PnLUM : System for Prediction of Next Location for Users with Mobility*

Le T. Nguyen, Heng-Tze Cheng, Pang Wu, Senaka Buthpitiya, Jiang Zhu and Ying Zhang
Carnegie Mellon University, Moffett Field, California, USA
{le.nguyen, hengtze.cheng, pang.wu, senaka.buthpitiya, jiang.zhu, joy.zhang}@sv.cmu.edu

ABSTRACT

As context-sensing smartphones and context-based services gain mainstream popularity, there is an increased interest in developing techniques that can predict user's future location. Location prediction capabilities can be used in many scenarios such as preheating home environment for the owner's arrival or for preemptive allocation of resources on cellular networks with expected heavy traffic. In this paper we present a soft classifier fusion technique for predicting a user's future location. Instead of one complex model for capturing user mobility patterns, we use multiple models, each focusing only on one aspect of location prediction such as time-location dependency or likelihood of location transitions, and fuse predictions from each model by considering their prediction confidences. Through the extensive evaluations on real-world data from the Nokia Mobile Data Challenge dataset, we show that our fusion approach outperforms the standard supervised classification algorithms such as SVM or decision tree.

1. INTRODUCTION

In this paper, we present a fusion of machine learning algorithms to model the movement of mobile device users. The fusion of machine learning algorithms are geared for predicting the user's next significant location based on the context of the user during the last 30 mins at his last significant location. We present an evaluation of the predictive power of the fusion model and the individual models on the Nokia Mobile Data Challenge (MDC) dataset [5].

Using probabilistic methods for modeling the movement patterns of users has been attempted in previous work, such as [2], [1], [4], [9]. [2], [13] and similarly [1] use a Markov model to predict a user's mobility pattern.Ashbrook and Starner use a similar method with n-th order Markov models to predict the next building a user would visit when observing the previous building visited by the user . Liu and Maguire also employ a Makovian approach to user location prediction [6]. These works are restricted to very coarse grained location prediction and require relatively clean location data. The Predestination system by Krumm and Horvitz [4] uses the Bayes rule for predicting future locations, and is based on the observation that a partially traveled route is usually an efficient path to the final destination. Ziebart et al. use a Makovian model for destination prediction as well as shorter term predictions of the route to be traveled [13]. This work uses smaller window of historic data to perform prediction. Buthpitiya et al. use a n-gram based model to detect anomalies in user behavior and predict the location of a user after a time window has elapsed [3]. Their work primarily focuses on anomaly detection rather than location prediction. Patterson et al. uses a Bayesian model augmented with information on the user's mode of transport to predict the users future location [9]. While this work provides a strong modeling approach, its predictive capabilities are limited by the lack of temporal information used in the model. Mayrhofer [7] build on existing work to extend location prediction techniques for context prediction in general. The work also develops a general architecture that uses a neural gas approach. The predictive capabilities of the models in existing work are limited as the models focus on location information, and ignore most other forms of sensor and context information (e.g., call-logs, accelerometer readings).

We use a collection of models, each to model a set of sensor and context information best suited it, and focus on fusing their outputs to provide accurate predictions of the user's next location and are able to outperform previous work in terms of predictive capability.

In this work, we make the following contributions toward the problems of modeling users' mobility patterns and predicting the users' future locations:

- We present a fusion of model for modeling human mobility.
- We show how this model is used for predicting the future significant locations of a user given the user's current context.
- We evaluate the fusion model's location prediction capabilities with extensive real-world data from the MDC dataset.

2. FEATURE EXTRACTION

In order to have a unified sensing framework, we consider all mobile contextual information being collected from "sensors". A sensor can be a piece of physical hardware such as accelerometers or gyroscopes. It can also be a "soft sensor" such as the value returned by the OS about whether the phone is charging or the current ringtone setting, etc. Before describing the training of the prediction model, we first describe which features are used in our work, what is the intuition of using them and how the features are extracted from both hardware sensors and soft sensors.

2.1 Phone Usage Features

*This material was prepared for the Mobile Data Challenge 2012 (by Nokia) Workshop; June 18-19, 2012; Newcastle, UK. The copyright belongs to the authors of this paper.

Features	Values
Call Log State	incoming call, outgoing call, missed call, incoming message, outgoing message
Ringtone Mode	normal, ascending, ring once, beep, silent
Charging State	charger not connected, charging, charging completed, charging continued after brief interruption

Table 1: The list of *phone usage* features used in this work.

