УДК 004.75

Petrashenko A., Ph.D.,
Zamiatin M., Ph.D.,
Pazii A.

A LOCATION PREDICTION APPROACH OF MOVING OBJECTS

The National Technical University of Ukraine "I.Sikorsky Kyiv Polytechnic Institute"

petrashenko@gmail.com
dsz@ukr.net
anny290994@gmail.com

Growing popularity of location based services is leading to an increasing volume of mobility data. In this paper we introduce a data mining approach to the problem of predicting the next location of a moving object with a certain level of accuracy. We use apriori algorithm to build a probabilistic model of future object location. The experiments have demonstrated that our technique gives reasonably accurate prediction

Keywords: Moving Object, Data Model, Spatio-Temporal Analysis, Apriori algorithm

Introduction

In recent years, the increasing number of mobile devices with the use of wireless communication, such as Bluetooth, Wi-Fi and GPRS, inasmuch as, advances in positioning technology compel manufacturers to equip their devices with positioning sensors that utilize Global Positioning System (GPS) to provide accurate location of a device. This means that the movement of people or vehicles within a given area can be observed and collected by the wireless network infrastructures. In order to fully exploit the possibilities offered by location-aware services (navigational services, traffic management and location-based advertising), it is crucial to determine the current position of a moving object at any given point in time.

Due to the unreliable nature of mobile devices (such as power supply shortage of a moving object) and the limitations of the positioning systems (such as signal congestions, signal losses due to natural phenomena, or the existence of urban canyons), the location of an object is often unknown for a long period of time. Whenever the location of a moving object is unknown, a robust method of possible location prediction of a moving object is required in order to anticipate or prefetch possible services in the next location.

We want to show that data mining techniques can be successfully used for location prediction. The idea is to transform frequent trajectories into movement rules, because individuals tend to follow common paths (people go to work every day by similar routes, and public transport crosses similar routes in different time periods). In order to predict the location of a moving object we compute for each possible location the probability of prediction correctness based on the support and confidence of discovered movement rules.

Associattion analysis

We use methodology known as association analysis, which is useful for discovering interesting relationships hidden in large data sets. Association rule mining is an important data mining model studied extensively by the database and data mining community. The uncovered relationships can be represented in the form of association rules or sets of frequent items. There are two key issues that need to be addressed. First, discovering patterns from a large transaction data set can be computationally expensive. Second, some of the discovered patterns are potentially spurious because they may happen simply by chance.

Let $I = \{i_1,i_2,. . .,i_d\}$ be the set of all items and $T = \{t_1,t_2,...,t_N\}$ be the set of all transactions. Each transaction t_i contains a subset of items chosen from I. The transaction width is defined as the number of items present in a transaction. A transaction t_j is said to contain an itemset X if X is a subset of t_j.

An important property of an itemset is its support count, which refers to the number of transactions that contain a particular itemset. Mathematically, the support count, σ(X), for an itemset X can be stated as follows:

$$\sigma(X) = |\{t_i|X \subseteq t_i,\, t_i \in T\}|,$$

where the symbol | · | denotes the number of elements in a set.

Association Rule: An association rule is an implication expression of the form

$$\text{Support},\ s(X \longrightarrow Y) \ = \ \frac{\sigma(X \cup Y)}{N};$$

$$\text{Confidence},\ c(X \longrightarrow Y) \ = \ \frac{\sigma(X \cup Y)}{\sigma(X)}.$$

$X \rightarrow Y$, where X and Y are disjoint itemsets,

i.e., $X \cap Y = \emptyset$. The strength of an

association rule can be measured in terms of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in Y appear in transactions that contain X. The formal definitions of these metrics are:

Definition 1. (Association Rule Discovery). Given a set of transactions T, find all the rules having support ≥ minsup and confidence ≥ minconf, where minsup and minconf are the corresponding support and confidence thresholds.

A common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks:

1. Frequent Itemset Generation, whose objective is to find all the itemsets that satisfy the minsup threshold. These itemsets are called frequent itemsets.

2. Rule Generation, whose objective is to extract all the high-confidence rules from the frequent itemsets found in the previous step. These rules are called strong rules.

A basic idea of A-Priori is all nonempty subsets of a frequent itemset must also be frequent. According to the idea, a two-step process is employed as generate and prune actions:

1. Candidate generation: This step is to generate a new set of candidate k-itemsets by join k-1 itemset itself found in the previous iteration.

