A Frequent Pattern based Prediction Model for Moving Objects

Juyoung Kang and Hwan-Seung Yong

Dept. of CSE, Ewha Womans University, Seoul, 120-750 Korea

Abstraction

Huge amounts of moving object data have been collected with the advances in wireless communication and positioning technologies. Trajectory patterns extracted from historical trajectories of moving objects can reveal important knowledge about movement behavior for high quality LBS services, especially for location prediction. Existing approaches cannot forecast accurate locations in the distant future since they use motion functions which emphasize the recent movements of objects. In this paper, we propose a new approach which utilizes frequent trajectory patterns to predict location. Using line simplification and clustering, the proposed method simplifies trajectories and clusters them into spatio-temporally meaningful regions. After original trajectories are discretized into the sequences using regions, trajectory patterns from discretized sequences are extracted using a prefix-based projection approach. Then, we construct a tree-structured prediction model using these patterns, which allows an efficient indexing of discovered patterns to find the best match. We experimentally analyze that the proposed method's efficiency in discovering trajectory patterns, predicting a future location accurately even though the query time is far in the future.

Key words:

Spatio-temporal data mining, Location prediction, Trajectory pattern mining

1. Introduction

With the advances of positioning technology and wireless communication, an accurate location of a mobile device can be provided and utilized in various kinds of location based services. For example, users with GPS-equipped cell phones can log their positions at a fixed time interval and transmit it to the server of the wireless carrier. Reliable and high quality LBS services, such as traffic management or route finding system, require not only the current positions but also the future locations of users. Although the accuracy and the availability of positioning data have been increased, we cannot track an object for a long time due to the failures in GPS systems and mobile devices as well as the limitations of wireless networks. When the current location of a moving object is not available, a reliable method for location prediction of a moving object is required [1]. Since existing methods adopt mathematical functions based on objects' recent movements, they may not provide an accurate prediction results for a location in the distant future. Motion functions that are used for location prediction cannot represent an object's movement since it moves in a far more complex way in reality. A sudden detour from the path or complicated movement along the road networks cannot be captured by linear or nonlinear motion functions [2].

Since moving objects such as mobile users or vehicles often go along similar routes, we can give a reasonable answer to the prediction query about the object's location, if the movement patterns are known in advance. Data mining techniques can be used to discover spatio-temporal regularities in trajectories. Existing studies on discovering trajectory patterns simply discretize spatial and temporal properties into location symbols based on a fixed size grid. Then, frequent sequential patterns are extracted from the sequences of discretized location symbols. There are some limitations in these approaches. Due to the improper cell size, hidden patterns in trajectories may be lost during the discretization step. Moreover, the redundant appearance of the same symbols in the discretized sequences, which means the duration value between two different locations, not only hinders the efficient processing of data, but also decreases the interpretability of extracted patterns [3].

In order to tackle this problem, we introduce a compact representation of trajectories, which approximates the movements of objects into spatio-temporal regions. In this paper, we address the problem of discovering frequent trajectory patterns using this data abstraction as well as predicting the future location of an object based on the extracted patterns. Our approach first approximates original trajectories into the simplified sequences and finds frequent patterns from the sequences. Finally, a prediction model is constructed based on these patterns and the best match for an object's location is determined among all the possible paths in the model.

The rest of this paper is organized as follows. In Section 2, we describe related works. The problem of extracting frequent trajectory patterns and predicting future location based on patterns is represented in Section 3. We present experimental evaluations of our approach in section 4. Finally, Section 5 provides concluding remarks and discussions for future works.

2. Related Works

There have been many studies that address the problem of predicting an object's future location. Efficient access

Manuscript received March 5, 2010
Manuscript revised March 20, 2010

methods for predictive query processing have been proposed in [4] and [5]. They use motion functions to estimate objects' future locations that are based on the recent movements of objects. Although Recursive Motion Function (RMF) [5] is known as the most accurate one among the existing methods, it cannot give us an accurate answer when the query time is far from the current time.

