Using an Adaptive Search Tree to Predict User Location

Sechang Oh*

Abstract—In this paper, we propose a method for predicting a user's location based on their past movement patterns. There is no restriction on the length of past movement patterns when using this method to predict the current location.
For this purpose, a modified search tree has been devised. The search tree is constructed in an effective manner while it additionally learns the movement patterns of a user one by one. In fact, the time complexity of the learning process for a movement pattern is linear. In this process, the search tree expands to take into consideration more details about the movement patterns when a pattern that conflicts with an existing trained pattern is found. In this manner, the search tree is trained to make an exact matching, as needed, for location prediction.
In the experiments, the results showed that this method is highly accurate in comparison with more complex and sophisticated methods. Also, the accuracy deviation of users of this method is significantly lower than for any other methods. This means that this method is highly stable for the variations of behavioral patterns as compared to any other method. Finally, 1.47 locations were considered on average for making a prediction with this method. This shows that the prediction process is very efficient

Keywords—Location Prediction, Learning System, Search Tree, Context-Awareness

1. INTRODUCTION

Context-aware systems recognize an individual's situation, and react to it accordingly. Such systems can promote and mediate one user's interactions with devices, computers, and other people. There are many kinds of context-aware applications such as proximate selection, automatic contextual reconfiguration, contextual information and commands, and context-triggered actions [1]. And the LBS (Location-Based Service) is a typical and important application area of the context-awareness. The possibility of a location-based context aware service is shown in [2].

The acquisition of location information is the most basic and vital function in the implementation of the LBS. The acquisition of location information can be divided into the two problems of location tracking and location prediction.

The problem with location tracking is tracing the current position of a moving object or having a person to maintain the LBS [3]. There are two approaches for this problem. One is to periodically obtain location information using location measuring devices such as a GPS

Manuscript received August 3, 2011; first revision April 2, 2012; accepted May 18, 2012.
Corresponding Author: Sechang Oh
* Dept. of Information and Communication, Sejong Cyber University, Seoul, Korea (scoh713@gmail.com)

receiver or an active badge [4]. And the other is to estimate the current position of the extension of the direction of movement from the previous position assuming that the location does not change drastically [5, 6]. These studies have been mainly conducted on the cellular phone system that provides communication services while maintaining the connection of the wireless network [7-9].

On the other hand, the problem with location prediction is predicting the current location of a user based on their previous movement patterns [10]. Branch prediction methods have been used to solve this problem [11]. They are very simple methods that were developed for maximizing the performance of CPU by predicting the branch instructions [12]. Recently, studies have been conducted to achieve more precise prediction results using complex models such as the Markovian approach or the Bayesian network [13, 14].

However, for making a prediction these methods only consider a limited number of recently visited places. The number is called, "the order of the predictor in branch prediction." It is usually determined experimentally. A method called prediction by partial matching (PPM) is developed in the data compression area, which flexibly matches patterns with data by adjusting the order within the range of the maximum value to maximize the prediction performance [15]. However, this approach cannot make predictions for the movement patterns when its length exceeds the maximum value of order. In addition, the complexity rapidly increases as the maximum value of order increases. In this paper, a method for prediction is proposed to accommodate all kinds of movement patterns of any length without restrictions on the order. For this, a modified search tree is suggested. Also, the method for adapting and predicting the user's behavior with the adapted search tree are suggested.

2. PROPOSED METHOD

The proposed search tree in this paper is not a binary search tree. Preferably, the order of this tree is the number of locations that users can visit. The search tree consists of nodes, and each node stands for the location that the user visited in the past. The path from the root node to a decision node is the reversed sequence of locations that the user visited. Each decision node then provides a prediction value for the current location of a user.