2. Candidate Pruning: This step eliminates infrequent candidate k-itemsets depending on the support-based pruning strategy.

A-Priori algorithm (Figure 1): Find frequent itemsets using an iterative level-wise approach based on candidate generation

1. A transactions in database D are scanned to determine frequent 1-itemsets,

L_1 by comparing with minsup

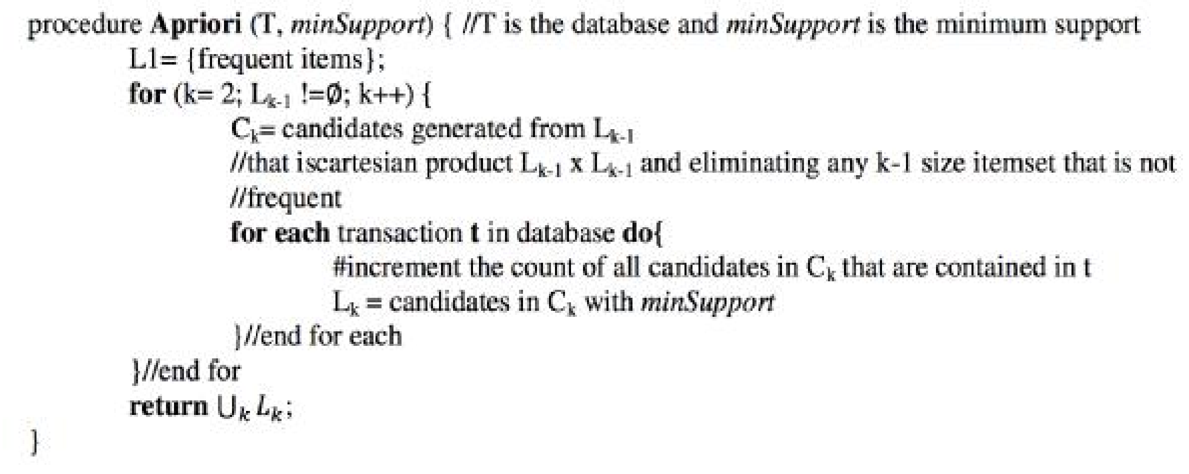
2. Generate candidate k-itemset C_k from the two joining k-1 itemsets, L_{k-1} and remove its infrequent subset.

3. Scan D to get support count for each k-itemsets, C_k .

4. The set of frequent k-itemsets, L_k is then determined. L_k results from support count of candidate k-1 itemset minsup

5. Back to step 2 until there is no candidate k+1 itemsets, C_{k+1}

6. Extract the frequent k-itemsets, $L = L_k$.



```
procedure Apriori (T, minSupport) { //T is the database and minSupport is the minimum support
        L1= {frequent items};
        for (k= 2; $L_{k-1}$ !=∅; k++) {
                $C_k$= candidates generated from $L_{k-1}$
                //that iscartesian product $L_{k-1}$ x $L_{k-1}$ and eliminating any k-1 size itemset that is not
                //frequent
                for each transaction t in database do{
                        #increment the count of all candidates in $C_k$ that are contained in t
                        $L_k$ = candidates in $C_k$ with minSupport
                }//end for each
        }//end for
        return $\bigcup_k L_k$;
}
```

Fig.1. Apriori Algorithm Pseudocode

It must be assumed that we found the frequent itemsets from transaction in a database D that meet a threshold of support, and extract support calculated for each of these itemsets. Thus, we can find all association rules as follows:

1. For each frequent itemset L, generate all nonempty subsets of L.

$$s \to (L\text{-}s)\ if \frac{supportcount(L)}{supportcount(s)} \geqslant minconf$$

2. For each nonempty subsets of L, output the rule.

Prediction strategy

The idea is to discover movement rules with support and confidence greater than user-defined thresholds of minsup and minconf, respectively. And match movement rules against the trajectory of a moving [2]. In order to take only the portion of trajectories we select a spatial area and a time period, then algorithm extracts from the selected trajectories the frequent movement patterns. After, we use association rules as predictive rules in rule based classifiers to define regions that are frequently visited and typical travel time.

