EEE International Conference on Big Data and Smart Computing (IEEE BigComp 2017), Jeju, Korea, February 13-16, 2017, (c) IEEE 2017

Predicting Indoor Spatial Movement Using Data Mining and Movement Patterns

Luan D. M. Lam and Antony Tang
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Hawthorn, Victoria 3122
Email: {ldlam, atang}@swin.edu.au

John Grundy
Deakin University, Geelong, Australia
School of Information Technology
Faculty of Science Engineering and Built Environment
Email: j.grundy@deakin.edu.au

***Abstract*—The ability to accurately predict the movement trajectory of people holds potential benefits for many applications, such as aged care and retail. Such movement predictions rely on collecting and analyzing large amounts of positioning data from sensors. In this work, we describe new algorithms to mine and predict people's movement in an indoor environment. Movement patterns are mined from historical positioning data, and the patterns are used to construct a probability tree which is a visual representation of frequent movements. We have conducted an empirical study in a staff tearoom to capture positioning data, mine movement patterns and construct a probability tree. We show the predictive power of the algorithms using different trajectory estimation strategies.**

***Index Terms*—Location Prediction, Spatial Behaviors, Sequential Pattern Mining, Indoor Environments**

1. Introduction

Location prediction has attracted much research interest due to many potential applications. Knowledge of where people would typically go could be used to improve their convenience and safety, for example, by optimizing nurses' trajectories in a ward, or detecting elderly people's abnormal behaviors in terms of unusual spatial movements. It could help to inform targeted retail promotions, optimal location of products, and customer and staff safety.

In this work, we investigate movement prediction using spatial movement patterns. Such patterns can be gained by analyzing peoples' historical movements. In the case of elderly people living alone at home or in a nursing home, this data gathering needs to be continuous and becomes very large. Analyzing movement patterns related to a group of people could reveal where they typically go and their sequence of visits. On the other hand, analyzing movement patterns of each individual could gain insights into his or her living habits and associated challenges.

There exist many location prediction techniques utilizing movement patterns for predicting next locations. For example, works such as [9], [14], [15] rely on certain pattern mining techniques to extract movement patterns from historical movement data, based on which the next location is predicted. However, these techniques have limitations. They do not utilize patterns that re-occur frequently in long movement sequences, which are common in indoor environments. Besides, current techniques in the literature do not predict consecutive places of visits. Instead, the prediction is to provide probabilities that a place would eventually be visited. Furthermore, as people's movement behaviors may change over time, there needs to have a mechanism to continuously update movement patterns, which has not been studied in current location prediction techniques.

In this work, we mine people's movement patterns from historical movement data captured by sensors. From the mined patterns, we construct a probability tree to visually depict the common movement behaviors. The tree comprises of nodes, each represents a location, connected by edges, representing the transitions between nodes. Each node has a probability indicating the likelihood of itself being visited next if its parent has been visited. We propose a mechanism that allows continuous updating of movement patterns and the probability tree. We predict the next location by finding matches between a person's current trajectory and the probability tree. We investigate different estimation strategies to allocate priority to each match, based on which the next location is predicted, including: allocating priority equally and allocating priority according to the specificity of the match. We test our algorithms and compare the results using empirical data from movement in a real-world environment.

The structure of this paper is as follows. We present the motivation for our work in Section 2, and related works in Section 3. Background about movement patterns is presented in Section 4. Our approach is described in Section 5. Experimental details are provided in Section 6. We conclude the paper in Section 7.

2. Motivation

Consider a typical ward in a hospital. Nurses are stationed at the nursing station to use computers, they collect medicines and utensils from a treatment room, visit patients, treat patients and so on. The routines of nurses are dictated by the different tasks they undertake during a shift. However, hospital management does not know how best to physically design nursing station to model typical movements of nurses in order to optimize movements. In a similar way, the staff tea room in a typical organization has numerous users, various artifacts such as chairs, tables, sink, microwave oven, coffee machine, notice board, water cooler and so on. Often limited thought goes into positioning these based on potential or actual usage. The study of people's spatial movement and prediction in such environments can help us to better understand typical movement behaviors for different purposes, and use these for movement optimization, anomaly detection, improve safety and efficiency, and better support living and working patterns.

We conducted an empirical study in a typical large organization staff tearoom to capture and analyze staff movements. We wanted to understand the places that they visit whilst in the tearoom. Do they go to the refrigerator in the morning to store their food before they make tea? Or do they wash their cups before they get milk? We wanted to understand their space and artifact usage patterns with a view to being able to both predict these and use the predictions.

