Neurocomputing 176 (2016) 98–106

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Exploiting machine learning techniques for location recognition and prediction with smartphone logs ☆

Sung-Bae Cho

Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, South Korea

ARTICLE INFO

Article history:
Received 10 March 2014
Received in revised form
29 January 2015
Accepted 6 February 2015
Available online 8 May 2015

Keywords:
Location recognition
Location prediction
Location-based services
Hidden Markov model
k-Nearest neighbor
Decision trees

ABSTRACT

Due to the advancement of mobile computing technology and the various sensors built in the smartphones, context-aware services are proliferating to everyday life. Location-based service (LBS), which provides the appropriate service to smartphone users according to their contexts, is becoming more popular, and the location is one of the most important contexts in LBS. Extracting and recognizing meaningful location and predicting next location are crucial for successful LBS. Many researchers have attempted to recognize and predict locations by various methods, but only few consider the development of real working system considering key tasks of LBS on the mobile platform. In this paper, we propose a location recognition and prediction system in smartphone environment, which consists of recognizing location and predicting destination for users. It recognizes user location by combining k-nearest neighbor and decision trees, and predicts user destination using hidden Markov models. To show the usefulness of the proposed system, we have conducted thorough experiments on real everyday life datasets collected from 10 persons for six months, and confirmed that the proposed system yielded above 90% of average location prediction accuracy.

1. Introduction

As the mobile devices are popular in everyday life, smartphone has become a powerful platform for mobile context-aware services, which have attracted great attention and propelled active investigation about inferring user's mobile contexts for useful services [1–4]. It provides with a lot of interesting information about users, especially as various sensors have been equipped in recent mobile devices. One of the most important contexts might be the location. The proper services and information could be delivered according to user's current location or future location. In spite of a lot of research with respect to location prediction, however, there are few real working systems which recognize and predict the location on the mobile device.

In this paper, we address the key issues of location-based services by exploiting machine learning techniques to develop a hybrid AI system for location prediction with smartphone logs. The main components of the system are as follows.

1.1. *Extracting meaningful locations*

To effectively extract the meaningful locations, we exploit the G-means clustering method, which automatically determines the number of clusters by performing statistical test iteratively. The stay points are identified automatically for G-means algorithm to reduce the burden in the mobile computing.

1.2. *Recognizing and predicting locations*

To effectively recognize the locations in outdoor as well as in indoor, we utilize several classification methods such as k-nearest neighbor (kNN) and multiple decision trees (DT). The location prediction problem is formulated as a path classification and model selection problem. The proposed method constructs a hidden Markov model (HMM) for each path by utilizing intermediate locations, which come from the location recognition phase, and predicts the next location by selecting the model that produces the largest matching score. The model uses the smartphone logs such as transportation mode, day of week and time to predict the context-sensitive locations reliably.

1.3. *Managing models and log data*

To make the system user-friendly, we develop an intuitive user interface for location recognition and prediction. The system has

☆This is the extended paper presented at the HAIS: Y.-J. Kim and S.-B. Cho, A HMM-based location prediction framework with location recognizer combining k-nearest neighbor and multiple decision trees, in: Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, pp. 618–628, 2013.

E-mail address: sbcho@cs.yonsei.ac.kr

http://dx.doi.org/10.1016/j.neucom.2015.02.079
0925-2312/

the functions of location management and path management to manage user's locations and paths appropriately.

The problem of location prediction can be formulated as follows. Given an observed user trajectory $T=\ell_0\rightarrow\ell_1\rightarrow\ell_2\rightarrow\cdots\rightarrow\ell_t$ and the location set $L=\{L_1, L_2, L_3, \ldots, L_n\}$, where ℓ_i means the ith location that user has visited, we want to predict the next location ℓ_{t+1}. In Fig. 1, given the information of trajectory, $L_2\rightarrow L_4\rightarrow L_3\rightarrow\ldots\rightarrow L_5$, we have to predict the next significant location of L_4.

In this paper, the location prediction problem is transformed into the path classification problem. Given an observed user trajectory, $T=\ell_0\rightarrow\ell_1\rightarrow\ell_2\rightarrow\cdots\rightarrow\ell_t$, the set of locations $L=\{L_1, L_2, L_3, \ldots, L_n\}$, the set of user's movement paths $P=\{p_1, p_2, p_3, \ldots, p_m\}$ where $p_i=(\ell_s^i, \ell_d^i)$ (s and d denote the start and destination indicators, respectively), and the set of intermediate locations $I=\{I_1, I_2, I_3, \ldots, I_n\}$ between the start and destination locations of each path p_i, we have to classify an observed user trajectory T into one of the user's movement paths. By solving this problem, we can get a classified path and predict user's next location by returning destination location of the path.

Many researchers have attempted to work out the problem of location recognition and prediction by using several approaches. Table 1 shows some related works on location prediction.

In addition, Ashbrook et al. extracted user's significant locations from GPS data and presented a location predictor based on the Markov models [11]. Krumm et al. devised a method called predestination that predicts driver's destination as trip progresses [12]. Alvarez-Garcia et al. presented a new approach to predict destinations given the only data of a partial trip by using HMMs and local street-map [13]. Petzold et al. presented a dynamic Bayesian network to predict the next indoor location and compared with the state predictor and multi-layer perceptron predictor [14].

Yavas et al. presented a data mining algorithm for the prediction of user movements in a mobile computing system [15]. Monreale et al. proposed the trajectory pattern tree aimed at predicting the next location of a moving object with a certain level of accuracy [16]. Morzy mined the database of moving object locations to discover frequent trajectories and movement rules, and matched the trajectory of a moving object with the database of movement rules to build a probabilistic model [17]. Petzold et al. compared various methods for the next location prediction [18]. They conducted the comparative experiments using dynamic Bayesian network, multi-layer perceptron, Elman net, Markov predictor and state predictor. Scellato et al. presented a novel framework for predicting user's next locations based on nonlinear time series analysis [19].

