectories reported. The number of sources is increased from 10 to 50 and the average number of events recorded per trajectory is 8, 14, 17 and 27 for hours 1 – 4, respectively.

the dynamic programming tables become large and take significantly longer to compute.

Remark: The dynamic programming computation is expensive, and we are required to make approximations to compute the frequent trajectories when working with day-long activities or beyond. The trajectory data could also be considered to further decrease the trajectory length. One idea may be to work with slightly compressed trajectories and then to generate the full paths from the interesting trajectories using map matching or similar techniques. Furthermore, as each trajectory can be processed independently, distributed computing models, e.g. Hadoop, could be used in the future to exploit parallelism by working with the trajectories independently.

V. RELATED WORK

The identification of positioning information regarding people, cars, and other devices is very useful for supporting many of life's daily activities. There are various technologies available to identify location such as Global Positioning Systems (GPS), Radio Frequency Identification (RFID), location estimation of 802.11, GSM beacons, smart phone sensors, infrared or ultrasonic systems [13]. The development of these technologies has made it very easy to produce large-scale trajectory data which trace moving objects. Moving objects produce continuous traces in a geographical space from which samples of location points visited by the moving object are taken. A spatial trajectory is an example of trajectory data which include spatial information along with location information. The sampling rate and duration of a trajectory can be chosen depending on the application.

In the past decade, many techniques have been developed for trajectory data mining. However, there are several challenges posed by the processing of trajectory data. Typically, the volume of rapidly accumulated trajectory data is large, which makes storage of the data a non-trivial task. Moreover, the definition of a similarity metric for comparing trajectories, which is a basic requirement for trajectory mining, is very important as trajectories may be generated with different sampling strategies considering varying sampling rates. In addition to this, processing queries on large volumes of trajectory data is complicated because of the time and space complexities [14], [15]. Typically, pre-processing is performed on the data which involves cleaning, segmentation, completion, calibration, and sampling.

Multiple sensors may detect an object but a deterministic location may not exist. Therefore, cleaning is undertaken which discards the impossible trajectories by considering several constraints such as speed and unreachability constraints [16]. A trajectory may be divided into segments, i.e. sub-trajectories, which correspond to the underlying structures in the data like a path with multiple road segments [17]. The segmentation allows us to efficiently store sample points of moving objects aligned by time intervals. Low sampling rates may be used for trajectory collection which allow only partial observations of the actual routes because of storage and transmission considerations. Such trajectories are known as uncertain trajectories. Some approaches have already been developed to complete uncertain trajectories and support data mining tasks [20]–[24]. Heterogeneous trajectories represent discrete approximations of the original routes and have various strategies and rates of sampling. The heterogeneity may negatively affect the similarity measurement of a trajectory. Therefore, the use of techniques for transformation of heterogeneous trajectories to ones with unified sampling strategies are required [25], [26]. Various sampling techniques are available

for reducing a large trajectory database by selecting only those trajectories which accurately represent the original trajectory. A key point is that the sample should capture the hidden mobility pattern from the original trajectory [17], [18], [27], [28].

VI. CONCLUSION

In this study, we proposed a framework for probabilistic trajectory extraction and mining from uncertain trajectory data. This is the first study on the subject and many interesting directions need to be explored, e.g. going beyond the number of sources and hours considered in this study. We would also be interested in identifying and developing alternative approaches with the use of which we can make the approach more scalable, e.g. a trajectory compression scheme could be developed to further decrease the length of the trajectories. Further, an approximation scheme could be developed to avoid the dynamic programming processing at the cost of some accuracy. The inherent independence of the trajectories could be used to adapt the proposed algorithm to a distributed computing environment, e.g. Map-Reduce. This work has focused on mining uncertain trajectories. An insight into the discovered trajectories and assessment of the usefulness of the trajectories could also be subjects for interesting future work. We have used expected support as the interestingness measure. Other interestingness measures proposed in the literature, e.g. probabilistic frequentness, could also be investigated and a comparison made with the expected support in terms of time and the discovered trajectories.