Table 1 shows the *phone usage* features. The intuition of using these features is that phone usage patterns might be correlated with user's mobility patterns. For example, a user may decide to go home upon receiving a phone call from a family member, or to drive to a restaurant after making phone calls to his/her friends. In that case, there is a certain correlation between the *call log state* and the next location of a user.

For the *ringtone mode*, transitions of ringtone mode may imply changes of user's location. For example, a user might switch the ringtone mode from normal to silent before going to a meeting or a class. For the *charging state*, most users only charge their phones at certain locations (e.g., home or office). It is unlikely that a user will charge the phone while commuting or shopping in a store.

2.2 Motion Feature

The intuition of using motion information for next location prediction is that what a user does at time t may imply where he/she is going at time $t+1$. Motion features are extracted from the accelerometer data through a two level clustering. First, we discretize the raw 3-axis accelerometer readings using XMeans clustering [10]. The original three dimensional real value accelerometer readings are converted to one of the V symbols. In our experiment, we set $V = 200$. Thus, each 15 seconds (per 5 minutes) accelerometer readings are then converted to a sequence of 15×30 symbols as the accelerometer is sampled at 30 Hz.

Since the typical visit's duration is longer than 5 minutes, each visit contains multiple motion labels sequences. Therefore, we convert the data from time-domain to the frequency domain by counting the normalized frequencies of each motion symbol type and use the V-dimensional frequency vector to represent the motion characteristics of a visit. Then we apply a second level clustering on all V-dimensional frequency vectors in order to create at least N high-level motion clusters using XMeans. In our work, we empirically choose N to be 10. Now each visit is represented by a label such as "M_1", "M_2" or "M_{10}". Informally, this motion label describes user's most significant motions during a visit, such as whether he/she was "walking" or "driving".

2.3 Time Feature

Intuitively, a user's current location should correlate with time. For example, a user is more likely to be home in the morning and at work during the day. In our work, we analyzed the time-location correlation in details in order to estimate how accurately we can predict a user's location based on the time information.

In order to analyze the time-location correlation, we first discretize time into the following 48 time segments: weekday/0, weekday/1, ..., weekday/23, weekend/0, ... weekend/23. Each segment is one hour long and starts with the time denoted in the index. For example, weekday-1 describes the time segment between 01:00am to 01:59am UTC of a weekday. Using this method we can analyze users' location pattern dependent on time and the day of the week. To avoid over-fitting the model, we consider only weekday versus weekend instead of considering each day of the week (Monday, Tuesday, ..., Sunday) separately.

By summing up the total duration (in hours) a user spent at a certain location at a certain time we can identify user's most significant locations for a certain time segment. For example, when a user visits location L_1 starts at 10:40 and ends the 11:30, we count 20 minutes towards the label weekday/10 and 30 minutes towards the label weekday/11. Table 2 shows an example of the total time the user 20 from the MDC dataset spent at different time segments. For example on weekdays from 10:00 to 10:59 the user spend in total 53.7 hours at location L_1, 32.5 hours at location L_2, etc.

Time segment t_i		Location L_j				
Day	**Time**	L_1	L_2	L_{10}	L_{17}	...
weekday	10	53.7	32.5	0.8	2.6	
weekday	11	71	22.4	1.4	2	...
weekday	12	79.6	15	2.5	2.5	
...	...	...	...	...	...	

Table 2: The table shows for each time segment the sum of hours the user 20 spent at a certain location.

We observed from the MDC data that for each user there is a small set of locations that is significant to this user. E.g. for user 20 we observed that three locations L_1, L_2 and L_17 cover almost 90% of the time of all the visits. Figure 1 shows the three most significant places of user 20 from the MDC dataset. The sum of the duration of these three places covers almost 90% of the time the user spent at any places.

Figure 1 shows the duration distribution of the significant places of user 20. The X axis represents the time of the day. The values on the Y axis represent the percentage of time duration the user 20 spent at a certain location. From this figure we can observe that on weekdays the user typically spends mornings and evenings at location L_1 and during the day the user is at location L_2. Based on this information we can infer that location L_2 is a workplace whereas location L_1 is very likely to be the user's home.