A trajectory of a moving object is a sequence of time-stamped locations, representing the traces collected by some wireless/mobility infrastructure (GSM, GPS, etc). The location is abstracted by using ordinary Cartesian coordinates, as formally stated by the following definition:

Definition 2. A Trajectory or spatio-temporal sequence is a sequence of triples

$T =< x_0,y_0,t_0 >,...,< x_n,y_n,t_n >$, where

t_i (i = 0 . . . n) denotes a timestamp

such that $\forall 0<i<n$ $t_i < t_i+1$ and (x_i, y_i) are

points in R^2.

Intuitively, each triple $< x_i , y_i , t_i >$ indicates that the object is in the position (x_i,y_i) at time t_i [3]. Consider an example in Figure 2, where a user has two trajectory patterns (i.e., T1 and T2). In T1, the user usually has the routing path from his home (i.e., A) to his study place (i.e., C) alone with the main street (i.e., B). As can be seen in Figure 2, our trajectory model includes not only travel time information between two hot regions but also appearing time information about when this user appears. Furthermore, since these two sequential patterns have two common regions (i.e., A and B) these sequential patterns could be represented as a compact model in which common regions (i.e., A and B) are shared by two trajectory patterns. Traditionally, sequential patterns will have supports that indicate their appearing counts among trajectories. In this example, T1 (i.e., A → B → C) has higher support than T2. Assume that recent movements are A and B, and this user travels at region B around 10:00am. From these two sequential patterns, the next location of this user is C since T1 has higher support. However, one could imply that if a user moves to C, the time will be 11:00 (i.e., the sum of the current time and time interval between B and C). The appearing probability at region C at 11:00 via the routing path A → B is zero. On the other hand, this user may appear at region E at 12:00 with the probability larger than zero. Thus, in this case, the next location of the user is region E, showing that by exploring both spatial and temporal information, the accuracy of location prediction could be enhanced [1].

After extracting the frequent regions, each raw trajectory is transformed into a sequence of frequent regions. Such a sequence is called a region-based moving sequence and defined as following:

Definition 3. A Region-based Moving Sequence is a sequence of frequent regions

$S_r = \{(r_1, t_1), (r_2, t_2), ..., (r_n, t_m)\}$ with time constraint $t_1 < t_2 < ... < t_m$, where (r_i , t_j) indicates the object visits the frequent region r_i at timestamp t_j .Trajectories are transformed into region-based moving sequences:

$\{(c_{16} , t_1), (c_{26}, t_2), (c_{25}, t_3), (c_{35}, t_4), (c_{45}, t_5), (c_{42}, t_8), (c_{41}, t_9), (c_{31}, t_{10}), (c_{30}, t_{11})\}$.

There are three steps as follows:

Step 1. Frequent Region Discovery: In this step, we extract frequent regions from a set of trajectories to capture movement behaviors. A frequent region contains a sufficient number of trajectories whose data points are within the corresponding region.

Step 2. Trajectory Transformation: According to the set of frequent regions determined by Step 1, each raw trajectory is transformed into a region-based moving sequence. Location points that are not in frequent regions will be regarded as noise.

Step 3. In location prediction module, given current movements of an object and a query future time, we propose to use a location prediction algorithm to traverse the object and estimate the future location at the query time [1].