2.1 Location Prediction

The prediction for the current location of a user is a traversal process in this search tree with the sequence of recently visited locations. A decision node, which is shown in a gray box in Fig. 1, has a prediction value that is marked on the right side of the arrow. This means that the user will go to the location next time. All terminal nodes in this search tree are the decision nodes. In other words, the traversal process returns a prediction result when a terminal node is reached. For example, we assume that the user visited the locations 4, 11, and 1 sequentially. When we want to predict the user's current location using the search tree in Fig. 1, the traversal starts from the root node and goes through node 1, node 11, and finally node 4. Since node 4 is a decision node, the prediction is made that the user's current location is 2. Likewise, if the user visited locations 0, 11, and 1 sequentially, then the user is probably in location 11 now.

Also, some decision nodes are non-terminal in this search tree. This is because the decision nodes are arranged redundantly during the training process to deal with the conflict of other pre-

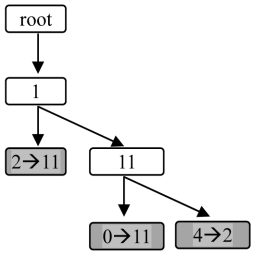


Fig. 1. Search Tree for Location Prediction

diction values emerging for the same path. So actually, the first visited decision node, rather than the terminal nodes, provides the prediction value in the traversal process.

The algorithm for predicting the location of the user is described below.

```
Process     Prediction ()

histo ← the array of locations that the user visited;
current ← the index for the last item of array histo;
i ← current; node ← root;
while (node is not a decision node)
     node ← child of the node that has the value of histo[i];
     if the node is NULL, then report it as"unpredictable";
   i ← i-1;
report node.prediction_information;
```

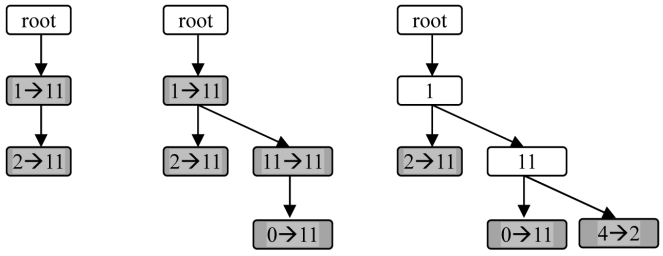
The time complexity of the prediction process is O(n) where n is the length of the array histo.

2.2 Training Process

The search tree is built by an additive training process. Initially, the search tree consists of only the root node. At this point, several nodes are added by a training sample. For example, we consider the training sample "2, 1 → 11," which means the user visited location 11 after location 2 and location 1 sequentially. Node 1 is expanded from the root node. Likewise, node 2 is expanded from node 1. The nodes are expanded precisely for this training sample, and initially these expanded nodes are decision nodes. Therefore, node 1 and node 2 in Fig. 2 (a) are decision nodes with the prediction value of 11. The search tree now has additional decision nodes under the decision node. This redundancy is to prepare for a decision node being changed to a non-decision node by additional training.

Now, let us consider how the tree is changed by the additional training process. Fig. 2 (b) is the result of additional training with the sample "0, 11, 1 → 11." The training process starts at the root node, and then goes to node 1. Node 1 remains a decision node since it has a prediction value of 11 that is identical to the final location of the training sample. Then, node 11 is expanded from node 1, since node 11 does not exist. Also, it initially becomes a decision node with the prediction value of 11. In the same way, node 0 is expanded from node 11 and it is initialized into being made a decision node with the prediction value of 11.

A decision node can be converted to a non-decision node while training. This occurs when the final location of a new training sample is not identical to the prediction value in an existing deci-



(a) train 2, 1 → 11 (b) train 0, 11, 1 → 11 (c) train 4, 11, 1 → 2

Fig. 2. Construction Process of the Search Tree

sion node during path traversal. Because of the conflict between information, the decision node cannot determine the prediction value anymore. As a result, it needs to convert into becoming a non-decision node. For example, Fig. 2 (c) is the result of additional training with the sample "4, 11, 1 → 2." Since the first location to consider is 1, the predictor goes to node 1 from the root node. At this point, node 1 cannot have the prediction value because the prediction value of 11 for node 1 and the final location of the training sample are in conflict. Therefore, node 1 is converted to a non-decision node and it goes to node 11. Also, node 11 is converted to a non-decision node since the prediction value of it is different from the final location of the training sample. Finally, node 4 is expanded from node 11 and it is initialized into being made a decision node with the prediction value of 2.