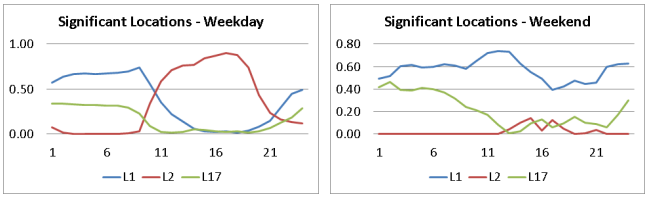

Figure 1: Distribution of three most significant locations for user 20 on a weekday and on the weekend.

For each time segment t_i we computed the location distribution $p_{t_i}(L_j)$ for each location L_j. For example for the user 20, the value of $p_{weekday/10}(L_1)$ is 56%. We used these probabilities to compute the entropy value for each time segment:

$$e_{t_i} = -\sum_j p_{t_i}(L_j) \cdot log(p_{t_i}(L_j)) \quad (1)$$

The lower the entropy, the less the uncertainty in a user's location at a certain time. For each user we computed the entropy value for each of 48 segments in order to build an entropy vector with 48 values:

$$E = [e_{weekday/0}, e_{weekday/0}, \ldots, e_{weekend/23}] \quad (2)$$

We then use the entropy vectors from all 80 users $E_{u_1} \cdots E_{u_{80}}$ and cluster them with k-means clustering, where k is empirically set to 4. Thus, the centroids of the clustering is also a vector with a length of 48. Figure 2 shows the centroids of 4 clusters. We gave each cluster a label which intuitively describes the properties user's location entropy:

1. *Party person*: spends night hours at various locations;
2. *Postman*: during the day he spends time at various locations, likely starts from places close to office in the morning then moves around the city and then comes back to office;
3. *Traveler*: in general, more difficult to predict locations;
4. *Family person*: in general, more predictable (low entropy).

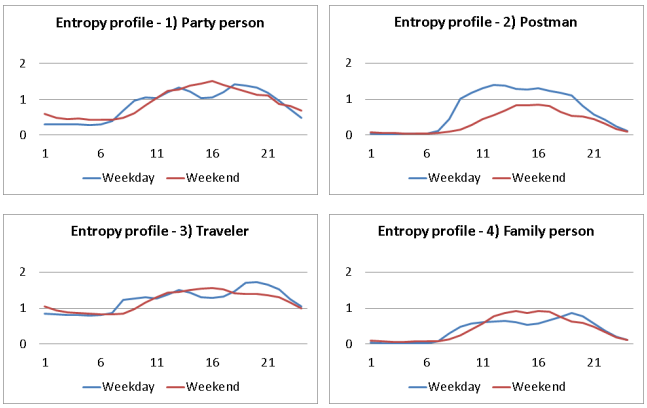

Figure 2: Entropy profiles of four different type of users.

In the following sections we will show how the location prediction accuracy depends on the entropy profile of each user.

3. PREDICTION MODELS

We developed and evaluated several statistical models to predict a user's next location given his/her current contextual information. Each model and their combinations are evaluated on the same testing data generated from the toy data provided with dataset A in the MDC's dedicated task 2. The testing data set contains 3,375 records. We use all remaining data as the training data set. In our experiments, we report prediction accuracy values on both training and testing data.

3.1 Time-based Location Prediction

The first model used in our work is the time-based location prediction model. This model uses only the time feature to predict user's next location. Specifically, we analyze the correlation between an end time of a visit and the location of the next visit. For example, for a certain user we observe that when a visit ends at 5pm then his or her most likely next location will be home.

We use Maximum Likelihood Estimations (MLE) to derive the probability $P(l_j|f_{endtime})$ where $f_{endtime}$ is the end time of the current visit and l_j is a possible location of the next visit.

In order to evaluate the performance of this model find an optimal next location l_{t+1} for a given $f_{endtime}$:

$$l_{t+1} = \arg\max_{l_j} P(l_j|f_{endtime}) \quad (3)$$

Table 3 shows the training and testing accuracy for each entropy profile/cluster. Each accuracy value corresponds to an average values for all user belonging to one cluster. As expected the prediction accuracy depends on the entropy profile of the user. Obviously, predicting next locations for a traveler is more difficult than predicting next location for a family person.

