Next Place Prediction using Mobile Data

Le-Hung Tran
École Polytechnique
Fédérale de Lausanne
(EPFL), Switzerland
hung.tranle@epfl.ch

Michele Catasta
École Polytechnique
Fédérale de Lausanne
(EPFL), Switzerland
michele.catasta@epfl.ch

Luke K. McDowell
Dept. of Computer Science
U.S. Naval Academy
Annapolis, MD
lmcdowel@usna.edu

Karl Aberer
École Polytechnique
Fédérale de Lausanne
(EPFL), Switzerland
karl.aberer@epfl.ch

ABSTRACT

Recently, location-based applications and services for mobile users have attracted significant attention. In this context, one challenging problem is predicting the future location of a mobile user given his or her current location and associated metadata. Solving this problem enables many interesting applications such as location-aware mobile advertisements, traffic warnings, etc. In this paper, we present an approach based on user-specific decision trees learned from each user's history. The classification tree is built based on simple, intuitive features with some mobile data-specific enhancements. We demonstrate the performance of our approach by evaluating with a real-life dataset provided by Nokia, and show that it outperforms a simpler baseline.

General Terms

mobile computing, prediction

1. INTRODUCTION

This project was conducted as an applicant for Dedicated Task 2 of the Nokia Mobile Data Challenge. The main objective is to predict the next destination of a mobile user based on the current context, such as the current location and time. Mobile data comes from a data collection campaign that was carried out by Nokia and its Swiss partners near the Lake Geneva region. The mobile phone data was collected on 24/7 basis over a period of about 14 months.

In this project, instead of trying to apply complex mathematical models [4], we investigate the performance of a tailored implementation of a decision tree. In particular, we create one decision tree per user, where most of tree structure is built using a standard tree learner. However, we make several modifications to the standard tree construction and usage based upon our analysis of the domain. First, the root of the tree is constrained to always make an initial split decision based on the user's current location. Second, we modify the standard prediction decision made by leaf nodes to adjust for additional time-based correlations in our data (see Section 4.1.2). Finally, cross-validation is used to optimize several decisions related to the tree and to pre-processing of the data, such as when older training data should be discarded.

Our decision to use decision trees instead of some other technique was driven by the advantages of using a simple (and well-studied) tool that we could inspect while running. This decision enabled us to manually refine the prediction branches, and see instantaneously if our idea was improving the prediction capabilities – an important capability due to the variability and noisiness of the data. Most of the refinements we implemented were guided by common sense. We argue that, in this way, we have been able to exploit most of the behavioural data related to a user.

This paper is organized as follows. The next section introduces the problem and the provided data, while Section 3 describes the pre-processing that we performed on this data. Section 4 describes our training and prediction approach. Section 5 explains our experimental results, while Section 6 presents some related works. Section 7 concludes the paper.

2. PROBLEM STATEMENT

This section summarizes the main tasks and specifications of the Dedicated Task 2: Next Place Prediction. Separate training and test data sets were provided. The test set consists of data related to a collection of time intervals just before a user transitions to a new place (which marks the end of a "visit"). The task involves predicting the next destination of a user given the available data from a specific time interval just before such a transition. The end point of the time interval always corresponds to the leaving time of a visit, and the start point of this interval is set to 10 minutes before the start time of the visit itself. Ground truth for each location is represented by place IDs, which are provided as annotations for sequences in the training data.

From the challenge statement, there are some important

points to be considered:

- GPS information is not provided for each visit. Instead, we only know the place ID of each visit. The only geographical related information provided about these places are the distances between them. However, these distances were classified into 4 groups: less than 1km, 1-5km, 5-10km, or 10km+, which provides limited information that could be useful for prediction. We also have the acceleration data which was collected from the accelerometer of the phone, but the data is very sparse and there are very large discrepancies between users.

- Semantic labels such as "home", "work", etc. are not supplied for each place, but only a generic place ID.

- The test dataset is the continuation (in time) of the training dataset. In addition, prediction can be made based only on the knowledge about the current state, using a ten minute window. This limited temporal informaton prevents us from usefully employing a transition model such as Markov chains.

- In the test dataset, there are cases where the user visits new places which did not occur in the training dataset. The challenge instructions indicate that we should predict these new place as place ID 0.

3. DATA PRE-PROCESSING

This section describes helpful processing of the data that we performed prior to classification. These steps include the detection of likely homes and workplaces (Section 3.1), the creation of a new feature that identifies user-specific holidays (Section 3.2), steps to create synthetic "new place" labels (Section 3.3), and data cleaning processes (Sections 3.4-3.5).

3.1 Home and Workplace Detection

Users visit their home and workplace most frequently. Therefore, it is very important to detect these places for each user. Firstly, if we know a user's home and workplace, we can predict more easily the next visit of that user to home or workplace. Secondly, if the home and work locations are known, we can easily extract useful information about the user such as working time, changing home, changing workplaces, going on vacation, etc. which leads to better prediction for all places. Thus, detecting the home and workplace of each user is the first and most important pre-processing step.