REFERENCES

[1] D. Suciu, D. Olteanu, C. Ré, and C. Koch, "Probabilistic databases," *Synthesis Lectures Data Manage.*, vol. 3, no. 2, pp. 1–180, 2011.
[2] H. J. Levesque, "A logic of implicit and explicit belief," in *Proc. AAAI*, Aug. 1984, pp. 198–202.
[3] N. Khoussainova, M. Balazinska, and D. Suciu, "Probabilistic event extraction from RFID data," in *Proc. IEEE 24th Int. Conf. Data Eng. (ICDE)*, Apr. 2008, pp. 1480–1482.
[4] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 29, 2015.
[5] P. Wang, Y. Fu, G. Liu, W. Hu, and C. Aggarwal, "Human mobility synchronization and trip purpose detection with mixture of Hawkes processes," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 495–503.
[6] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng, "Planning bike lanes based on sharing-bikes' trajectories," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1377–1386.
[7] Y. Fu *et al.*, "Sparse real estate ranking with online user reviews and offline moving behaviors," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2014, pp. 120–129.
[8] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 186–194.
[9] F. Giannotti *et al.*, "Unveiling the complexity of human mobility by querying and mining massive trajectory data," *VLDB J. Int. J. Very Large Data Bases*, vol. 20, no. 5, pp. 695–719, 2011.
[10] T. Calders, C. Garboni, and B. Goethals, "Approximation of frequentness probability of itemsets in uncertain data," in *Proc. IEEE 10th Int. Conf. Data Mining (ICDM)*, Dec. 2010, pp. 749–754.
[11] M. Muzammal and R. Raman, "On probabilistic models for uncertain sequential pattern mining," in *Advanced Data Mining and Applications*. Berlin, Germany: Springer, 2010, pp. 60–72.
[12] M. Muzammal and R. Raman, "Mining sequential patterns from probabilistic databases," *Knowl. Inf. Syst.*, vol. 44, no. 2, pp. 325–358, 2015.
[13] H. Jeung, H. Lu, S. Sathe, and M. L. Yiu, "Managing evolving uncertainty in trajectory databases," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1692–1705, Jul. 2014.
[14] R. G. Baraniuk, "More is less: Signal processing and the data deluge," *Science*, vol. 331, no. 6018, pp. 717–719, Feb. 2011.
[15] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, "VAIT: A visual analytics system for metropolitan transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1586–1596, Dec. 2013.
[16] B. Fazzinga, S. Flesca, F. Furfaro, and F. Parisi, "Cleaning trajectory data of RFID-monitored objects through conditioning under integrity constraints," in *Proc. 17th Int. Conf. Extending Database Technol. (EDBT)*, Athens, Greece, Mar. 2014, pp. 379–390.
[17] N. Pelekis, I. Kopanakis, C. Panagiotakis, and Y. Theodoridis, "Unsupervised trajectory sampling," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML PKDD)*, Barcelona, Spain, Sep. 2010, pp. 17–33.
[18] Y. Li *et al.*, "Sampling big trajectory data," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Melbourne, VIC, Australia, Oct. 2015, pp. 941–950.
[19] M. Gohar, M. Muzammal, and A. Rehman, "SMART TSS: Defining transportation system behaviors using big data analytics in smart cities," *Sustain. Cities Soc.*, to be published.
[20] P. Banerjee, S. Ranu, and S. Raghavan, "Inferring uncertain trajectories from partial observations," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Shenzhen, China, Dec. 2014, pp. 30–39.
[21] M. Li, A. Ahmed, and A. J. Smola, "Inferring movement trajectories from GPS snippets," in *Proc. 8th ACM Int. Conf. Web Search Data Mining (WSDM)*, Shanghai, China, Feb. 2015, pp. 325–334.
[22] M.-F. Chiang, Y.-H. Lin, W.-C. Peng, and P. S. Yu, "Inferring distant-time location in low-sampling-rate trajectories," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Chicago, IL, USA, Aug. 2013, pp. 1454–1457.
[23] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Washington, DC, USA, Apr. 2012, pp. 1144–1155.
[24] N. D. Larusso and A. Singh, "Efficient tracking and querying for coordinated uncertain mobile objects," in *Proc. 29th IEEE Int. Conf. Data Eng. (ICDE)*, Brisbane, QLD, Australia, Apr. 2013, pp. 182–193.
[25] H. Su, K. Zheng, H. Wang, J. Huang, and X. Zhou, "Calibrating trajectory data for similarity-based analysis," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, New York, NY, USA, Jun. 2013, pp. 833–844.
[26] H. Su, K. Zheng, J. Huang, H. Wang, and X. Zhou, "Calibrating trajectory data for spatio-temporal similarity analysis," *VLDB J.*, vol. 24, no. 1, pp. 93–116, Feb. 2015.
[27] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, and Y. Theodoridis, "Segmentation and sampling of moving object trajectories based on representativeness," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1328–1343, Jul. 2012.
[28] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.