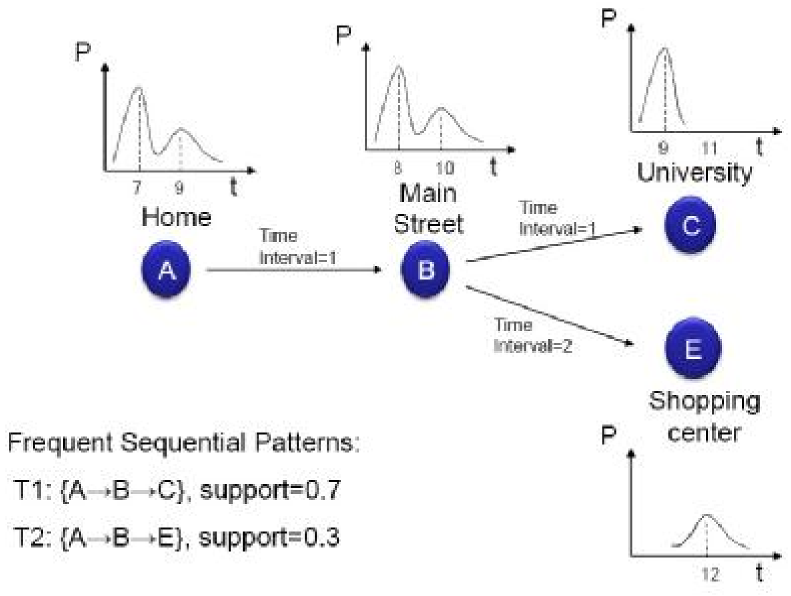
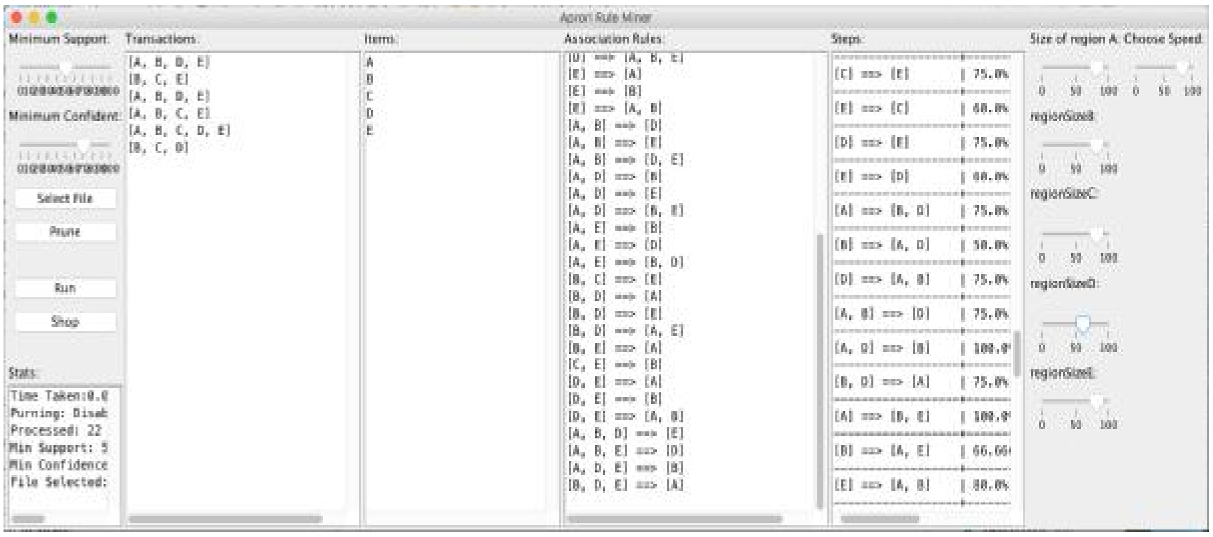


Fig. 2. Exploring Spatial-Temporal Trajectory Model for Location Prediction [1]



```
Transactions:
[A, B, D, E]
[B, C, E]
[A, B, D, E]
[A, B, C, E]
[A, B, C, D, E]
[B, C, D]

Items:
A
B
C
D
E

Stats:
Time Taken:0.0
Purning: Disab
Processed: 22
Min Support: 5
Min Confidence
File Selected:

Association Rules:
[D] ==> [A, B, E]
[E] ==> [A]
[E] ==> [B]
[E] ==> [A, B]
[A, B] ==> [D]
[A, B] ==> [E]
[A, B] ==> [D, E]
[A, D] ==> [B]
[A, D] ==> [E]
[A, D] ==> [B, E]
[A, E] ==> [B]
[A, E] ==> [D]
[A, E] ==> [B, D]
[B, C] ==> [E]
[B, D] ==> [A]
[B, D] ==> [E]
[B, D] ==> [A, E]
[B, E] ==> [A]
[C, E] ==> [B]
[D, E] ==> [A]
[D, E] ==> [B]
[D, E] ==> [A, B]
[A, B, D] ==> [E]
[A, B, E] ==> [D]
[A, D, E] ==> [B]
[B, D, E] ==> [A]

Steps:
[C] ==> [E]          | 75.0%
[E] ==> [C]          | 60.0%
[D] ==> [E]          | 75.0%
[E] ==> [D]          | 60.0%
[A] ==> [B, D]       | 75.0%
[B] ==> [A, D]       | 50.0%
[D] ==> [A, B]       | 75.0%
[A, B] ==> [D]       | 75.0%
[A, D] ==> [B]       | 100.0
[B, D] ==> [A]       | 75.0%
[A] ==> [B, E]       | 100.0
[B] ==> [A, E]       | 66.66
[E] ==> [A, B]       | 80.0%
```

Fig. 3. Exploring Spatial-Temporal Trajectory Model for Location Prediction [1]

Conclusion and results

In this paper, we presented approach to predict an object's future locations which is based on modified a-priori algorithm. We focus on that how to discover frequent movement patterns and manage these patterns to answer predictive queries. Also, was added opportunity to analyze the dependence on the size of the region and the time of movement of the object as presented at Figure 3.