Most of the works related with location prediction have the limitation that they mainly focus on the performance of methods and only few consider the development of real working system on mobile computing environment. In this paper, we present a hybrid system of location recognition and prediction with smartphone and conduct a realistic experiment with large data collected by 10 users for six months. The system can be used personally, so that the number of users is less important than the duration of collection. We have collected the everyday life data for six months per each user, which is not trivial and expensive to obtain.

The rest of this paper is organized as follows. Section 2 presents the details of the proposed system. Section 3 illustrates the personalized location prediction system implemented and the results of experiments. The summary and future work are described in Section 4.

2. The proposed system

The proposed system for predicting user's next location is illustrated as shown in Fig. 2. k-Nearest neighbor (kNN) and decision tree (DT) are used for recognizing the current location from the information of T_n and S_n; the time and transportation mode when user visits the nth location, respectively. The information can be used to model user's paths discriminatively. Hidden Markov models (HMMs) are used for predicting the destination by path classification.

2.1. Location extraction

To model user's paths effectively, we extract the intermediate locations between start and destination locations by clustering GPS data. Most of the previous works based on clustering method used k-means clustering algorithm [11,20,21]. However, this is not

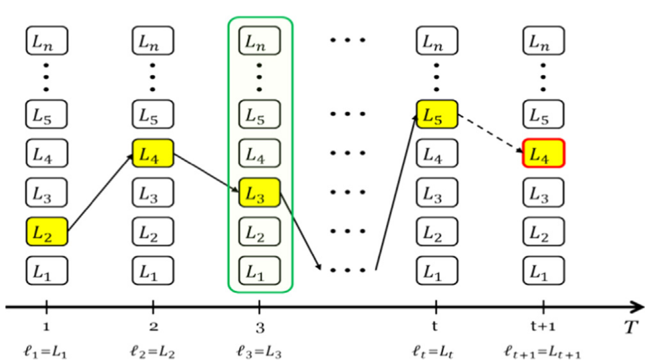

Fig. 1. An example of location prediction problem.

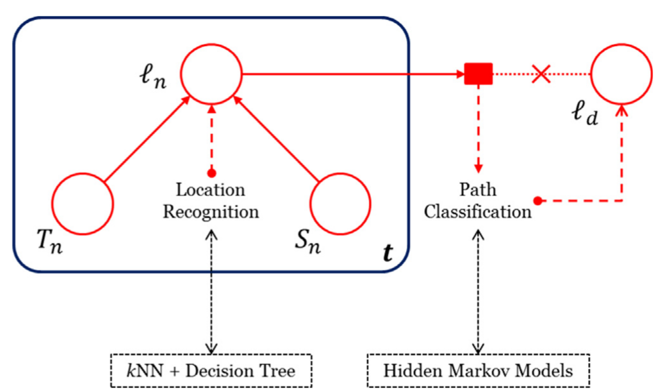

Fig. 2. Overview of the proposed system.

Table 1
Related works of location prediction.

Year	Author	Input	Method	Dataset
2012	Mathew et al. [5]	Location sequence, time, day of week	HMM	GeoLife dataset
2012	Gambs et al. [6]	POI sequence	Mobility Markov Chains	Phonetic dataset, GeoLife dataset, Synthetic dataset
2012	Heo et al. [7]	GPS, activity, day of week, time, velocity	Hierachical DBN	Data for 2 months
2008	Burbey et al. [8]	Location, time	PPM	UCSD dataset
2006	Simmons et al. [9]	GPS, Map DB	HMM	Data for 1 month
2004	Liao et al. [10]	GPS	Hierarchical DBN	Data for 2 months

suitable for real-world system because it requires the pre-knowledge about the number of *k*. Instead of using k-means clustering algorithm, we use Gaussian-means method, simply called as G-means clustering method, which is based on statistical test for the hypothesis that a subset of data follows a Gaussian distribution. It automatically selects the number of clusters *k* by iteratively performing k-means clustering method until the test accepts the hypothesis that the data assigned to each k-means center are Gaussian [22]. Fig. 3 shows the process of G-means clustering method.

G-means clustering method is useful for automatically finding the number of clusters that satisfies the Gaussian distribution. However, it would take too much time to apply G-means method to the user trajectory data. To improve the performance of extracting meaning locations and reduce the computing time, we perform a preprocessing called stay point extraction.

According to the statistics, about 85% of user's GPS points are stay points. Because the running time of G-means method is proportional to the number of data points, extracting stay points has a great significance. It helps reduce the running time of G-means method so as to reduce the burden in mobile devices. In this paper, we use naïve Bayes classifier to determine whether the user is moving or staying. Naïve Bayes classifier is a simple probabilistic model based on Bayes theorem and its low complexity of computation is merit to run on mobile phone environment. As input of naïve Bayes classifier, we use accelerometer, magnetic and orientation sensors. Fig. 4 is an example of extracting locations by G-means clustering method.

2.2. Location recognition

Location recognition task is crucial for predicting user's location because the location sequence of a path is used to learn the path classification model. The box in Fig. 1 represents the location recognition part. For the outdoor environment, several researchers utilized GPS signal as input to classify location [23–25]. However, it has limitations for location recognition to use only GPS. The GPS signal includes large uncertainty: In urban canyon or with bad weather condition, the quality of GPS signal is quite low. Moreover, the GPS signal has large error in indoor environment. For the indoor environment, many researchers used Wi-Fi information to classify location [26,27]. Compared to the other information such as RFID, Bluetooth and Razor, Wi-Fi is a better choice for recognizing indoor location. As more and more people use Wi-Fi to access the Internet, there is no need to construct specific environment for identifying locations.

The proposed system takes place-related contexts (GPS, Wi-Fi) and other contexts (visit time, weekday/weekend) as input and uses the DT-based method integrated with kNN to recognize location. A DT uses a branching method to illustrate every possible outcome of a decision, and it can be also operated efficiently in

Table 2
Algorithm of location recognition.