Entropy Profile	$time_{train}$	$time_{test}$
1) Party person	49%	50%
2) Postman	49%	56%
3) Traveler	38%	43%
4) Family person	60%	63%
Average accuracy	**48%**	**52%**

Table 3: Accuracy of the **time-based** location prediction

3.2 Dynamic Bayesian Network

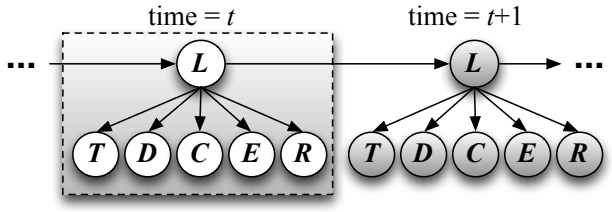

Figure 3: Graphical model of our Dynamic Bayesian Network. L denotes the next user location. T, D, C, E, R represent the time of day, day of week, call log state, charging state, and ringtone state, respectively.

Since the trace of the user locations and the contextual features are time sequences, it is natural to model the temporal dependencies among them using a dynamic Bayesian network. A dynamic Bayesian network (DBN) is a Bayesian network that represents sequences of variables. The DBN used in this work is shown in Figure 3. There are several assumptions implied in this model: First, the next location depends on the current location and all the contextual features. Second, the next location is conditionally independent of the previous locations given the current location. Finally, to simplify the model, we further assume that the contextual features are conditionally independent of each other given the next user location. In this work, the dynamic Bayesian network is implemented using the Bayesian network toolbox (BNT) [8]. The conditional probability tables are trained using MLE and the inference is done using the junction tree algorithm.

Since we can do inference on a single variable or a sequence of variables using DBN, we evaluated the DBN using the feature set described in Section 2 on two different tasks. The first task is Next Location Prediction as specified in the dedicated task 2 of the MDC [5]. In this task, since we only have the information of a user's current location and contextual features, the DBN reduces to a static Bayesian network. The second task is Location Sequence Prediction, where the system predicts a sequence of future locations given the sequence of contextual features while the locations are unobserved. For the second task, for each user we use the first 90% of data for training and the rest 10% for testing. The results are shown in Table 4. The accuracy of the second task is lower than the first task because it is inherently a harder problem. From the result we can see that using DBN for Next Location Prediction outperforms the baseline results using only bigrams, because

DBN models both the temporal dependency between locations and the relationship between a location and the contextual data of the mobile phone user.

Entropy Profile	Next Location Prediction	Location Sequence Prediction
1) Party person	45.7%	41.5%
2) Postman	46.4%	34.6%
3) Traveler	36.9%	27.4%
4) Family person	61.4%	55.0%
Average accuracy	**46.9%**	**38.6%**

Table 4: Accuracy of **DBN-based** location prediction.

3.3 Decision Tree, SVM and k-NN

The future location prediction problem can be framed as a classification problem, where for a given input of contextual data of the current visit, we try to output the correct user's next location. In our experiment, the following features are used for training a supervised classification model: 1) motion feature from accelerometer data, 2) place Id of current visit, 3) call log of the user, 4) charging status of the phone, 5) ringtones, 6) day of the week of the current visit, and 7) hour of current visit. The extraction of these features is described in Section 2.

Three classifiers are trained and tested on the toy dataset: 1) pruned decision tree (C4.5/J48), 2) one vs. one multiclass SVM with a polynomial kernel. 3) k-NN where k is empirically set to 5.

All the experiments in this section are done by using the API of WEKA [12] and its implementation of above classifiers. The results are shown in Table 5.

Classifier	Training Accuracy	Testing Accuracy
J48	69.0%	47.3%
SVM	70.5%	47.1%
k-NN	54.1%	47.7%

Table 5: The performance of different classifiers on toy dataset. Decision tree, SVM and k-NN algorithms have similar prediction accuracies on the testing data set.

To better understand the correlation between the contextual feature and the next location, we calculate the information gain of each feature with respect to the next location. Table 6 shows the information gain of different features. Values are averaged among all users.

Ranking	Feature	Information Gain
1	Current Location	1.12
2	The hour of current visit	0.92
3	Motion	0.60
4	Call log	0.33
5	Charing Status	0.19
6	The day of visit	0.14
7	Ringtone	0.13

Table 6: Information gain of different features with respect to next location.

As we can see in Table 6, the user's current location and time is the most informative feature to predict the next location. This suggests that people tend to stay at certain places at certain hours.

Motion is another significant feature. Yet it is not clear whether motion implies the user's current location which in turn implies his next location, or his current motion directly determines his future location.Comparing to the top three features, the charging status, call log, day of visit and ringtone are less significant features, which indicates that they are less relevant from user's location transition pattern.