The following algorithm is for training the search tree described above:

```
Process    Training ()

histo ← the array of locations that the user visited;
current ← the index for the last item of the array histo;
i ← current-1; node ← root; dest ← histo[current];
while (i >= 0)
    next_node ← the child of node that has the value of histo[i];
    if next_node is NULL
        add a child to the value histo[i] and to the prediction information
    dest;
        node ← just added node;
    i ← i-1; node ← next_node;
```

Many training samples can be made from the record of locations that a user visited in a day. The number of training samples is n-1 if a user visited n locations in a day. This is because a training sample consists of at least one visited location and a final location. In the training samples, the length of the list of visited locations varies from 1 to n-1. The time complexity of training processes for each training sample is linear. Equation (1) shows the number of comparisons during the search tree traversal for training all of these samples.

$$\sum_{i=2}^{n}(i-1)=\frac{n^2-n}{2} \tag{1}$$

Therefore, the time complexity of the training process for all of the samples occurring in a day is $O(n^2)$. Also the resulting search tree may have a maximum of L^n - 1 nodes, where L is the number of locations that users can visit.

3. EXPERIMENTAL RESULTS

3.1 Experiment Environment

In this paper, the data provided by AILTB (Augsburg University Indoor Location Tracking Benchmarks) was used for experiments. This benchmark data was collected for the 4 researchers--A, B, C, and D--who all participated in the experiment. Every change to the location of researchers was recorded on the PDA. In this way, the collected information, including user IDs and visited locations, was used for experiments. Fig. 3 shows the floor plan used for the environment that was used for the experiment.

In the experiment, 16 locations were taken into account, as shown in Fig. 3. This included "away," which means the user was not on this floor but it excluded the "stairway" and "elevator." The data was collected for 4 people during the summer and the fall, as shown in Table 1.

One location predictor was constructed for each user. The data collected in the summer was used for training the predictors. The data collected in the fall was used to evaluate the performance of the trained predictors.

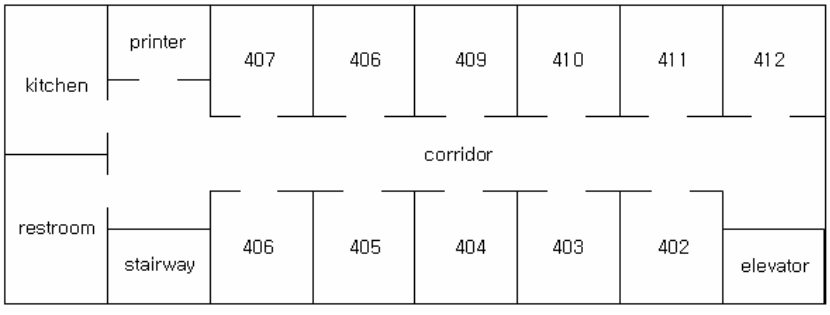


Fig. 3. Floor Plan of Sectors

Table 1. Experimental Data

Person	Period	Time Period	Number of Entries
A	Summer	1 week	101
	Fall	4 weeks	432
B	Summer	2 weeks	448
	Fall	5 weeks	982
C	Summer	2 weeks	351
	Fall	4 weeks	911
D	Summer	2 weeks	158
	Fall	7 weeks	848

3.2 Analysis of the Results

Fig. 4 shows the accuracy of the prediction for each user. The accuracy is evaluated as 80.67% on average.

Table 2 shows the comparison of the proposed method with existing predictors that were used in other studies [16]. According to this table, the predictor representing the highest accuracy on average is the Bayesian Network and the proposed method showed as the second highest accuracy. But the deviation of accuracy that depends on the user is high in most predictors, including in the Bayesian Network. This means that individual tuning is necessary. However, the results show the universality of the proposed method for individual characteristics since the deviation of accuracy of the method is the lowest with 2.55 in standard deviation. Another advantage of the proposed method is that its training process is simpler than the Bayesian Network's.