Algorithm: Location Recognition

```
Input: Position (Current GPS coordinate, Wi-Fi AP information)
Output: Identified location name or "Unknown"
N ← kNN(Position)/* candidate locations*/
for k = 1, …, N do
  result ← DecisionTree(k)
  if (result = "Yes") then
    return GetLocationName(k)
  end if
end for
return "Unknown"
```

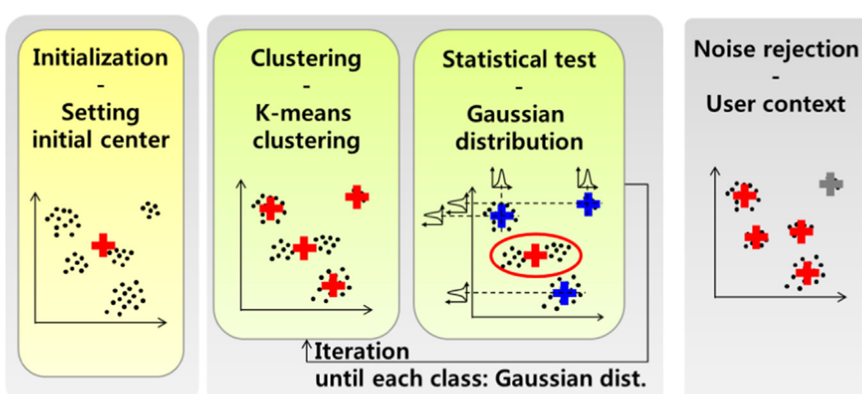

Fig. 3. Process of G-means clustering method.

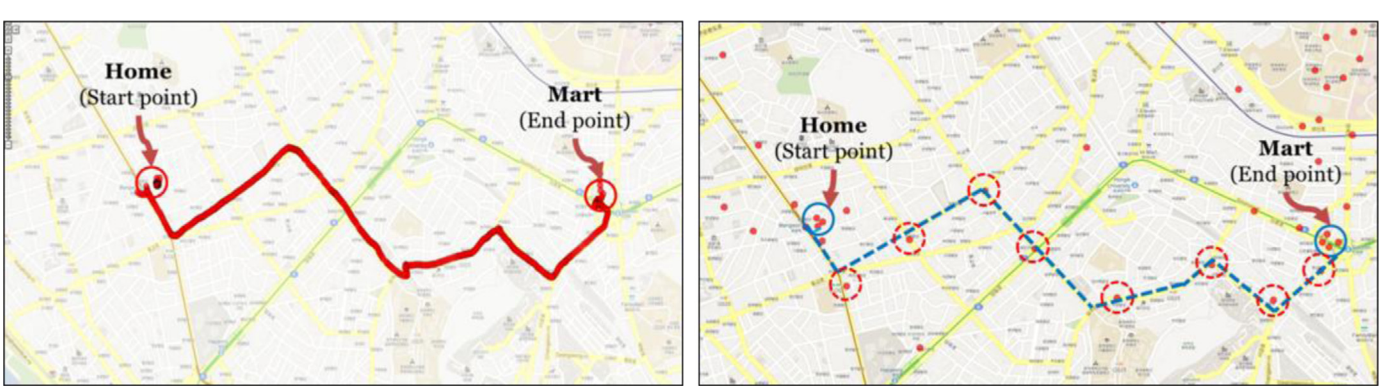

Fig. 4. The left figure shows the original user trajectory and the right one shows the result of extracting intermediate locations.

mobile devices because it can be trained in a short time [28–30]. In the outdoor environment, where GPS is available, kNN method is used to classify user's location. To reduce the computation, the proposed method first filters data with date and time information and then selects k GPS points nearest to the current location. With the k candidates, kNN classifies the current location by majority vote. In the indoor environment, where GPS is not available, multiple decision trees are applied to identify the current location. Table 2 shows the location recognition method as pseudo code.

The location recognition method firstly performs kNN at user's current position, and then filters the locations of which current position is out of their boundary. Secondly, decision trees of the candidate locations are invoked to identify current user's location. For the input of the method, we use two types of information, GPS and Received Signal Strength Indicator (RSSI) of Wi-Fi access point.

Each decision tree model is constructed with two location information. The system will find the nearest location to construct decision tree model together with the current location information. To find the nearest location from the current one, it needs to satisfy the following condition. The physical distance of the two locations is the shortest or two locations have most of the Wi-Fi APs in common.

2.3. Location prediction

We use HMM as a classifier for classifying user's path. A HMM is a statistical Markov model, which works with a Markov process with unobserved (hidden) states. HMMs are well known for their application in recognizing temporal patterns such as speech, hand-writing, gesture, and so on [31]. The use of HMMs for path classification enables us to account for location characteristics as hidden states, and also to account for the effects of each individual's previous actions. Fig. 5 shows the overview of location prediction. It consists of preprocessing part, location extraction part, location recognition or mapping part, and location prediction part.

To model the user's path, we need to generate a path by exploiting user trajectory data. According to the trait of log collector, user trajectory data are collected by each path, and by mapping GPS trajectory into the symbolic locations (extracted by G-means clustering method) we can easily find the proper path.