In the recent years, several methods for predicting future locations based on frequent patterns have been proposed. In [6], a mining algorithm for predicting user movements in a PCS (Personal Communication Systems) network was proposed. They define the mobility pattern as a sequence of cells and mine frequent paths based on sequential pattern mining. Morzy applied the PrefixSpan algorithm to predict the location of a moving object [1]. Since he assessed the extracted rules based on the support and confidence of the rules, spatio-temporal closeness of the rule to the given query trajectory was not considered in the evaluation process. In [2], a hybrid approach, which estimates an object's future location, was proposed by Jeung et al. They selectively used a motion function for near future predictions or a pattern based prediction for a query of the distant future. In [7], Giannotti et al pursued a similar goal. They propose a method to predict the next location of a moving object. They utilized movement patterns named Trajectory Patterns, which were proposed in their previous work. Although their approach is similar to ours in representing spatio-temporal movements in an abstracted way, they require more complex spatio-temp -oral computations for estimating prediction candidates.

3. A Prediction Model based on Frequent Patterns

Mainly, our approach for predicting objects' future locations is divided into two sub-problems: (i) mining frequent trajectory patterns and (ii) location prediction based on the extracted patterns. This section first defines the problem of mining spatio-temporal patterns in trajectory data and proposes a mining algorithm which uses the line simplification and clustering. Then, a tree structured prediction model that is constructed based on the extracted patterns will be described. Finally, we present a prediction strategy for a new trajectory.

3.1 Mining Frequent Trajectory Patterns

A trajectory of a moving object is a temporally ordered sequence over a long history, consisting of spatial locations that are measured in 2-dimensional coordinates at each time-stamp. We can represent a spatio-temporal sequence S as a consecutive set of location measurements, $S = \{(x_1, y_1, t_1),(x_2, y_2, t_2),..., (x_n, y_n, t_n)$, where (x_i, y_i) is the location of object at timestamps t_i, $(t_i < t_{i+1})$. In order to mine the frequent patterns based on the sequential pattern mining approach, continuous spatial and temporal values should be discretized prior to the mining process. To discretize trajectory data, each (x_i, y_i) at timestamp t_i is transformed to the *id* of the spatial region describing the object's location. Since the interval between consecutive timestamps is fixed, the sequence is converted to a generalized sequence of location symbols "l_1l_2…" and temporal properties of movements, therefore, are abstracted into redundant symbols and their sequential order [8]. In general, the data space is partitioned into a fixed size grid or cells based on communication infrastructure, and thus location symbols consist of the IDs of the cells. Although this is simple and intuitive, we may not obtain satisfactory results in finding spatio-temporal patterns for several reasons. First, an inappropriate cell size results in losing some patterns during the discretization. Second, since the performance of the sequential mining process is closely associated with the length of sequence and the number of different items appearing in the sequence, redundancy of the same symbol deteriorates the mining performance. Third, inaccurate and unintuitive patterns could be derived from data. Actually, $\{A…AB…BC…C\}$ has a different temporal meaning compared to $\{ABC\}$. However, if the former is found to be a frequent pattern, $\{ABC\}$ will also be frequent irrelevant to its actual frequency in the database. Thus, we need to represent the trajectory in a better way that incorporates temporal constraints. Data space can be partitioned into disjointed areas which represent meaningful spatio-temporal changes in objects' movements. If we represent original trajectories as sequences of these areas, a spatio-temporal pattern can be defined as a pattern $ST = \{(R_1,d_1), (R_2,d_2),…,(R_n,d_n)\}$, where R_i is a region that spatially approximates points between P_{li} and P_{ri} in original trajectory T, and d_i is a duration of an object's movements within the corresponding region. Fig. 1 shows an example of a trajectory pattern which approximates objects' movements into a 3-dimensional region. Finally,

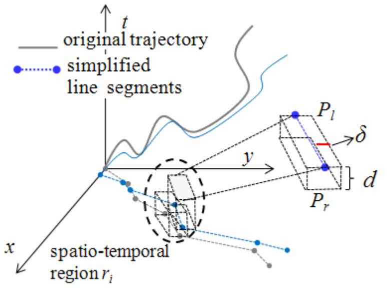


Fig. 1. A spatio-temporal region

we can reframe the problem of mining frequent trajectory patterns into a problem of discovering all frequent sequential patterns from sequences of these regions.