Table 3 shows the statistical data that indicates the time complexity of the prediction process using the trained search tree. The "Avg. # of Comparisons" in this table indicates the average value of how many times the comparison operator was used in the prediction process for each user. This means that the prediction results have been obtained after traversing the search tree to a depth of this value on average. According to the results, a comparison was performed 1.47 times on average to obtain one prediction result. The individual variation is also insignificant with 0.18.

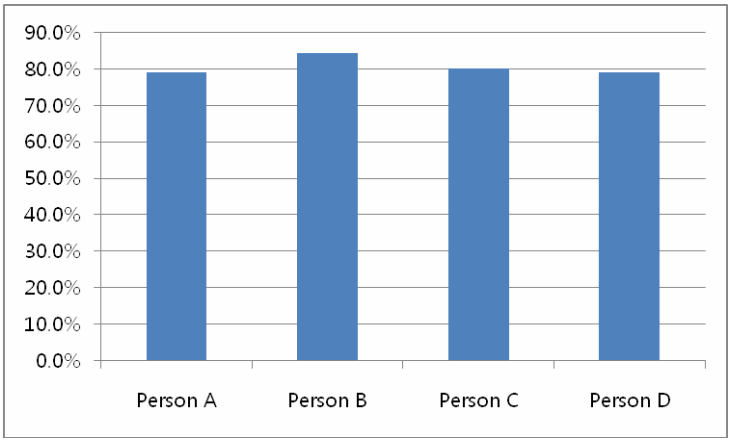


Fig. 4. Prediction Rate for Each Person

Table 2. Comparison of the Accuracy of Predictors

Predictor	Person				Avg.	Standard Deviation
	A	B	C	D		
Elman Net	91.07%	78.88%	69.92%	78.83%	79.68%	8.69
MLP	87.39%	75.66%	68.68%	74.06%	76.45%	7.88
Bayesian Network	85.58%	86.54%	86.77%	69.78%	82.17%	8.27
State Predictor	88.39%	80.35%	75.17%	76.42%	80.08%	5.96
Markov Predictor	90.18%	78.97%	75.17%	78.05%	80.59%	6.59
Proposed Method	79.05%	84.41%	80.19%	79.03%	80.67%	2.55

Table 3. Time Complexity of the Prediction Process

Measure	Person				Avg.
	A	B	C	D	
Avg. # of Comparisons	1.33	1.31	1.61	1.65	1.47

4. CONCLUSION

In this paper, we proposed a method for predicting the user's location based on the past movement patterns. In related studies, predictors consider a limited number of recently visited places to make a prediction. On the other hand, there is no restriction on the length of past movement patterns for using the proposed method to consider the current location prediction.

For this purpose, a modified search tree was devised. This search tree includes some decision nodes and a prediction is made when a decision node is reached during traversal. The search tree is constructed while it additionally learns the movement patterns of a user one by one. Of course, the learning process for a movement pattern is very efficient since the time complexity is linear. But the number of locations a user visited in a day is n, and $(n^2-n)/2$ training samples are generated from them. Therefore, the time complexity of the whole learning process is $O(n^2)$. In this process, the search tree expands to consider more details about the movement patterns when a pattern that conflicts with an existing trained pattern is found. In this manner, the search tree is trained to consider just the necessary places to predict the location.

In the experiments, the prediction rate of the proposed method is 80.67% accurate on average. It is highly accurate in comparison with more complex and sophisticated methods. Also, the deviation of the accuracy depending on the user is 2.55 in standard deviation. This is significantly lower than the other methods that range from 6 to 9 in standard deviation. This shows that the proposed method is much more adaptable than any other methods for a variety of individual characteristics. Finally, the proposed method considers 1.47 places on average to make a prediction. This shows that the prediction process is very efficient.

Basically, the method proposed in this paper uses additive training. It enables the predictor to distinguish the differences in the new pattern without forgetting the patterns already learnt. Obviously, this is a big advantage. However, it can also be a disadvantage when the movement patterns are radically changed due to changes in the job position. In this case, the predictor does not forget the unnecessary patterns that were learned in the past and this may conflict with new movement patterns. This problem can be solved by reconstructing the search tree for the user. For this purpose, the methods of detecting radical changes in the user's movement patterns needs to be further studied.