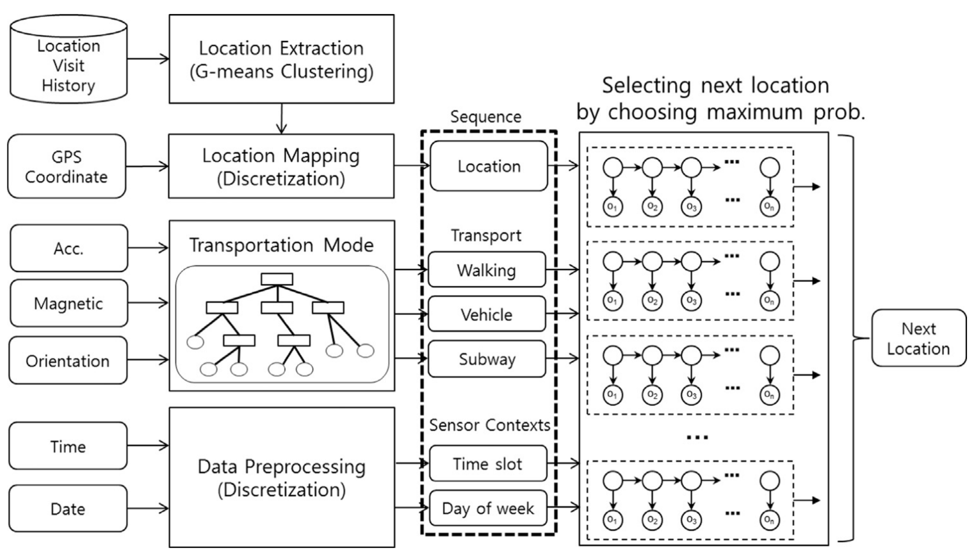

Fig. 5. Overview of location prediction.

Table 3
Algorithm of path generation.

Algorithm Path Generation

Input: Raw trajectory data $T_{GPS} = \{t^1_{1:N_1},\ t^2_{1:N_2},\ \dots, t^l_{1:N_l}\ \}$, a set of locations L, GPS error radius e_r
Output: Symbolic path data P_{Symbol}
$P_{Symbol} \leftarrow \varnothing$
for $i=1$ to l **do**
$p^i \leftarrow \varnothing$
for $j=1$ to N_i **do**
$p^i_j \leftarrow$ NULL
for $k=1$ to length (t^i_j) **do**
temp $\leftarrow$ LocationMapping $(t^i_j(\mathrm{k}), L, e_r)$
if (temp $\neq$ GetLastSymbol(p^i_j)) **then**
Attach Symbol $(p^i_j$, temp)
end if
end for
$p^i \leftarrow p^i \cup \{p^i_j\}$
end for
$P_{Symbol} \leftarrow P_{Symbol} \cup \{p^i\}$
end for
return P_{Symbol}

Table 3 shows the algorithm of path generation, where T_{GPS}, L, and e_r mean the set of trajectory, the set of locations and the set of GPS error radius, respectively.

The observation symbols of HMMs are constructed by using location information from location recognition, the time when user visits the location, and transportation mode classified by exploiting accelerometer, magnetic and orientation sensors [32]. Table 4 shows the observation symbols of HMM used in this paper.

Eq. (1) is the probability of the observation sequence O, given the HMM parameter λ_i of user path p_i.

$$P(O|\lambda_i) = \sum_{Q} P(O|Q, \lambda_i)P(Q|\lambda_i) \tag{1}$$

With the paths from path generation phase, we use Baum–Welch algorithm to learn and update HMM. Whenever new path data are available, we use the accumulated path information to update the HMM.

Given the observation sequence, it is possible to predict user's next location based on the path HMM. Given the user's GPS trajectory from one point to another, GPS signal can be transformed into symbolic location name. Transformed symbolic location name can compose the observation value of HMM models together with the other contexts such as time, day of week, and transportation mode. Afterwards, we find out all the paths that have the same start point as the observation sequence. Applying the observation sequence to each of the path HMMs leads to the matching probability. Finally, we select the path with the maximum probability value as follows.

$$\hat{P} := argmax_i P(T|\lambda_i) \tag{2}$$

Table 5 shows the pseudo code for location prediction algorithm.

Fig. 6 shows an example of location prediction process. By location mapping phase, user GPS trajectory data are transformed into input sequence. The proposed system uses the input sequence as observation to perform model selection. In this example, the proposed system selects the path HMMs starting from location L_5. Among the results, the system selects the path with the maximum probability. According to the result, the path Model $HMM_{5\to7}$ has the maximum probability, so that the system takes L_7 as the user's next location.

Table 4
Observation symbols of the destination prediction model.

Observation symbol	Description
LocationID (LID)	The identifier of location, ex) 1, 2, 3, …, n
TransportationMode (TM)	Walking, Vehicle, Subway
DayOfWeek (DoW)	SUN, MON, TUE, WED, THU, FRI, SAT
TimeOfDay	t0_t4, t4_t8, t8_t12, t12_t16, t16_t20, t20_t24

Table 5
Algorithm of location prediction.

Algorithm: Location Prediction

Input: Observation sequence $O_{seq} : o_1 \to o_2 \to o_3 \to \cdots \to o_m$
Output: Predicted destination $\hat{D}$
Path ← GetPaths(o_1)/* user paths starting from o_1*/
result: = 0
index: = 0/* for path index*/
for k = 1, …, |Path| **do**
  temp ← $HMM_{Path(k)}(O_{seq})$
  /* finding the index having maximum value */
  if (result < temp) **then**
    result: = temp
    index: = k
  end if
end for
$\hat{D}$ ← GetDestinationOfPath(index)
return Predicted location $\hat{D}$

3. System implementation

In this section, we present the personalized location prediction system implemented. The system is composed of the sensor log

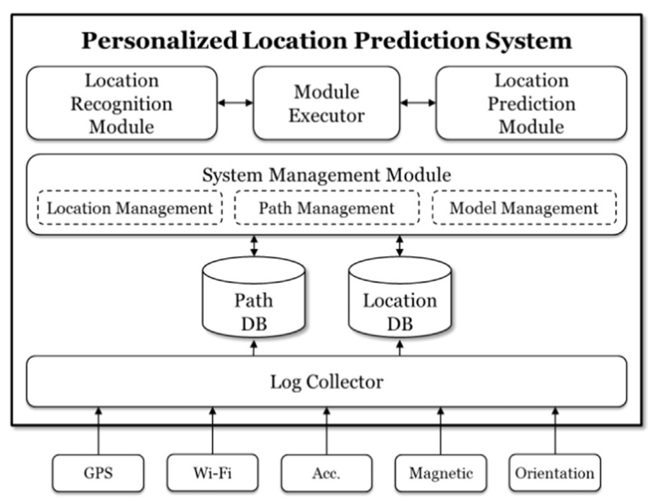

Fig. 7. Architecture of the personalized location-aware system.