In order to discover spatio-temporal patterns from trajectories, we propose a pattern mining method based on line simplification and clustering [3]. The data space needs to be partitioned into spatio-temporally meaningful regions by abstracting spatio-temporal properties in order to improve efficiency of the mining process. To address this problem, we first summarize trajectories into their approximations using line simplification. Line simplification is a method for compressing polylines within a deterministic error bound [9]. Starting from the p_1 and p_n of a whole trajectory, the algorithm abstracts it into line segments which approximate all the original points within the corresponding segment, such that the perpendicular distance from the centerline of the segment is, at most δ. To represent the segments, we construct feature vectors based on the simplified segments by incorporating temporal constraints and then normalize them to equalize the importance of all features. Next, we cluster similar segments into regional groups to partition the data space appropriately. We consider the preclustering phase of BIRCH [10] as a segments clustering step, which has linear time complexity in input size and stores a summary of data in a compact tree structure, called the CF-tree. As a result, spatio-temporal segments are grouped into clusters which divide the data space into disjointed groups. Then original trajectories are discretized into sequences of cluster-IDs into which the simplified segments fall in. We adopt a depth-first search based method extending from [11] to discover frequent trajectory patterns from these sequences. By scanning the sequence database, all single region ids with a support count of greater than given the *min_sup* are found as 1-length frequent patter. Starting from the discovered 1-length regions, we apply a variant of a *prefix*-projection algorithm to discover longer ones, as shown in the algorithm below. The algorithm extends prefixes by a depth-first traversal and iteratively generates sub-projections with new prefixes until the condition is met. Details about the frequent pattern mining in this subsection are described in the previous work of the authors [3].

```
Algorithm Mining ST-patterns
Input : Trajectory database $T$, a simplification threshold $\delta$, a
clustering threshold $\varepsilon$
Output : A frequent pattern tree based on trajectory patterns
begin
    $T'$ := Line_Simplification($T, \delta$);
    $V$ := FeatureVector_Construction($T'$) ;
    $R$ := Discovering_STRegions($V, \varepsilon$);
    $n$ := 1;
    $F$ := Pattern_Extraction($\langle\rangle$, $n$, $R$);
$FPT$ := Pruning_Rules($F$, $min\_conf$);
return $FPT$;
End
```

```
Algorithm Predicting Locations($t$, $q$, $FPT$, $CF$-$tree$)
Input : A new trajectory $t$, a query time $q$, a frequent pattern
tree FPT , $CF$-$tree$ from the mining phase
Output : A best score prediction result
    $t'$:= Discretize($t$, $CF$-$tree$);
    $S$ := {$\alpha$–projected patterns for all patterns in $F_\alpha$ }
    for each $path \in FPT$ {
        $score$ := Calculate_Score($path$, $t'$, )
        if $score$ is the best
            $prediction$ = Compute_Prediction($path$, $q$)
}
return $prediction$;
End
```

3.2 Prediction Model for Moving Objects

When the frequent trajectory patterns are discovered, association rules are generated from the patterns. We can adopt the idea of a rule-based classifier to build a prediction model. Predicting the next location of a new trajectory is merely a problem of matching it to the rule's antecedents and finding the best match. If we find the best matched rule, the result is a region symbol in the rule's consequence, which is temporally closest to the query time. For example, when we have a pattern $\langle r_1 r_2 r_3 r_4 \rangle$ and a new trajectory is discretized to $\langle r_1 r_2 \rangle$, then $\langle r_3 \rangle$ will be the next location if the query time is closer to the region r_2. $\langle r_3 \rangle$ will be the prediction result in other cases.

The number of rules generated from the frequent patterns is enormously large, thus we have to prune the rules before constructing a prediction model in order to reduce the number of rules and the size of the model. A minimum confidence level can be used to select rules with strong confidence.

In order to efficiently access generated patterns, we present a tree-based indexing structure called FPT (Frequent Pattern Tree). It is modified from a TRIE structure and compactly represents the selected patterns from the previous step. The path from a root to each node corresponds to a trajectory pattern and each node, except the root, has a parent, which is a region symbol appeared at preceding position in the rule's body part. As shown in Fig. 2, each node, except the root, is consist of three entries of the form of (*region*, *duration*, *support*); where *region* is the region symbol in a trajectory pattern, *duration* is the duration value of the region, and *support* means the support count of the pattern which is described by a path from the root to the corresponding node.