To this end, we collected movements and mined movement patterns over a period of time. Our example experimental area, shown in Fig. 1, contains a number of places that are of interest, such as microwave oven, where people visit to heat up a meal or a drink, or refrigerator, where people go to put things in or get things out. In our work, these places are called Points-of-Interest (POIs). A person at a POI has an implication of his or her current activity. There are seven POIs in the experimental area:

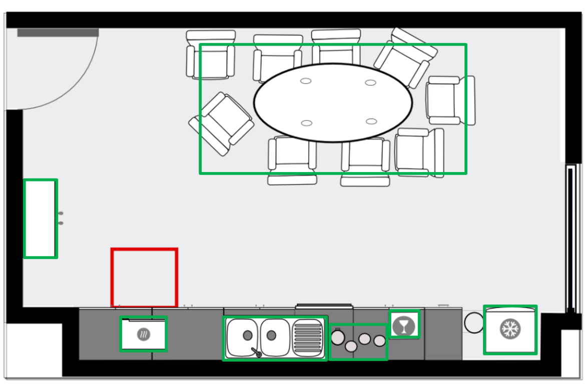

Figure 1. The experimental area. The area contains a number of Points-Of-Interest (POIs). The green rectangles indicate POIs. The red rectangle is an example of a POI area where a person stays upon when using the microwave oven; every POI has a corresponding POI area.

- **microwaveoven**: which corresponds to using the microwave oven.
- **sink**: which corresponds to using the sink, water taps or tissue dispenser.
- **teaarea**: which corresponds to accessing jars containing tea bags, sugar bags, sticks, etc.
- **coffeemachine**: which corresponds to making coffee with the machine.
- **fridge**: which corresponds to accessing the refrigerator.
- **table**: which corresponds to sitting at the table and chairs.
- **cabinet**: which corresponds to accessing the cabinet.

We collect movement data and mine movement patterns based on POIs. Details of this process are described in Section 6.

3. Related Work

Various data mining algorithms have been developed, extracting different classes patters from input sequences [4], [18]. A number of research works have been conducted to mine trajectory patterns for location prediction. A data mining approach for location prediction has been proposed for use in mobile environment [14]. In this work, from a long sequence of cells that a mobile user has visited, a set of user actual paths (UAPs) is obtained by breaking the long input sequence into multiple sub-sequences. From the set of UAPs, user mobility patterns (UMPs) are mined, where the support of a candidate pattern is calculated based on the matching level of the candidate pattern with the UAPs containing it. The matching level of a candidate pattern and an UAP is defined by means of the notion of string alignment [3]. From the set of the mined patterns, mobility rules are generated by decomposing a pattern into multiple sets of head and tail. The confidence of a rule is calculated by dividing the support of the original pattern by the support of the head of the rule. Rules whose confidence exceeds a minimum threshold are used for prediction. Given a current cell trajectory of a user, all matching rules, whose head is contained in the trajectory and ends at the last cell of the trajectory, are selected; the predictive power of a matching rule is calculated by summing the confidence of the rules and the support of the pattern from which the rule is formed.

Semantic trajectories have also been mined and used for location prediction [15]. In this work, both semantic trajectory and geographic trajectory are used to predict a mobile user's next location. Semantic trajectories are defined as sequences of locations labelled with semantic tags - such as *Bank*, *Park*, *Restaurant*, etc. - to capture the landmarks visited. Geographic trajectories are sequences of stay locations tagged with timestamps. Mobile users' historical movement data are converted into both semantic sequences and geographic sequences. Semantic trajectory patterns for each user are mined using the PrefixSpan algorithm [5]; mined semantic trajectories are then used to build up a *semantic trajectory pattern tree* (STP-Tree). Similarities between two mobile users' movements are calculated based on the notion of *Maximal Semantic Trajectory Pattern Similarity* (MSTP-Similarity) [16], upon which clusters of users with similar movement behaviors are formed. Geographic patterns are also mined for each cluster of users, which are then used to build up a *stay location pattern tree* (SLP-Tree) for each cluster. To predict location, geographic score and semantic score are calculated by measuring the matching score of the user's current trajectory with the corresponding stay location patterns in the user's cluster and the user's personal semantic trajectory patterns respectively.

An extension of the PrefixSpan [5] has been developed to mine frequent trajectories for location prediction [9]. In this work, the two-dimensional movement area is divided into a set of rectangular regions of fixed size, which is referred to as *cells*. An object's movements are then defined as a sequence of traversed edges of the pre-defined cells. The support of a trajectory is the fraction of input trajectories containing it. From mined patterns, movement rules, in the form of $tail \Rightarrow head$, are derived whose *confidence* is calculated by dividing the support of the original pattern by the support of the tail of the rule. There are three strategies used for predicting location based on derived movement rules: whole matcher which will apply rules whose tail matches the current trajectory, last matcher which will apply rules whose tail matches the last element of the current trajectory, and longest last matcher which will apply rules whose tail matches any sequence of consecutive elements of the current trajectory that includes the last element. In the longest last matcher, the strength of the prediction is calculated as the product of the confidence of applied rule and the fraction of elements in the current trajectory that match the tail of the rule; while in the other two strategies, the prediction strength is the confidence of applied rules.