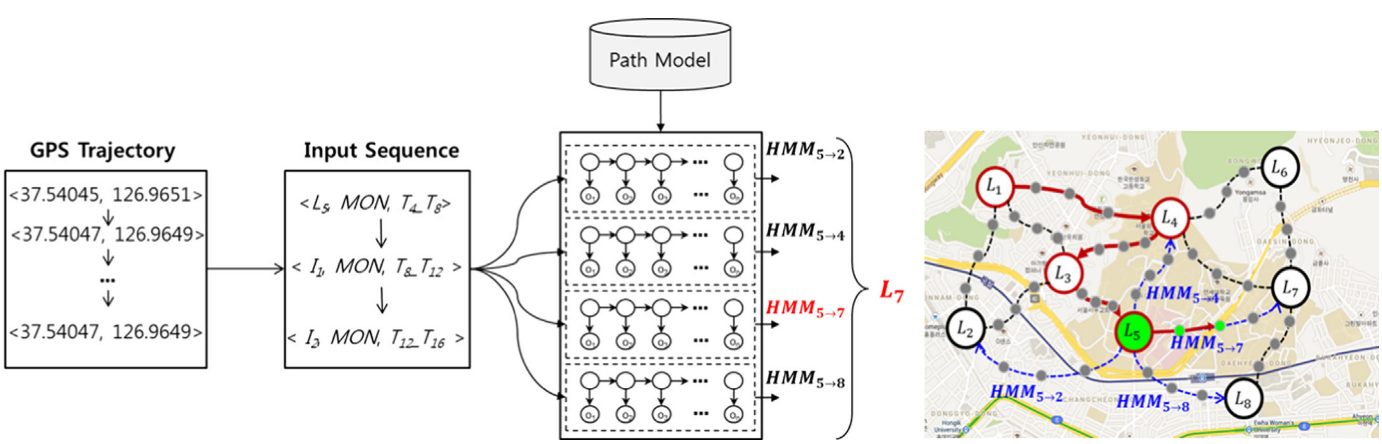

Fig. 6. An example of location prediction process.

collector, location recognition module, destination prediction module, and system management module, as shown in Fig. 7.

We implemented mobile sensor log collector, and integrated it into our system. The types of sensors used are shown in Table 6, and sensor collection period is 0.5 s. Fig. 8(a) shows the implemented log collector.

Fig. 9 shows the flow of location prediction. Our system for predicting user's next location is implemented in the location recognition and prediction modules.

Table 6
Various types of sensors used for data collection.

No.	Sensor	Description
1	GPS	Latitude, Longitude
2	Wi-Fi	MAC address, RSSI
3	Acceleration	3-Axis double type data
4	Magnetic field	3-Axis double type data
5	Orientation	Orientation, pitch, roll
6	Time stamp	Date, time
7	Transportation mode	Staying, walking, vehicle, subway

System management module consists of three parts including location management, path management, and model management. In location management, the locations labeled by user are managed in relational database. The user can add new locations, modify the information of registered locations, and delete existing locations. Fig. 8(c) shows a screenshot of location setting. The paths of user are also managed in relational database. Model management is responsible for learning location recognition and prediction models. Fig. 8(d) is an example of learning location prediction model in smartphone.

4. Experiments

To show the performance of the proposed system, we conduct experiments for evaluating the prediction accuracy. At first, we give an account of the dataset and performance analysis of the experiments. For the experiments, ten undergraduate students (6 males, 4 females) of Yonsei University collected smartphone sensor data for six months from July to December in 2013. We used the Samsung Galaxy S4 for the experiments. The size of collected data is shown in Table 7.

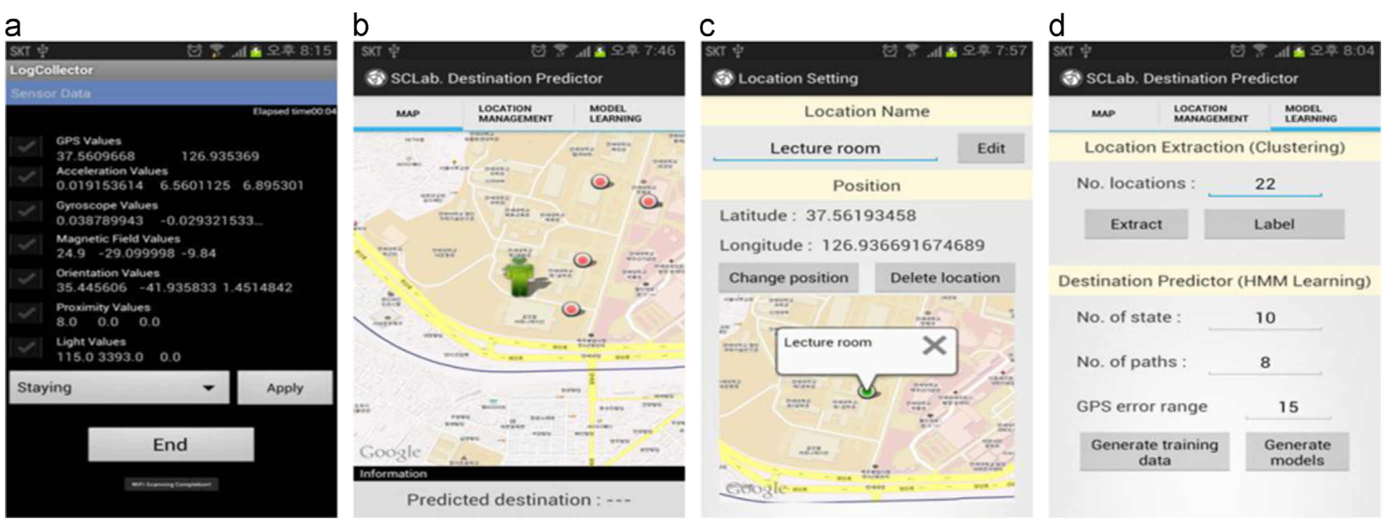

Fig. 8. Screenshots of the implemented system: (a) mobile sensor log collector; (b) initial map screen; (c) specific location setting; and (d) model learning for destination prediction.