3.4 Language Model-based Location Prediction

Based on Markov assumption that a user's next location depends solely on his current location, we applied a bi-gram language model for next location prediction. We consider each location as a "word" in a language and use the default smoothing and backoff in SRI language model toolkit [11] to train a bi-gram language model on the training data set. The model can then be used for estimating user's next location l_{i+1} given the current location l_i:

$$l_{i+1} = \arg\max_{l_j} P(l_j|l_i) \tag{4}$$

Table 7 shows the training and testing accuracy achieved by using the language-model approach.

3.5 Soft Classifier Fusion

In our work, we fuse the prediction results of described supervised classification models to achieve higher location prediction accuracy. Assuming we have n models. Each model i can estimate $P_i(l_j|f_i)$ where f_i is a set of input features of this model (e.g. time or current location) and l_j represents a possible next location. Our goal is to find the optimal next location for given input features by fusing the probability values of different models: $l_{t+1} = \arg\max_{l_j} \sum_i \alpha_i P_i(l_j|f_i)$.

α_i is a weight of a specific model. In our work, we exhaustively generated a large amount of weight combinations in order to find the best parameter set for the training data. Then we use this parameter configuration to perform location prediction on both training and test dataset.

The weight combination achieving highest training and testing accuracy is $\alpha_{lm} = 0.55$, $\alpha_{time} = 0.45$ and $\alpha_{other} = 0$ where α_{lm} is the weight of the language model-based approach, α_{time} the weight of the time-based model and α_{other} are the weights of all other models. Table 7 shows the prediction accuracy of 61% for both training and testing data when the mentioned weight combination is applied.

Entropy Profile	lm_{train}	lm_{test}	$fuse_{train}$	$fuse_{test}$
1) Party person	64%	60%	70%	73%
2) Postman	50%	43%	56%	52%
3) Traveler	53%	50%	62%	64%
4) Family person	55%	47%	64%	61%
Average accuracy	**55%**	**51%**	**61%**	**61%**

Table 7: Training and testing accuracy achieved by using only a language model (lm_{train} and lm_{test}) and by fusing the language model with time-based model for next location prediction ($fuse_{train}$ and $fuse_{test}$).

4. SUMMARY

In this paper, we discussed different approaches to attack the MDC's dedicated task 2. After experimenting with different prediction algorithms such as Decision Tree, k-NN, SVM and DBN,

we present a soft classifier fusion technique used for predicting user's future location. Instead of using one complex model for capturing user's mobility patterns, we use multiple models, each focusing only on a certain aspect of location prediction such as time-location dependency or likelihood of location transitions. Our approach fuses location predictions of the individual models while considering the prediction confidence of each model. Through the extensive evaluations on real-world data from the MDC dataset, we show that our fusion approach outperforms the standard supervised classification algorithms.

5. REFERENCES

[1] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[2] A. Bhattacharya and S. K. Das. Lezi-update: an information-theoretic framework for personal mobility tracking in pcs networks. *Wireless Networks*, 8(2/3):121–135, 2002.

[3] S. Buthpitiya, Y. Zhang, A. Dey, and M. Griss. n-gram geo-trace modeling. In K. Lyons, J. Hightower, and E. Huang, editors, *Pervasive Computing*, volume 6696 of *Lecture Notes in Computer Science*, pages 97–114. Springer Berlin / Heidelberg, 2011.

[4] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *Ubicomp 2006*, volume 4206/2006, pages 243–260, 2006.

[5] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T. Do, O. Dousse, J. Eberle, and M. Miettinen. The mobile data challenge: Big data for mobile computing research. In *Mobile Data Challenge by Nokia Workshop*, 2012.

[6] G. Liu and G. Maguire, Jr. A class of mobile motion prediction algorithms for wireless mobile computing and communication. *Mobile Networks and Applications*, 1(2):113–121, 1996.

[7] R. Mayrhofer. *An architecture for context prediction*. Trauner, 2004.

[8] K. P. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33:2001, 2001.

[9] D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *Ubicomp 2003*, volume 2864/2003, pages 73–89, 2003.

[10] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *In Proceedings of the 17th International Conf. on Machine Learning*, pages 727–734, 2000.

[11] A. Stolcke. Srilm - an extensible language modeling toolkit. pages 901–904, 2002.

[12] Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/˜ml/weka.

[13] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell. Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In *UbiComp '08*, pages 322–331, 2008.