Whilst there are many pattern-based location prediction techniques, these techniques do not take into account the re-occurrences of a pattern in long movement sequence and do not predict consecutive places visited. They do not have a mechanism to allow continuous updating of movement patterns. Various machine learning techniques have been developed to predict people's location given their historical movements, such as dynamic Bayesian network [10], [12], neural networks [13], Bayes rule [6], etc. A survey of different machine learning techniques for location prediction - including dynamic Bayesian network, multi-layer perceptron, Elman net, Markov predictor, and state predictor - has been conducted [11]. The analysis results indicate that each technique possesses particular strengths and weaknesses, and the choice for a technique would depend on the application requirements and characteristics. In this work, we chose to mine and use positioning patterns for location prediction as with the understanding of positioning patterns, prediction results are more intuitive and understandable to human.

4. Problem Definition

We define *movement sequence* to be a collection of places that are visited in a sequential manner. *Movement patterns* are movement sequences that have frequently occurred. There are different types of patterns. In this section, we provide definitions for eventual pattern and immediate pattern. Eventual patterns represent frequent movements where each POI would be eventually visited if a POI before it has been visited; immediate patterns represent frequent movements where the set of POIs are visited consecutively according to their order in the patterns. More details are provided in [7][1].

An eventual movement sequence is a collection of POIs in a fixed order. Given a set of *points of interest* $I = \{P_1, \ldots, P_m\}$, an eventual movement sequence *s* on *I* is an ordered sequence of POIs $s =< P_1, P_2, \ldots, P_n >$, where $P_i \in I$ for $i = 1, \ldots, n$. For example, a sequence that the *fridge*, *microwaveoven*, *table* and *sink* were visited sequentially is shown as:

$$s_{Example} =< fridge, microwaveoven, table, sink >$$

The number of POIs in a sequence is the length of that sequence. A sequence with length l is called a l*-sequence*. A movement sequence can be used to depict a single trip of a person moving in the monitoring area, in which their movement is represented sequentially in terms of the POIs that they visited.

An eventual movement sequence $\alpha =< P_1, \cdots, P_n >$ is an *eventual sub-sequence*[2] of an eventual movement sequence $\beta =< P'_1, \cdots, P'_m >$ if there exist integers $j_1, j_2, \cdots, j_n$ such that $1 \leq j_1 < j_2 < \cdots < j_n \leq m$ and $P_1 = P'_{j_1}$, $P_2 = P'_{j_2}$, $\cdots$, $P_n = P'_{j_n}$. In this case, β is also called the super-sequence of α, denoted as $\alpha \sqsubseteq \beta$. We also say that β contains α, or α is contained in β.

Minimal occurrences (MOs) of a sequence are occurrences that do not contain any other occurrence, and as such are *minimal*. The *support* of a movement sequence is defined as the number of its minimal occurrences. Given a sequence dataset S and a minimum support threshold $minSup$ [3], a movement sequence s is said to be frequent in S if its support in S is no less than $minSup$. In this case, s is also called a (movement) pattern in S.

Given a movement sequence α, a minimal occurrence of α (sid, t_s, t_e) is said to be an immediate minimal occurrence if it is not a minimal occurrence of any proper super-sequence of α. The set of all immediate minimal occurrences of a sequence α is denoted as $imo(\alpha)$. A movement sequence α is said to be an immediate pattern in a dataset S if its number of immediate MOs exceeds a minimum threshold $minSup$. In this case, the number of immediate MOs of the sequence α is also called the immediate support of α.

An immediate movement sequence $\alpha =< P_1, \cdots, P_n >$ is called an *immediate sub-sequence*[2] of an immediate movement sequence $\beta =< P'_1, \cdots, P'_m >$ if there exists an integer j such that $1 \leq (j + 1)$ and $(j + n) \leq m$, and $P_1 = P'_{j+1}$, $P_2 = P'_{j+2}$, $\cdots$, $P_n = P'_{j+n}$. In this case, β is also called the super-sequence of α, denoted as $\alpha \sqsubseteq \beta$.

The algorithm to mine eventual movement patterns has been provided in [8], whereas mining immediate movement patterns is provided in [7]. In this work, we aim to predict people's next locations by continuously mining and updating immediate movement patterns, and developing prediction

1. The notion of movement sequence defined in this paper corresponds to the notion of serial movement sequence in [7]

2. In this paper, we use the term *sub-sequence* (or *super-sequence*) to indicate both the eventual and immediate sub-sequence (or super-sequence) relationships, or either of them when the context is clear

3. In this paper, we refer $minSup$ as the mining threshold

techniques using the mined patterns. We specifically focus on the use of immediate movement patterns, as they represent consecutive places of visits and hence are more suitable for predicting the next location.