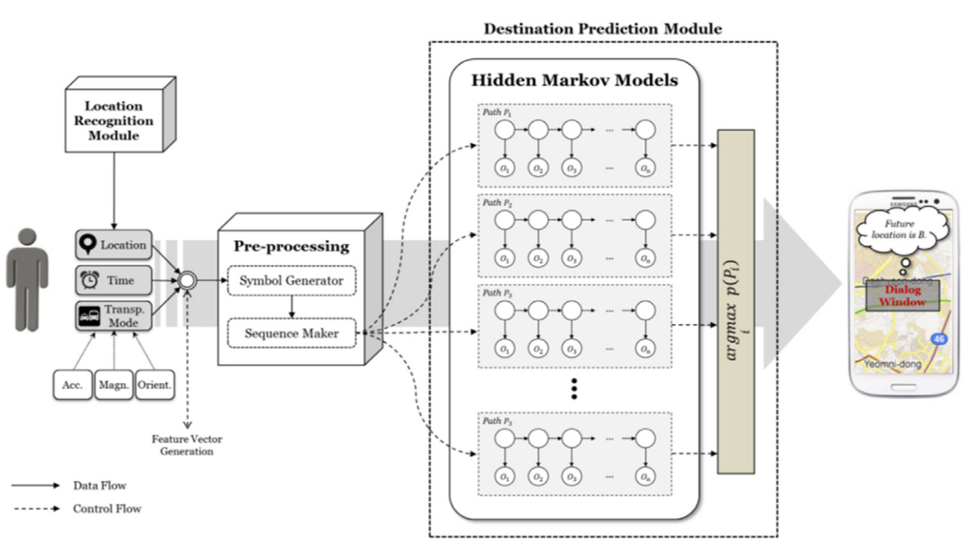

Fig. 9. Flow of location prediction.

Table 7
The information of collected data.

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
#Location	16	20	32	50	42	34	28	24	36	14
#Path	193	268	149	288	309	233	236	294	237	189
#Intermediate location	157	195	165	218	233	121	136	227	115	146
Size	2.44 GB	2.62 GB	1.46 GB	3.41 GB	3.66 GB	1.34 GB	989 MB	3.21 GB	2.37 GB	2.08 GB

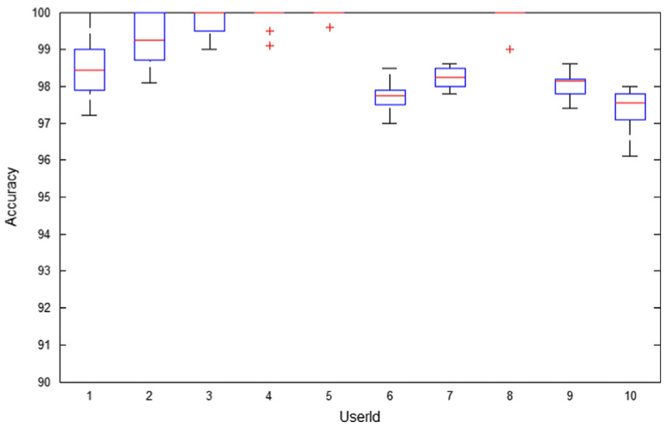

Fig. 10. Outdoor location recognition accuracy for each user.

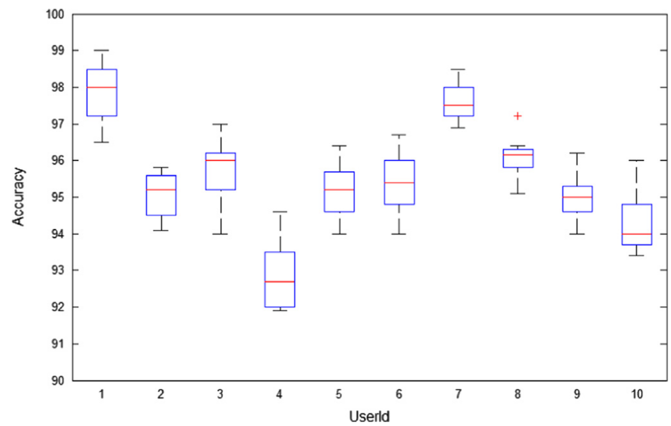

Fig. 11. Indoor location recognition accuracy for each user.

To measure the performance of the proposed system, the experiments of location extraction, location recognition and location prediction were conducted. In order to evaluate the place extraction performance, we clustered the dataset in Table 7 using G-means clustering method. The number of extracted intermediate locations is also shown in Table 7.

For the location recognition experiment, 10-fold cross validation is used to measure the performance. The average accuracy of outdoor location recognition is 98% and the average indoor location recognition is 95.5%. Figs. 10 and 11 show the accuracies of outdoor location recognition and indoor location recognition.

For the outdoor locations, because of far distance between each other, the accuracy is pretty high. As for the indoor locations, the average accuracy is lower than that for outdoor locations. The reason is that there are some indoor locations located near each other, which causes the difficulty to the recognition. Moreover, indoor location recognition mainly uses Wi-Fi RSSI to perform the recognition. Instability of Wi-Fi RSSI, however, is the other factor that affects to the accuracy.

Using the implemented location prediction models, we performed the leave-one-out cross-validation (LOOCV) for each dataset of the users in Table 7. Fig. 12 shows the experimental results.