For our experiment, we used two types of dataset: SmallDataSet.dat and LargeDataSet.dat. The results presented at Figure 4 (a) were compiled for the large dataset ~8k rows, results presented at Figure 4(b) were compiled for the small dataset. The general trend was that as the support level got lower and the data set bigger, the benefits of enabling pruning made a substantial difference it terms of candidate set(s) processed.

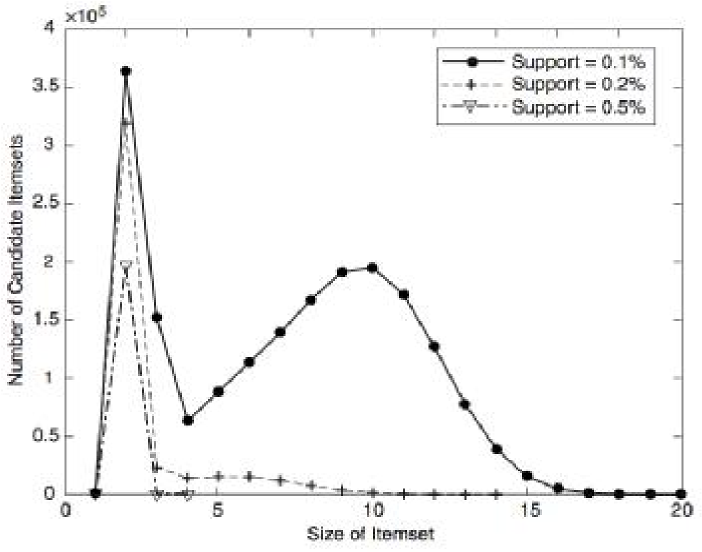
Time Taken:1.038 seconds Pruning: Enabled Processed: 308 Candidate set(s) Min Support: 50.0 % Min Confidence: 70.0 % File Selected: LargeDataSetSample1.dat	Time Taken:0.961 seconds Pruning: Disabled Processed: 332 Candidate set(s) Min Support: 50.0 % Min Confidence: 70.0 % File Selected: LargeDataSetSample1.dat
(a) Results for the large dataset	
Time Taken:0.004 seconds Pruning: Enabled Processed: 21 Candidate set(s) Min Support: 50.0 % Min Confidence: 70.0 % File Selected: SmallDataSetSample1.dat	Time Taken:0.007 seconds Pruning: Disabled Processed: 22 Candidate set(s) Min Support: 50.0 % Min Confidence: 70.0 % File Selected: SmallDataSetSample1.dat
(b) Results for the small dataset	

Fig. 4. Results

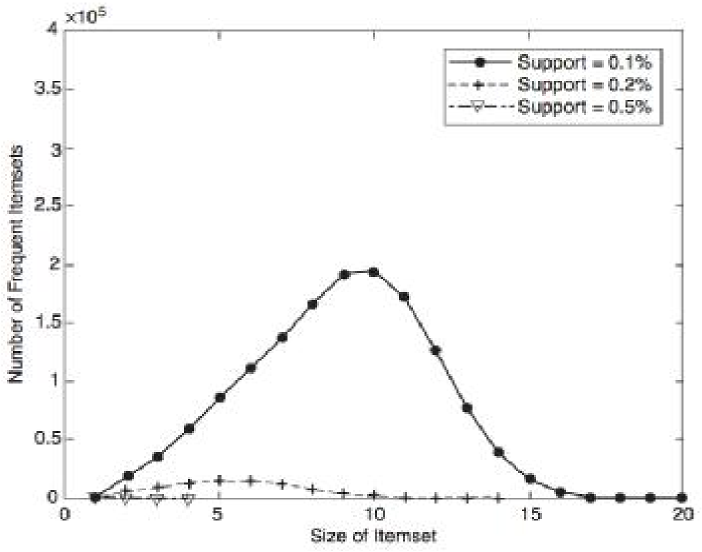
Support Threshold. Lowering the support threshold often results in more itemsets being declared as frequent. This has an adverse effect on the computational complexity of the algorithm because more candidate itemsets must be generated and counted, as shown in Figure5. The maximum size of frequent itemsets also tends to increase with lower support thresholds. As the maximum size of the frequent itemsets increases, the algorithm will need to make more passes over the data set.