5. Our Approach

We mine immediate movement patterns and use them to predict people's next location based on their current trajectory and the knowledge of their movements. Our system architecture comprises a number processing layers (see Fig. 2): the *Sensoring Layer* collects *Sensory Data* that are then fed into *Positioning Layer* to compute *Positioning Data*. The *Positioning Data* are in the form of Cartesian coordinates. People's movements are represented as sequences of coordinates ordered by ascending timestamps. These sequences are then *contextualised* to produce *Movement Sequences*, that is, the (x, y) coordinates are now translated into corresponding POI such as $fridge$ or $sink$. The results of the *contextualisation* process are sequences of visited places that are of interest to monitoring. These *Movement Sequences* are stored in a database where movement patterns are mined. The movement patterns and historical movement data are also used to predict the next location that would be visited by a person.

Movement patterns are then used to construct a probability tree. The probability tree is formed by grouping immediate movement patterns based on their common prefixes, and is represented in terms of nodes and edges; each node represents a POI and an edge depicts the transition between two corresponding nodes. At each node, there is a corresponding probability value indicating the likelihood of visiting this node given its parent node has been visited. The probability tree serves two purposes: firstly, it is a graphical representation of movement patterns and provides a holistic view of the corresponding spatial behaviors; secondly, the tree is a structure that contains the probabilities of the movement patterns or common trajectories for predicting next location. A snippet of such a tree is shown in Fig. 3. We show the algorithm to build up a probability tree from movement patterns in Section 5.1, the mechanism to continuously update movement patterns and probability tree in Section 5.2, and the algorithms to predict next location in Section 5.3.

5.1. Constructing probability tree

The algorithm to construct a probability tree is shown in Algorithm 1. The inputs to the algorithm are the set of mined movement patterns and a set of input movement sequences.

In step 1, we construct an empty tree containing only one root node. We loop through each movement pattern, and traverse through each POI sequentially. For each POI, starting at the root, we find a child node that matches the POI. There are two possibilities: if there is a match, we assign the support of the current pattern to the support of the found node if the support of the node is less than that of the pattern. In case that there is no match, we create a new node, whose support is equal to the support of the pattern, and assign this node as a child node of the current node. We then process the next POI, using either the found node if it exists or the newly created node as the next processed node.

In step 2, we compute, for each POI, the relative percentage of it being the first visited POI in movement sequences. We use these percentages to heuristically calculate the probability of each POI under the root node for two reasons. Firstly, we have observed that there are various activities that can be performed in our experimental area and every POI can be the first visited POI. Secondly, the first POI in a movement pattern is not necessarily the first visited POI in a movement sequence.

In step 3, we compute the probability of every node in the tree excluding the root. If the parent node is the root, it's probability is calculated based on the percentage values calculated in step 2. Otherwise, the probability is calculated as the ratio of its support to the sum of supports of all the child nodes of its parent.

A snippet of the probability tree obtained in one of our experiments is shown in Fig. 3. Note that the figure only shows a snippet of the whole tree which also consists of a root node and other branches. Each node in the tree has a probability value indicating the likelihood that it would be visited next if its parent node had been visited. For example, it can be seen from the snippet that the probability of a person visiting the *microwaveoven* right after entering the tearoom is not high (7.34%). Furthermore, it is likely that if the *microwaveoven* has been visited, the *sink* would be visited next (64.62%).

The usefulness of a probability tree depends on how well the mined movement patterns represent people's spatial behaviours; this requires the number of movement sequences to be statistically large enough. The minimum number of movement sequences can vary depending on the nature of the monitored environment. In our experiment, we first mine movement patterns from the set of 287 movement sequences, and then continuously update the mined patterns as new movement sequences are captured and processed.

Algorithm 1 Algorithm to construct probability tree

Input:
  A set of movement patterns
  A set of movement sequences
Output: A probability tree representing the sequential movement behaviors
Procedure:
  Step 1: Form a probability tree from movement patterns, initializing and updating the support for each node.
  Step 2: For each POI, calculate the percentage that it was the first visited POI in movement sequences
  Step 3: Update the probability for every node except the root. For each of the node in the tree, if it is the root node, the probability of its child nodes is calculated using the percentage values calculated in **Step 2**; otherwise, the probability of a child node is the ratio of its support to the sum of supports of all nodes that share the same parent with it including itself.