As a result of the experiment, the proposed system archives 90.89% of average accuracy. As can be seen from the figure, the performances vary depending on the paths. The accuracy of some paths is lower than 80% whereas the accuracy of most of paths is higher than 90%, because the paths with low performance have some characteristics in common. Overlapping path, irregular path trajectory and few training data are the main causes of low prediction performance. For the paths with high performance, the discriminative modeling method, non-overlapping path and regular path trajectory data are the main reasons.

To demonstrate the usefulness of HMM for the prediction of the destination, we compared it with dynamic time warping (DTW) method which is a template pattern matching method. Fig. 13 shows the average accuracy of prediction based on the data of the 10 users. The performance was measured in accordance with the different amount of sequence information. When the path of the progress is 0% that means to predict the destination without any previous sequence information at the origin, the prediction by DTW is impossible because of the short length of the input. However, HMM matches up the part of the path of the past. While the user moves toward the destination, the prediction accuracy can be increased. This result indicates that the proposed method is much more reliable to predict the final destination in the situation of less information.

5. Concluding remarks

In this paper, we have proposed a hybrid system for the location recognition and prediction. The proposed system addresses key issues of location-based services, such as location recognition and prediction. The proposed system uses a hybrid method combining kNN and decision tree to effectively recognize the locations not only in outdoor environment but also in indoor environment. For the location prediction part of the proposed system, HMM is used to identify user's next location using the location sequence together with other context information. To improve the performance, the prediction part of the proposed system automatically extracts the meaningful intermediate locations by exploiting G-means clustering algorithm. Moreover, to reduce the complexity of constructing probabilistic model and shorten the running time, the proposed system performs G-means algorithm only with the stay points. For the user's convenience, we developed user interface for recognition and prediction parts, and for efficient model management, we implemented functions of location and path managements to appropriately manage user's locations and paths.

To show the usefulness, we implemented a real working system on mobile devices. Experiments were performed to evaluate the accuracy of prediction model on mobile devices. We achieved the average prediction accuracy higher than 90% through the experiments. As a future work, we will investigate the incremental learning algorithm of location prediction model for more practical system to be adaptively learned through real-time data.

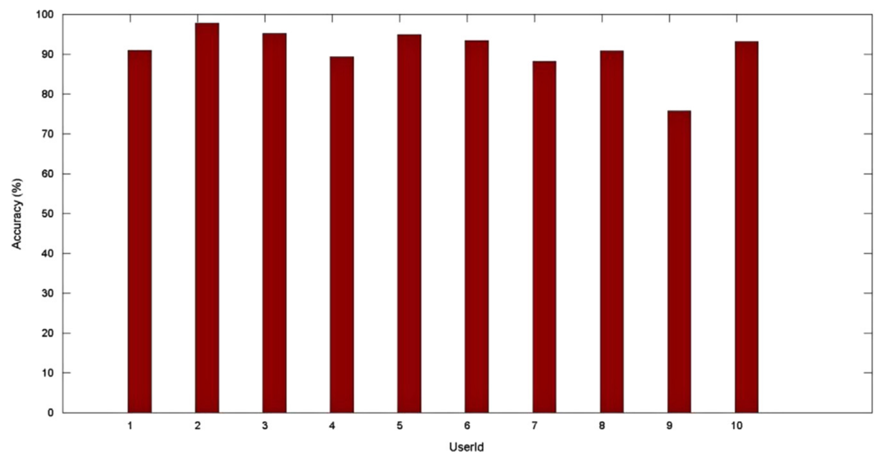

Fig. 12. Location prediction accuracy for each user.

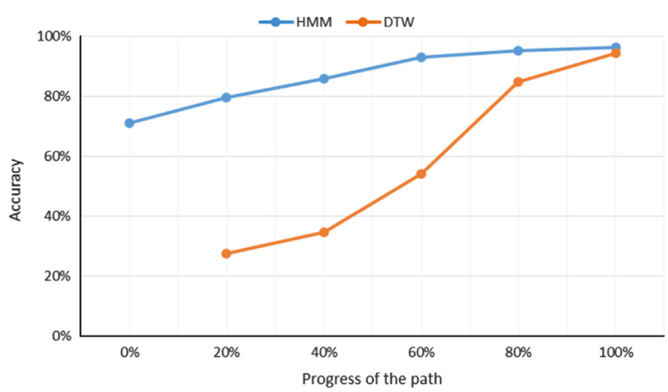

Fig. 13. Comparison of accuracy of HMM and DTW for location prediction.

Acknowledgments

This work was supported by the industrial strategic technology development program, 10044828, development of augmenting multisensory technology for enhancing significant effect on service industry, funded by the Ministry of Trade, Industry & Energy (MI, Korea) and made by a research team of Young-Seol Lee, Yong-Joong Kim, Sun-You Kim and Ho Xu in the softcomputing lab in Yonsei University.