REFERENCES

[1] Schilti, B., Adams, N., Want, R., "Context-Aware Computing Applications," Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications, 1994, pp.85-90.

[2] Y. Chon, H. Cha, "LifeMap: A Smartphone-based Context Provider for Location-based Service," IEEE Pervasive Computing, Vol.10, No.2, pp.58-67, April-June, 2011.

[3] Yan Shen, "Location Prediction for Tracking Moving Objects," IEEE Global Congress on Intelligent Systems, Vol.1, 2009, pp.362-366.

[4] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge location system," ACM Transactions on Information Systems, January, 1992, pp.91-102.
[5] Lei Mu, Geng-Sheng Kuo, Ningning Tao, "A Novel Location Algorithm Based on Dynamic Compensation Using Linear Location Prediction in NLOS Situations," IEEE 63rd Vehicular Technology Conference, Vol.2, 2006, pp.594-598.
[6] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, "An Adaptive Location Prediction Model based on Fuzzy Control," Journal of Computer Communications, Elsevier, 34(7), May, 2011, pp.816-834.
[7] Yao Yuan, Yucheng Zhang, Yi Huang, Xin Jin, XiaoFei Zheng, Jinglin Shi, "An Adaptive Wireless Paging Scheme Using Bayesian Network Location Prediction Model," IEEE 70th Vehicular Technology Conference, 2009, pp.1-5.
[8] Joon-Min Gil, Chan Yeol Park, Youn-Hee Han, Chong-Sun Hwang, Young-Sik Jeong, "Simulation of a Mobility Prediction Scheme Based on Neuro-Fuzzy Theory in Mobile Computing," Simulation 75:1, July, 2000, pp.6-17.
[9] N. Meghanathan, "A Location Prediction Based Routing Protocol and its Extensions for Multicast and Multi-path Routing in Mobile Ad hoc Networks," Elsevier Ad hoc Networks, Vol.9, No.7, September, 2011, pp.1104-1126.
[10] Jan Petzold, "Augsburg Indoor Location Tracking Benchmarks," Technical Report 2004-09, Institute of Computer Science, University of Augsburg, Germany, April 2004, www.informatik.uni-augsburg.de/skripts/techreports.
[11] Jan Petzold, et al., "Context Prediction Based on Branch Prediction Methods," July, 2003, Technical Report 2003-14, Institute of Computer Science, University of Augsburg, Germany, www.informatik.uni-augsburg.de/skripts/techreports.
[12] J.B. Chen, M.D. Smith, C. Young, N. Gloy, "An Analysis of Dynamic Branch Prediction Schemes on System Workloads," 23rd Annual International Symposium on Computer Architecture, 1996, pp.12-20.
[13] P. Fulop, S. Szabo, T. Szalka, "Location Prediction Methods with Markovian Approach and Extended Random Walk Model," IEEE CISIM (Computer Information Systems and Industrial Management Applications), 2007, pp.53-58.
[14] Yucheng Zhang, et. al., "Location Prediction Model Based on Bayesian Network Theory," IEEE GLOBECOM (Global Telecommunications Conference), 2009, pp.1-6.
[15] I-Cheng K. Chen, John T. Coffey, and Trevor N. Mudge, "Analysis of Branch Prediction via Data Compression," In ASPLOS VII, October, 1996, pp.128-137.
[16] Jan Petzold, Faruk Bagci, Wolfgang Trumler, Theo Ungerer, "Next Location Prediction Within a Smart Office Building," 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE'05) at the 3rd International Conference on Pervasive Computing, May, 2005.

Sechang Oh

He received the M.S. and Ph.D. degrees in computer science from the KAIST in 1990 and 1997, respectively. He worked at LG Corporation Institute of Technology for four years and worked at Ajou University for three years. Now he is a professor in Information & Communication Dept. at Sejong Cyber University. His research interests include pattern recognition, data mining, ubiquitous computing and complex system.