5.2. Continuously updating movement patterns

As people's movement behaviors may change over time, there is a need to continuously update the movement pat-

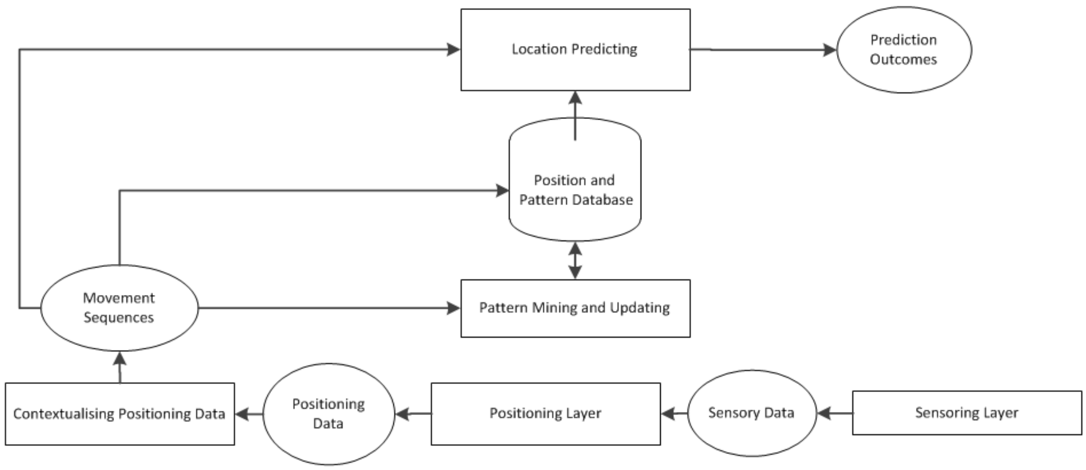

Figure 2. The prediction architecture.

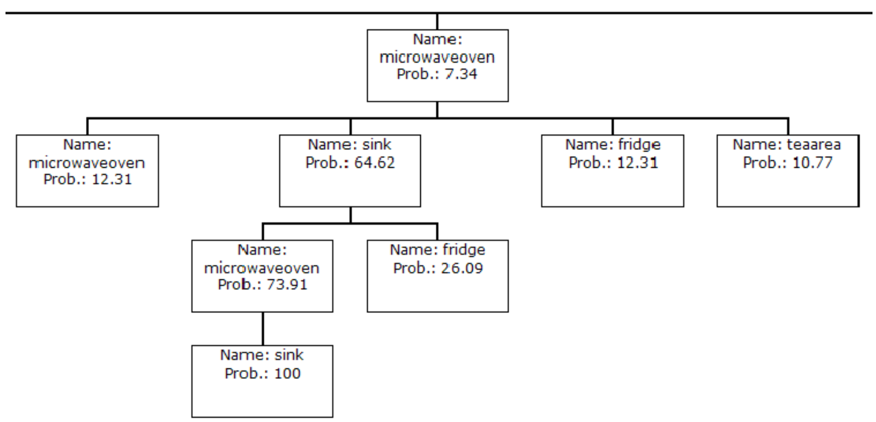

Figure 3. A snippet of a probability tree starting with the *microwaveoven* POI.

terns to reflect the current common spatial behaviors more accurately. The architecture shown in Fig. 2 allows such continuous updating to be achieved by feeding the newly captured movement sequences into the *Pattern Mining and Updating* layer, whose output is then stored in the database. Incremental sequential pattern mining has been studied by Zhang et al. [17], in which the mining efficiency is achieved by avoiding processing the bulk of the database using certain candidate pruning techniques. In this work, we extend the algorithm to mine immediate movement patterns, taking into consideration that a sequence of locations can be visited multiple times in long input movements. The support of a candidate sequence is counted only if certain conditions related to it are satisfied; details about how to count the support of an immediate pattern in a dataset can be found in [7].

5.3. Predicting next location

Having a probability tree to represent movement patterns, we can use it to predict movement trajectory as a person moves through the same area. This is achieved by comparing the current trajectory against the probability tree to find all matches. As each match may have different predictive powers and lead to different prediction outcomes, there is a need to allocate priority to each match. There exist a number of ways to allocate priority which could be in an ad-hoc manner. In this work, we investigate the use of 2 different estimation strategies to allocate priority to each match, these strategies are presented in Section 5.3.1. The algorithm to predict next location is then provided in Section 5.3.2.

5.3.1. Priority Allocation Strategy. A match of the current trajectory in the probability tree is a path in the tree whose sequence of nodes matches the entire current trajectory, or a sequence of consecutive POIs in the trajectory including the last POI. For example, consider the probability tree as in Fig. 3, assuming that the current trajectory is the sequence $< microwaveoven >$, there are 3 matches corresponding to the 3 *microwaveoven* nodes in the figure. If the current trajectory is $< microwaveoven, sink >$, then there are 2 matches corresponding to 2 paths containing $< microwaveoven, sink >$ in the figure, and there are 2 matches corresponding to 2 paths containing the $< sink >$; however, as the 2 matches corresponding to the $< sink >$ are contained by the 2 matches to the sequence $< microwaveoven, sink >$, there are only 2 matches overall. Furthermore, among these 2 matches, one has no next node, and as such is not used to predict the next location.