References

[1] G. Chen, D. Kotz, A Survey of Context-Aware Computing Research, Technical Report TR2000-381, Dartmouth, November 2000.
[2] R. Löwe, P. Mandl, M. Weber, Context directory: a context-aware service for mobile context-aware computing applications by the example of Google android, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops 2012, pp. 76–81, 2012.
[3] H.Y. Noh, J.H. Lee, S.W. Oh, K.S. Hwang, S.B. Cho, Exploiting indoor location and mobile information for context-awareness service, Inf. Process. Manag. 48 (1) (2012) 1–12.
[4] H.S. Park, K. Oh, S.B. Cho, Bayesian network-based high-level context recognition for mobile context sharing in cyber-physical system, Int. J. Distrib. Sensor Netw. (2011) 2011.
[5] W. Mathew, R. Raposo, B. Martins, Predicting future locations with hidden Markov models, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 911–918, 2012.
[6] S. Cambes, M.O. Killijian, M.N. del Prado Cortez, Next place prediction using mobility Markov chains, in: Proceedings of the 1st Workshop on Measurement, Privacy and Mobility, pp. 1–6, 2012.
[7] M.O. Heo, M. Kang, B.K. Lim, K.B. Hwang, Y.T. Park, B.T. Zhang, Real-time route inference and learning for smartphone users using probabilistic graphical models, J. KIISE: Softw. Appl. 39 (6) (2012) 425–435.
[8] I. Burbey, T.L. Martin, Predicting future locations using prediction-by-partial-match, in: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, pp. 1–6, 2008.
[9] R. Simmons, B. Browning, Y. Zhang , V. Sadekar, Learning to predict driver route and destination intent, in: Proceedings of the IEEE Intelligent Transportation Systems Conference 2006, pp. 127–132, 2006.
[10] L. Liao, D. Fox, H. Kautz, Learning and inferring transportation routines, in: Proceedings of the National Conference on Artificial Intelligence, pp. 348–353, 2004.
[11] D. Ashbrook, T. Starner, Using GPS to learn significant locations and predict movement across multiple users, Person. Ubiquitous Comput. 7 (5) (2003) 275–286.
[12] J. Krumm, E. Horvitz, Predestination: inferring destinations from partial trajectories, in: Proceedings of the 8th International Conference on Ubiquitous Computing, pp. 243–260, 2006.
[13] J.A. Alvarez-Garcia, J.A. Ortega, L. Gonzalez-Abril, F. Velasco, Trip destination prediction based on past GPS log using a hidden Markov model, Expert Syst. Appl. 37 (12) (2010) 8166–8171.
[14] J. Petzold, A. Pietzowsk, F. Bagci, W. Trumler, T. Ungerer, Prediction of indoor movements using Bayesian networks, in: Proceedings of the 1st International Conference on Location- and Context-Awareness, pp. 211–222, 2005.
[15] G. Yavas, D. Katsaros, Ö. Ulusoy, Y. Manolopoulos, A data mining approach for location prediction in mobile environments, Data Knowl. Eng. 54 (2) (2005) 121–146.
[16] A. Monreale, F. Pinelli, and R. Trasarti, WhereNext: a location predictor on trajectory pattern mining, in: Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining, pp. 637–646, 2009.
[17] M. Morzy, Mining frequent trajectories of moving objects for location prediction, in: Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, pp. 667–680, 2007.
[18] J. Petzold, F. Bagci, W. Trumler, T. Ungerer, Comparison of different methods for next location prediction, in: Proceedings of the 12th International Conference on Parallel Processing, pp. 909–918, 2006.
[19] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, A.T. Campbell, NextPlace: a spatio-temporal prediction framework for pervasive systems, in: Proceedings of the 9th International Conference on Pervasive Computing, pp. 152–169, 2011.
[20] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, J. Hughes, Learning and recognizing the places we go, in: Proceedings of the 7th International Conference on Ubiquitous Computing, pp. 159–176, 2005.
[21] J.H. Kang, W. Welbourne, B. Stewart, G. Borriello, Extracting places from traces of locations, in: Proceedings of the 2nd International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, pp. 110–118, 2005.
[22] G. Hamerly, C. Elkan, Learning the k in k-means, in: Proceedings of the Advanced in Neural Information Processing Systems, pp. 16, 2003.
[23] A. Papliatseyeu, O. Mayora, Mobile habits: inferring and predicting user activities with a location-aware smartphone, in: Proceedings of the 3rd Symposium of Ubiquitous Computing and Ambient Intelligent, Advances in Soft Computing, pp. 343–352, 2008.
[24] J.H. Kang, W. Welbourne, B. Stewart, G. Borriello, Extracting places from traces of locations, in: Proceedings of International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, pp. 110–118, 2004.
[25] M. Zignani, S. Gaito, Extracting human mobility patterns from GPS-based traces, Wireless Days (WD), IFIP (2010) 1–5.
[26] I. Bisio, F. Lavagetto, M. Marches, A. Sciarrone, GPS/HPS and Wi-Fi Fingerprint based location recognition for check-in applications over smartphones in cloud based LBSs, IEEE Trans. Multimed. 15 (4) (2013) 858–869.
[27] T.M.T. Do, D. Gatica-Perez, Contextual conditional models for smartphone-based human mobility prediction, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 163–172, 2012.
[28] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1993.

[29] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd International Conference on Machine Learning, pp. 161–168, 2006.

[30] T.S. Lim, W.Y. Loh, Y.S. Shih, A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, Mach. Learn. 40 (3) (2000) 203–228.

[31] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[32] Y.-S. Lee, S.-B. Cho, An efficient energy management system for android phone using Bayesian networks, in: Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops, pp. 102–107, 2012.

S.-B. Cho received the B.S. degree in computer science from Yonsei University, Seoul, Korea and the M.S. and Ph. D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Taejeon, Korea. He was an Invited Researcher of Human Information Processing Research Laboratories at Advanced Telecommunications Research (ATR) Institute, Kyoto, Japan from 1993 to 1995, and a Visiting Scholar at University of New South Wales, Canberra, Australia in 1998. He was also a Visiting Professor at University of British Columbia, Vancouver, Canada from 2005 to 2006. Since 1995, he has been a Professor in the Department of Computer Science, Yonsei University. His research interests include neural networks, pattern recognition, intelligent man–machine interfaces, evolutionary computation, and artificial life. Dr. Cho was awarded outstanding paper prizes from the IEEE Korea Section in 1989 and 1992, and another one from the Korea Information Science Society in 1990. He was also the recipient of the Richard E. Merwin prize from the IEEE Computer Society in 1993. He was listed in Who's Who in Pattern Recognition from the International Association for Pattern Recognition in 1994, and received the best paper awards at International Conference on Soft Computing in 1996 and 1998. Also, he received the best paper award at World Automation Congress in 1998, and listed in Marquis Who's Who in Science and Engineering in 2000 and in Marquis Who's Who in the World in 2001. He is a Senior Member of IEEE and a Member of the Korea Information Science Society, INNS, the IEEE Computational Intelligence Society, and the IEEE Systems, Man, and Cybernetics Society.