When the FPT is constructed, predicting the future location of a new trajectory simply is a problem of finding the closest path on the tree to the given partial trajectory. To match a new trajectory with all possible paths on FPT, we have to discretize a trajectory into the form of a sequence of regional symbols. As the summary of spatio-temporal approximation during pattern mining phase is maintained in a CF-tree, we can apply it to the discretization of the new query. The CF-tree summarizes

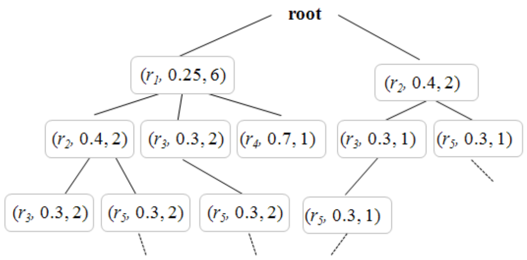


Fig. 2. A Frequent Pattern Tree

the information about clusters of data points in their nodes, which is a triple consists of ($N, \vec{LS}$, SS), where N is the number of data points in the cluster, $\vec{LS}$ is the linear sum of the N data points and SS is the square sum of the N data points. After a new trajectory is simplified into line segments with a given threshold δ, we can find the closest cluster to the segments using CF (Clustering Features) values [10]. Once a trajectory is discretized into a sequence of regions, it is not required to perform complex spatio-temporal computations to calculate the similarity between a path on the FPT tree and the given query. This reduces the processing time to evaluate all the paths for the given trajectory and improves the prediction performance.

To find the best path among all the paths on the FPT, we have to evaluate the closeness of the candidate paths on the tree and the given trajectory. Therefore we have to compute the matching scores of paths and select the temporally closest result to the query time. When we have path p and a new trajectory t, the score is calculated as follows.

$$Score\ (p, t) = S_m \times confidence \times T_{diff}\ (\ 0 \leq \text{Score} \leq 1\ )$$

- S_m : sum of the matching scores of all the nodes along the path, defined as $S_m = \sum_k^n \omega_k S_k$, where ω_k is a weight value for S_k and S_k is a match score between region symbols on the node and the new query.
- *confidence* : confidence of the rule.
- T_{diff} : the difference between the query time and the total length of the path.

For an accurate prediction, recent positions of a moving object should have a greater importance than older positions. ω_k , weight for S_k is used for this purpose. In addition, if the timestamp of the last position of a path is far away from the query time, the prediction result cannot be accurate. Thus, T_{diff} is included to incorporate temporal similarity to the query time.

4. Experimental Evaluations

In this section, we provide preliminary results of experimental evaluations of the proposed approach. C++ was used in implementation and experiments were performed on a Pentium D 3.4 GHz machine with 1GB memory. We use the C++ library of geometry functions for implementing line simplification and several distance functions. The preclustering phase of *BIRCH* is implemented based on the original paper [10] and open source code[1].

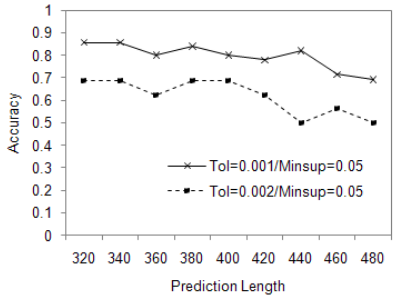


Fig. 3. Prediction accuracy of the proposed method

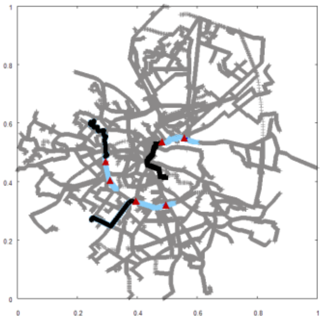
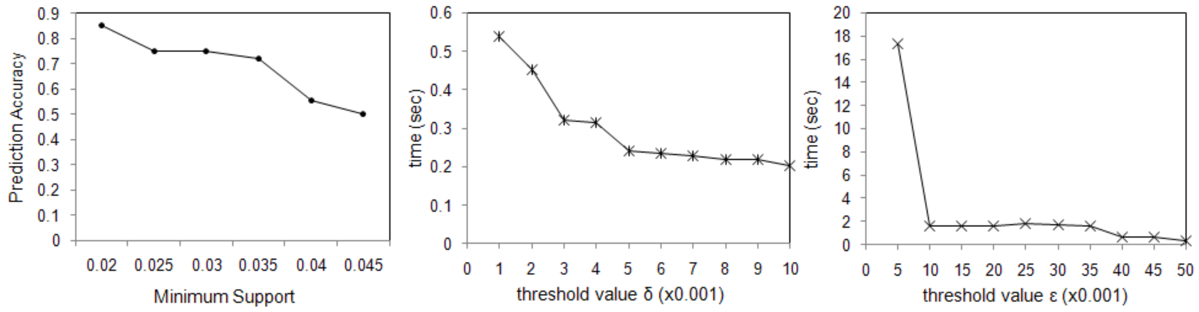