Each of the matches are then allocated a corresponding value indicating its priority or predictive power in making predictions for the next locations. In this work, we investigate the use of 2 different estimation strategies to allocate priority, namely: Equal Chance and Best Match. In the first strategy, the priority to different potential trajectory is allocated equally weight. In the second strategy, the priority is allocated proportionally to the length of the matching, i.e. greater weight is given to a long trajectory match. The Best Match strategy favors matches that share more common POIs with the current trajectory, and is more suitable to use when people's movements can be tracked accurately. On the other hand, when the movement data is noisy, the Equal Chance strategy may be more favorable as it ignores the matching lengths and treats all matches equally.

For example, assuming that the current trajectory is $< sink, microwaveoven >$ and the probability tree is as in Fig 3. There are 1 match to the sequence $< sink, microwaveoven >$, and 3 matches to the sequence $< microwaveoven >$. Among these matches, one match has no next node, and one (corresponding to the sequence $< microwaveoven >$) is contained in another (corresponding to the sequence $< sink, microwaveoven >$). As such there are 2 matches used for prediction, one corresponding to the sequence $< sink, microwaveoven >$ (we call this match as match A), and another corresponding to the sequence $< microwaveoven >$ (we call this match as match B). Using the Equal Chance strategy, the 2 matches would have equal predictive power which is 50%. The most likely next POI of match A is the $sink$ with the probability being: $50\% * 64.62\% = 32.31\%$; the most likely next POI of match B is the $sink$ with the probability being: $50\% * 100\% = 50\%$. Overall, the chance that the $sink$ would be visited next using the first strategy is $50\% + 32.31\% = 82.31\%$. Using the second strategy, the priority for each match is proportional to its length, as such the priority for match A is 66.67%, and that for match B is 33.33%. The chance of the $sink$ would be visited next is 76.41%

5.3.2. Next Location Prediction. The algorithm to predict the next location is presented in Algorithm 2. The inputs to the algorithm are the current trajectory, a probability tree, and boolean value indicating which priority allocation strategy should be used.

Algorithm 2 Algorithm to predict next location

Input:
A current trajectory
A probability tree
The priority allocation strategy
Output: A list of POIs and their corresponding probability to be the next location.
Procedure:
Step 1: If the current trajectory is blank, return POIs represented by child nodes of the tree root with corresponding probability as prediction results.
Step 2: Search for all the matches between the current trajectory and the probability tree.
Step 3: Allocate priority to each match according to the specified strategy.
Step 4: Compute prediction results by calling the **getNextLocation** subroutine.

In step 1, if the current trajectory is empty, we collect all child nodes of the tree root, and returns the corresponding POIs with their probability as the results.

In step 2, we loop until the current trajectory is empty. For each iteration of the loop, we find and save all matches to the current trajectory in the probability tree, and remove the first POI from the trajectory for the next search iteration.

In step 3, the priority of each match is computed depending on the strategy used. If the strategy is Equal Chance, the priority is allocated equally among the matches. Otherwise, the priory is allocated according to the matching length.

TABLE 1. SUMMARY OF MOVEMENT PATTERNS RELATED TO THE 2 VISITS TO THE $sink$ USING DIFFERENT VALUES OF THE MINING THRESHOLD $minSup$. THE LEFT COLUMN SHOWS DIFFERENT VALUES OF THE MINING THRESHOLD $minSup$. THE RIGHT COLUMN SHOWS THE MINED PATTERNS.

Mining threshold	Movement patterns related to the pattern $< sink, sink >$
15%	$< sink, sink >$
5%	$< sink, sink >$
	$< sink, microwaveoven, sink >$
	$< sink, teaarea, sink >$
2%	$< sink, sink >$
	$< sink, microwaveoven, sink >$
	$< sink, teaarea, sink >$
	$< sink, table, sink >$
	$< sink, fridge, sink >$

In step 4, the getNextLocation sub-routine (Algorithm 3) collects next locations, predicted using each match, and merges them together. We loop through each match and find the corresponding next locations. If a predicted location is not yet in the list of results, we add it to the list with the corresponding probability. Otherwise, we update its probability by summing the current probability with the one that has just been calculated.