Average Transaction. Width for dense data sets, the average transaction width can be very large. This affects the complexity of the Apriori algorithm in two ways. First, the maximum size of frequent itemsets tends to increase as the average transaction width increases. As a result, more candidate itemsets must be examined during candidate generation and support counting, as illustrated in Figure 6. Second, as the transaction width increases, more itemsets are contained in the transaction. This will increase the number of hash tree traversals performed during support counting.

Our experiments demonstrated that our technique gives reasonably accurate prediction and allows end users to tune the algorithm using a set of thresholds.

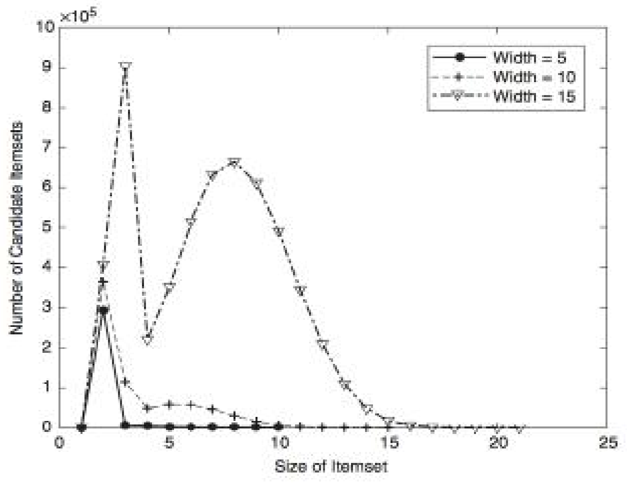


(a) Number of candidate itemsets.

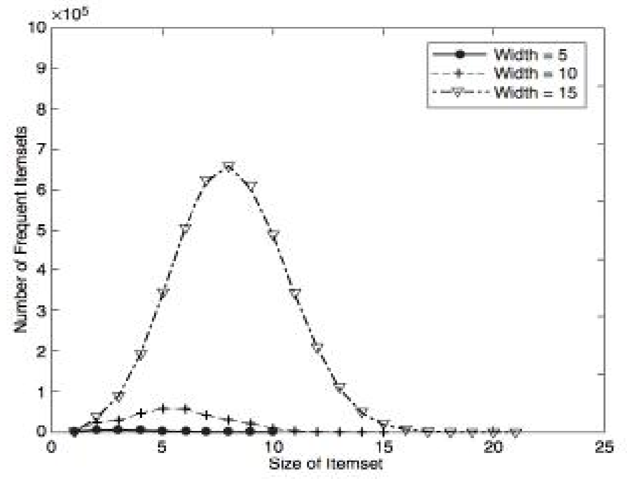


(b) Number of frequent itemsets.

Fig.5. Effect of support threshold on the number of candidate and frequent itemsets .



(a) Number of candidate itemsets.



(b) Number of Frequent Itemsets.

Fig.6 Effect of average transaction width on the number of candidate and frequent itemsets

References

1. P.-R. Lei, T.-J. Shen, W.-C. Peng, and I.-J. Su, "Exploring spatial-temporal trajectory model for location prediction," in 12th IEEE International Conference on Mobile Data Management, MDM 2011, Luleå, Sweden, June 6-9, 2011, Volume 1, 2011, pp. 58-67.

2. Mikołaj Morzy, Mining Frequent Trajectories of Moving Objects for Location Prediction, Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition, July 18-20, 2007, Leipzig, Germany

3. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: WhereNext: a location predictor on trajectory pattern mining. In: Proceedings of SIGKDD 2009, pp. 637–646. ACM, New York (2009)

4. H. Jeung, Q. Liu, H. T. Shen, and X. Zhou, “A Hybrid Prediction Model for Moving Objects,” in Proc. of ICDE, 2008, pp. 70–79.

Article to the Editor 29.05.2017