Fig. 4. Visualization of prediction results of 3 sample queries

Due to the lack of real trajectory data for privacy reasons, we generated synthetic data using Network-based Generator, by T. Brinkhoff [12]. We generated data based on the Oldenburg map. We use a dataset of 200 objects and 500 time units for the accuracy test. For the scalability test, the number of objects of dataset was varied from 100 to 500 and value of time units was set to 1000. For both cases, we set the maximum velocity of moving objects to 50 and the report probability to 1000 which means that the location is reported at every time stamp during movements. In order to compare the scalability of the mining process, we used two different existing methods GSP [13] and PrefixSpan [11]. For both cases, we discretized input data into a sequence of a location symbols using equal width discretization (EQW), which is mostly used in existing spatio-temporal mining studies.

We performed 10-cross validations on a dataset of 200 objects to evaluate the prediction accuracy. As test datasets, partial queries of 60% of the total length are used. Prediction length, which means how far the query time is from the length of the given trajectory, is the most critical factor to a degradation of the prediction accuracy. To test the accuracy of a query time in the distant future, we obse-

[1] http://pages.cs.wisc.edu/~vganti/birchcode/.



(a) Impact of *min_sup* on accuracy (b) Impact of δ on mining performance (c) Impact of ε on mining performance

Fig. 5. The impact of threshold values on pattern mining and prediction

rve the accuracy over increasing prediction length. Line simplification threshold δ was set to 0.001 or 0.002, clustering threshold ε and *min_sup* to 0.05 for both cases. Note, however, that generated data is distributed in a workspace of 1000×1000 units, all values are normalized between 0 and 1, and so threshold values should be within this boundary. The number of frequent patterns extracted from the mining phase was 192 and 718, when δ is set to 0.002 and 0.001, respectively. As shown in Fig. 3, our method shows very high accuracy with a moderate decrease as the prediction length increases, especially when δ is 0.001 (that is, the input data is properly discretized). To illustrate the results in detail, Fig 4 shows the visualization result for 3 test queries. A training data are represented by gray polylines and 3 different partial queries are represented in black, while the light blue indicates their whole path. The prediction results for 320 time units and 460 time units are depicted in red dots. As we expected, predicted positions locate correctly on the expected paths.

The next experiment examined the impact of threshold values required for the proposed method. At first, we tested the prediction accuracy with respect to the increased *min_sup* values. For all cases, δ was set to 0.002 and ε to 0.05. As we increased the *min_sup*, the number of rules generated from the mining phase decreased, thus the predictive power of the model fell slowly. Since the large number of patterns disturbed the efficient construction of an FPT model, we argue that there is a tradeoff between the predictive power of a model and the construction time. Second, the impact on the mining performance of the two threshold values δ and ε was evaluated. For testing the impact of δ, we set *min_sup* and ε to 0.05. For testing ε, we made no change in *min_sup* and the δ was set to 0.002. As we increase the threshold δ, the boundary of spatial area becomes large, thus a smaller number of simplified segments were generated for each trajectory. The length of sequence of spatio-temporal regions decreases as the value of δ grew. Therefore, as the threshold value increases, we expect that the running time of the mining process decreases, which is compatible with the result in Fig. 5b. Similarly, as threshold ε increases, more segments are grouped into one cluster and the number of different spatio-temporal regions is reduced. That is, ε is tightly associated with the different number of items (i.e. spatio-temporal regions) in the sequences. Although they do not have a linear relationship, Fig. 5 illustrates that the execution time phases down as the ε increases.

In the last experiment, we study the scalability of the proposed method by comparing total execution time with respect to the data size. We run a scale up experiment under two different *min_sup* values. We set δ to 0.06 and ε to 0.04 and divide the search space with a grid of 10×10 cells for EQW discretization for GSP and PrefixSpan algorithms. As shown in Fig. 6, the proposed method shows significant speed increase over the other methods. Since the performance of the mining process is highly associated with the length of input sequences and the number of different items appearing in the sequences, abstracted representations of our method result in a data reduction effect and the pruning of search space in the mining process. We can expect that the performance difference will be more pronounced when the cell size for discretization of the compared methods becomes smaller (like 20×20 cells).