6. Experiment

6.1. Experimental Settings

We conducted an experiment in a staff tearoom in a large organization. The experimental area contains a number of

TABLE 2. PREDICTION OUTCOMES USING DIFFERENT STRATEGIES. EACH ROW IN THE FIRST COLUMN SHOWS THE EMPLOYED STRATEGIES. THE SECOND, THIRD, AND FOURTH COLUMNS SHOWS THE PREDICTION OUTCOMES IN TERMS OF THE PERCENTAGES OF CORRECT PREDICTIONS, CORRESPONDING TO RESPECTIVELY WHEN THERE ARE 1, 2, AND 3 MOST PROBABLE PREDICTION RESULTS SELECTED IN EACH STRATEGY.

	Prediction of most probable POIs visited next		
	1 POI	2 POIs	3 POIs
Equal Chance	56.96%	78.48 %	97.47 %
Best Match	56.93 %	79.75 %	97.47 %
Random prediction	13.64 %	28.55 %	42.46 %

Algorithm 3 getNextLocation: Algorithm to get next locations from a list of matches

```
Input:
  A list of matches: matchingList
Output: A list of POIs and their corresponding probability to be the next location.
Procedure:
  results = empty list
  for each match in matchingList:
    childNodes = match.getLastNode().getChildNodes()
    for each aNode in childNodes:
      if ∃ atuple = {aPOI, prob} ∈ results such that
          aPOI == aNode.name:
        atuple.prob += match.priority * aNode.prob
      else:
        results.add({aNode.name,
                match.priority * aNode.prob})
    end for
  end for
  return results;
```

POIs, as depicted in Fig. 1. A camera was installed in the room to capture video stream, the images were processed by a positioning software to produce positioning results in terms of Cartesian coordinates. This software is built in-house and is tested to be accurate to within 25cm. For each POI in the experimental area, we defined an area that people usually stay within if they are utilizing that POI; for example, in Fig. 1, the green squares indicate POIs, and the red square shows an example of the area corresponding to the *microwaveoven* POI.

We extracted sequences of POIs by detecting if a positioning result falls within pre-defined areas. Specifically, a person is considered as visiting a POI if he or she stay within the corresponding POI area for longer than a minimum threshold. Such a threshold mechanism is used to ensure that the person actually visited the POI rather than just passed through it. Another threshold mechanism is used to guard against possible temporarily inaccurate positioning estimation, in which a person is considered as having left a POI if the leave duration exceeds a minimum threshold. Details about the POI extraction process can be found in [7].

6.2. Experimental Results

We captured seven weeks of data, corresponding to 317 movement sequences or trips, in the experiment. We used six weeks of data (corresponding to 287 movement sequences) to mine movement patterns, and one week of data (corresponding to 30 movement sequences) to validate the next location prediction algorithms.

Table 1 summarizes some of the movement patterns, related to the 2 visits to the $sink$, mined using various values for the mining threshold. It can be seen that the lower the threshold, the more patterns detected. For example, at 15% threshold, there is only 1 pattern detected, while at 2% threshold, there are 5 patterns detected. This means that with the lower threshold, we obtain more details about the movement behaviors. In this experiment, we mine patterns detected at 2% threshold to construct the probability tree.

The constructed probability tree consists of 58 nodes (including the root) distributed across 5 levels. Due to space limits, we do not present the tree in this paper, with just a snippet is shown in Fig. 3. The shortest branch corresponds to the *cabinet* which spreads over only two levels including the root level. The 3 branches, having the highest number of levels, start with a visit to *microwaveoven*, *sink*, or *teaarea*. It can also be seen from the tree that the *sink* the most likely POI visited first by people when they enter the tearoom, while the *cabinet* is the least popular first destination.

We used 30 trips, out of 317 trips, to validate the proposed prediction algorithms. For each of the 30 trips, the prediction was triggered every time a tracked target visits a POI. Overall, there were 79 predictions made. The prediction results are presented in Table 2. There were 3 prediction strategies employed in the experiment including: Equal Chance, Best Match, and random prediction. The first 2 strategies are presented in Section 5.3, the random prediction strategy is the control test. It does not possess any knowledge of historical movement data. It randomly picks any POI from the set of possible POIs, assuming that all of the next POIs are equally probable.

For each of the prediction strategies, Table 2 presents their prediction outcomes, corresponding to how many most probable outcomes are selected as prediction results for a prediction. For example, the second column shows the results where only the most probable POI among all possible POIs are selected as the prediction result; if this POI is the same as the next POI visited by the tracked target then the prediction is considered as correct. Similarly, the third and last columns show the successful prediction rates where there are, respectively, 2 and 3 most probable POIs selected as the prediction results.

It can be seen from the table that the successful prediction rates increase when more POIs are selected as prediction results. This follows the intuition that the more POIs selected, the higher the chance of getting correct predictions.