MUHAMMAD MUZAMMAL received the Ph.D. degree in computer science from the University of Leicester, U.K. He is currently with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China, and the Department of Computer Science, Bahria University, Islamabad, Pakistan. His research interests are in data mining, uncertain data, and human-centred computing.

MONEEB GOHAR received the B.S. degree in computer science from the University of Peshawar, Pakistan, in 2006, the M.S. degree in technology management from the Institute of Management Sciences in 2009, and the Ph.D. degree from the School of Computer Science and Engineering, Kyungpook National University, South Korea, in 2012. From 2012 to 2014, he was a Post-Doctoral Researcher with the Software Technology Research Center, Kyungpook National University, South Korea. From 2014 to 2016, he was a Foreign Assistant Professor with the Department of Information and Communication Engineering, Yeungnam University. He has been a Senior Assistant Professor with the Department of Computer Science, Bahria University, Islamabad, Pakistan, since 2016. His research interests include network layer protocols, wireless communication, mobile multicasting, wireless sensors networks, TRILL, and internet mobility.

ARIF UR RAHMAN received the Ph.D. degree from the University of Porto, Portugal. He is currently with the Department of Computer Science, Bahria University, Islamabad, Pakistan, and the Faculty of Computer Science, Free University of Bolzano, Italy. His areas of interest include information retrieval, digital preservation, and data analysis.

QIANG QU received the M.Sc. degree in computer science from Peking University and the Ph.D. degree from Aarhus University. He is currently an Associate Professor and the Executive Director of the Global Centre for Big Mobile Intelligence, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (CAS). He is supported by the CAS Pioneer Hundred Talents Program. His current research interests are in data-intensive applications and systems.

AWAIS AHMAD received the Ph.D. degree in computer science and engineering from Kyungpook National University, Daegu, South Korea. He is currently an Assistant Professor with the Department of Information and Communication Engineering, Yeungnam University. In 2014, he was also a Visiting Researcher with INTEL-NTU, National Taiwan University, Taiwan, where he involved in the Wukong Project (Smart Home). He was serving as a Lab Admin of CCMP Labs from 2013 to 2017. Since 2013, he has authored over 70 international journals/conferences/book chapters in various reputed IEEE, Elsevier, and Springer journals, whereas in leading conferences, such as the IEEE Globecom 2015, the IEEE Globecom 2016, the IEEE LCN 2016, and the IEEE ICC 2017, respectively. His current research interest includes big data, Internet of Things, social Internet of Things, and human behavior analysis using big data. He has been a Key Contributor in the said fields. He was a recipient of three prestigious awards, such as the IEEE Best Research Paper Award: International Workshop on Ubiquitous Sensor Systems (UWSS 2015), in conjunction with the Smart World Congress (SWC 2015), Beijing, China, 2015, the Research Award from President of Bahria University Islamabad, Pakistan in 2011, the Best Paper Nomination Award in WCECS 2011 at UCLA, USA, and the Best Paper Award in First Symposium on CS&E, Moju Resort, South Korea, in 2013. He was honored as a Best Outgoing Researcher of CCMP Labs. He is also serving as a Guest editor in various Elsevier and Springer journals, including FGCS, Sustainable Cities and Society, and RTIP. He is an Invited Reviewer of various journals, including the IEEE COMMUNICATION LETTERS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEM, the IEEE JSTAR, the *Ad-Hoc Networks* (Elsevier), the *Computer Network* (Elsevier), and the IEEE *Communications Magazine*.

GWANGGIL JEON received the B.S., M.S., and Ph.D. *(summa cum laude)* degrees from the Department of Electronics and Computer Engineering, Hanyang University, Seoul, South Korea, in 2003, 2005, and 2008, respectively. From 2008 to 2009, he was with the Department of Electronics and Computer Engineering, Hanyang University, from 2009 to 2011, he was a Post-Doctoral Fellow with the School of Information Technology and Engineering, University of Ottawa, and from 2011 to 2012, he was an Assistant Professor with the Graduate School of Science and Technology, Niigata University. He is currently an Associate Professor with the Department of Embedded Systems Engineering, Incheon National University, Incheon, South Korea. His research interests fall under the umbrella of image processing, particularly image compression, motion estimation, demosaicking, and image enhancement and also computational intelligence such as fuzzy and rough sets theories.