5. Conclusions

In this paper, we presented a new approach predicting the future location of moving objects based on frequent trajectory patterns. Trajectory patterns are utilized to construct a prediction model that allows us to predict accurate locations; particularly the movements of objects that are too complex to be represented by a motion function. We introduced the problem of sequential representation of temporal properties degrading the mining efficiency and the compactness of extracted patterns. To address this problem, we proposed an efficient method for mining frequent trajectory patterns. The proposed method discovers spatio-temporal regions using line simplification and clustering, and extracts frequent patterns in a prefix-

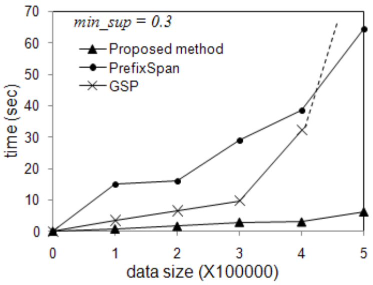


Fig. 6. Mining efficiency under different data sizes

projection approach. We also present a tree-structured prediction model using the extracted patterns. By scoring all potential paths on the frequent pattern tree, we can acquire the prediction result from the best matched path to a given trajectory. Experimental results demonstrate that our method forecasts locations accurately when the query time is far from the current time. It also discovers frequent trajectory patterns more efficiently than existing approaches.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-511-D00311)..

References

[1] M. Morzy. “Mining frequent trajectories of moving objects for location prediction,” MLDM, volume 4571 of LNCS, pp. 667–680. Springer, 2007.

[2] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou, “A hybrid prediction model for moving objects ,” In. Proc. Of ICDE, pp. 70–79. 2007.

[3] J. Y. Kang, H. Yong, “Mining trajectory patterns by incorporating temporal properties” In Proc. of 1st International Coference on Emerging Database (EDB 2009), pp. 63–68, 2009.

[4] Y. Tao, D. Papadias, and J. Sun, “The tpr*-tree: An optimized spatio-temporal access method for predictive queries,” in VLDB 2003, pp. 790–801, 2003.

[5] Y. Tao, C. Flaoutsos, D. Papadias, and B. Liu, “Prediction and indexing of moving objects with unkown motion patterns,” in SIGMOD, pp. 611–622, 2004.

[6] G. Yavas, D. Katsaros, O. Ulusoy, and Y. Manolopoulos. “A data mining approach for location prediction in mobile environments,” Data and Knowledge Engineering. vol. 54, no. 2, pp. 121–146, 2005.

[7] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. “WhereNext: a Location Predictor on Trajectory Pattern Mining,” in Proc. of KDD 2009, pp. 637–646, 2009.

[8] D. H. Cao, N. Mamoulis, D.W. Cheung, “Discovery of Periodic Patterns in Spatiotemporal Sequences,” IEEE. Transactions on Knowledge and Data Engineering, vol. 19, no. 4, pp. 453–467, 2007.

[9] H. Cao, O. Wolfson, and G. Trajcevski, “Spatio-temporal data reduction with deterministic error bounds,” The VLDB Journal, vol. 15, no. 3, pp. 221–228, 2006.

[10] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An efficient data clustering method for very large databases,” in Proc. Of ACM SIGMOD Conference on Management of Data, pp. 103–114, 1996.

[11] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu. “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth,” in Proc. of 17th International Conference on Data Engineering, pp. 215–224, 2001

[12] T. Brinkhoff, “Generating Network-Based Moving Objects,” in Proc. of SSDBM 2000, pp. 253–255, 2000

[13] R. Srikant and R. Agrawal. “Mining Sequential Patterns: Generalizations and Performance Improvements,” In Proc. of the 5th International Conference on Extending Database Technology, pp. 3–17, March 1996.

Juyoung Kang received the B.E. and M.E. degrees and finished, from Ewha Womans Univ. in 1999 and 2001, respectively. After working as a research member (from 2003) in Korean Electric Power Research Institute, a senior researcher (from 2005) in Neomtel co., she has been a Ph.D student under supervision of Prof. Yong in Ewha Womans Univ. from 2001 to 2003, and from 2006 to now. Her research interest includes data mining and information retrieval.

Hwan-Seung Yong Hwan-Seung Yong is a Professor of Computer Science and Engineering at Ewha Womans University, Republic of Korea since 1995. He received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Seoul National University. He has five years of industrial experience as Researcher at ETRI (Electronics and Telecommunications Research Institute) and served as a visiting scientist at the IBM T.J. Watson Research Center. He is member of KIISE (The Korean Institute of Information Scientists and Engineers) and ACM. His current major research interests include agile methods, data mining, multimedia database and ontology search.