The table shows that the Equal Chance and Best Match strategies produce consistently better predictions than the random prediction, indicating that using knowledge of historical movement information can produce more accurate predictions than predicting with no prior knowledge. The prediction results using the first 2 strategies are similar, this could be because in our experiment, people's movements were short in length and the next POI visited is strongly related to the last visited POI.

7. Conclusion

In this work, we describe new algorithms to mine movement patterns from movement sequences. Movement patterns are sequences of locations that are frequently visited. In order to obtain accurate and up-to-date movement patterns, we continuously capture peoples' movements, processing large amount of data in real-time. We also construct probability tree as we update movement patterns. The probability tree is then used to predict the trajectory of a person, given their current trajectory and the knowledge of their movement patterns.

We have shown that by mining the patterns, we can represent people's frequent movement behaviors. We demonstrate the usefulness of the mined patterns by using them to predict the next locations a person would visit. Our empirical study shows that accurate next location prediction can be achieved using this approach. For example, the accuracy of a next POI prediction is more than 56%, and the accuracy increases to more than 78% when 2 most probable POIs are selected. In comparison to the control tests, the random prediction without any knowledge of historical movement data only achieved 13.64% and 28.55% respectively.

Our future work is to investigate the use of temporal information to achieve higher prediction accuracy. Besides, our approach does not consider cases where POI areas can be overlapping; for example, in some environments, *microwaveoven* may be put on top of *fridge*. Our future work is to enhance the practicality of our work by dealing with such situations. Furthermore, our prediction methods rely on the accuracy of the positioning system in use. In some applications, positioning sensors could be error-prone; this could result in inaccurate movement sequences, for e.g. when the sensors fail to detect that a POI has been visited, or falsely report a visit when there is none. In these situations, cleaning techniques, such as those proposed in [1], [2], should be applied to correct movement sequences before they are fed into our location prediction mechanism.

References

[1] A. I. Baba, M. Jaeger, H. Lu, T. B. Pedersen, W.-S. Ku, and X. Xie, "Learning-based cleansing for indoor rfid data," in *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016, pp. 925–936.

[2] B. Fazzinga, S. Flesca, F. Furfaro, and F. Parisi, "Exploiting integrity constraints for cleaning trajectories of rfid-monitored objects," *ACM Transactions on Database Systems (TODS)*, vol. 41, no. 4, p. 24, 2016.

[3] D. Gussfield, "Algorithms on strings, trees, and sequences," *Computer Science and Computional Biology (Cambrigde, 1999)*, 1997.

[4] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.

[5] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proceedings of the 17th international conference on data engineering*, 2001, pp. 215–224.

[6] J. Krumm, R. Gruen, and D. Delling, "From destination prediction to route prediction," *Journal of Location Based Services*, vol. 7, no. 2, pp. 98–120, 2013.

[7] L. D. M. Lam, A. Tang, and J. Grundy, Technical Report - Indoor Spatial Positioning Analysis. [Online]. Available: http://opax.swin.edu.au/~ldlam/IndoorSpatialPositioningAnalysis.pdf

[8] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data mining and knowledge discovery*, vol. 1, no. 3, pp. 259–289, 1997.

[9] M. Morzy, "Mining frequent trajectories of moving objects for location prediction," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 667–680.

[10] L. Nguyen, H.-T. Cheng, P. Wu, S. Buthpitiya, and Y. Zhang, "Pnlum: System for prediction of next location for users with mobility," in *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated challenge*, vol. 2, no. 2, 2012.

[11] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Comparison of different methods for next location prediction," in *Euro-Par 2006 Parallel Processing*. Springer, 2006, pp. 909–918.

[12] J. Petzold, A. Pietzowski, F. Bagci, W. Trumler, and T. Ungerer, "Prediction of indoor movements using bayesian networks," in *International Symposium on Location-and Context-Awareness*. Springer, 2005, pp. 211–222.

[13] L. Vintan, A. Gellert, J. Petzold, and T. Ungerer, "Person movement prediction using neural networks," in *First Workshop on Modeling and Retrieval of Context*, vol. 114, no. 4, 2004, pp. 1–12.

[14] G. Yavaş, D. Katsaros, Ö. Ulusoy, and Y. Manolopoulos, "A data mining approach for location prediction in mobile environments," *Data & Knowledge Engineering*, vol. 54, no. 2, pp. 121–146, 2005.

[15] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 34–43.

[16] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Mining user similarity from semantic trajectories," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*. ACM, 2010, pp. 19–26.

[17] M. Zhang, B. Kao, D. Cheung, and C.-L. Yip, "Efficient algorithms for incremental update of frequent sequences," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 186–197.

[18] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: A survey," *Technical Report CAIS Nayang Technological University Singapore*, pp. 1–26, 2003.