Typically, most users are at home at midnight and at their workplace in the morning and afternoon. For these users, a simple method is to identify the places where the user usually is at a certain time period of the day. For example, the place where the user usually is at midnight has high probability of being the home of the user. However, there are several issues we must address. First, users' vacations change their movement patterns, e.g. staying at different places at night and not going to work. Our implementation used the different chararacteristics of these periods to identify such vacations and discard them as noise. We will investigate more on vacation on the next section about holiday detection.

Place	Properties
Home	- user stays at this place most of the time between 23:00 to 4:00. - user visits this place at least 20 days in a duration of 30 days. - on weekdays, user usually leaves this place in the morning during 6:00 to 10:00 and returns in the afternoon between 16:00 to 20:00.
Work	- user stay at this place most of the time between 9:00 to 12:00 and 13:00 to 17:00. - user visits this place at least 15 days in a duration of 30 days. - on weekdays, user usually goes to this place in the morning during 7:00 to 10:00 and leaves this place in the afternoon between 16:00 to 18:00.

Table 1: Home and workplace properties.

The second problem is that users may change their home and work place one or several times. As we have to predict the future movements of a user, intuitively, we only need to consider the last home and workplace of the user. However, the decision tree will learn predictions which are highly dependent on visit frequency, and excluding prior home and work locations would eliminate many useful visits that characterize a user's typical patterns. To include all such information while still recognizing the current home and work locations, we adopt the following strategy. First, we find all the home places of users. Then, we mark all of these "home" places with the same place ID as the last home place. A similar process is done for workplaces.

Another challenge is that there are some users who go home and go to work at very uncommon times. To deal with such users, one solution is to detect the time that the user is often at home and at work by analyzing the most frequenctly visisted places. Still in this case it is not simple to distinguish the home and workplace of such users. There are several approaches to identify home and workplace based on their properties, such visit duration, which can overcome this issue. However, in our training data we found that there were only a few users who had this kind of behaviour, and that more complex methods were not guaranteed to work well. Thus, we only implemented a mechanism to highlight users with irregular behaviour (that did not meet expected patterns). For such users, we were able to manualy adjust the presumed home time and working time as needed.

Table 1 shows the properties of home and workplace which our detector uses to identify all the home and workplaces of a user. By using a combination of these properties, our detector can naturally cope with noise induced from user vacations, visits to friends' houses, etc. Using these properties, combined with the adjustments described above, we identified the probable homes and workplaces of each user. On the training data, we estimated that the detector correctly identified home and work locations about 95% of the time.

3.2 Holidays Detection

Holiday detection is important for prediction because the behaviour of users changes during holiday periods. However, determining the holidays for a specific user is not simple. The time and duration of holidays depends on the user's job and geographic location. For instance, users may be a "normal" worker who has only public holidays, a Ph.D. student or university staff member who has several school breaks, or an undergraduate student who has multiple long holidays like summer holiday, Easter, etc. In addition, users also have one or more personal holidays during each year. Finally, public holidays vary between countries or states in the same country.

As the holiday schedule varies for each user, we do not aim to develop a perfect detector. Our approach to this problem has two steps. The first step is creating a holiday knowledge base for multiple user types. In our data, all of the users are from nearby Lausanne, Switzerland. Thus, the knowledge base is initialized with three types of holidays based on Swiss holidays: normal user, university staff and student. For each user, we manually estimated the user type by examining the training data, and used this type to create a personal holiday calendar. Next, we added additional personal holidays to each user's calendar by inferring from the visit sequence. In particular, any weekday where the user did not go to work was assigned a certain probability of being a holiday. These probabilities were increased when days without work were consecutive, and/or were consecutive with public holidays. Finally, a certain probability threshold was applied to get the final personal holiday calendar.

3.3 New Places Detection

Since the challenge task also involves predicting "new" places, the overall performance improves if we can detect the first time that a user goes to a new place. Unfortunately, this is challenging since there aren't any previous patterns for new places like new home, new work places, new friend's house, new restaurants, etc. We can, however, identify new places that are related to sight-seeing or travelling. Users often visit these new places on weekends or while on holiday, and a key property of these places is that they are visited very few times.

Based on this assumption, we developed a method to learn some of the "new place" visiting habits of users. First, our detector finds all sight-seeing and travelling places in the training dataset based on the properties described above. Next, all of these places are considered as one **New place** and marked with place ID 0 in the training data. Thus, we can accumulate sufficient visit frequency for these likely "new" places, and the decision tree can predict place ID 0 for some of the test set visits.

3.4 Gaps Removal

In the training data, some of the data is marked as "untrusted transition" and "untrusted end visit," perhaps because the user's device lost the signal for some period of time. Removing all such untrusted visits provides more certain visit information, but also discards a significant amount of potentially useful information. Thus, during data preprocessing, we remove such visits only if there is a large gap with their consecutive visits.

Features	Description
PlaceID	place ID of the visit
isHoliday	true if the visit is in holiday; false otherwise
isWeekend	true if the visit is weekend; false otherwise
Weekday	weekday of the visit
LeavingTime	the end time of the visit
Duration	duration of the visit

Table 2: Features considered for predicting the user's next visit place.

3.5 Old Data Removal

Since the data was collected over a long period, the oldest part of the data may not reflect the current movement habit of each user. Such "stale" data may decrease the performance of the prediction. In order to avoid this problem, we used the home/work detector to identify time periods when a user appeared to have a significant change in behavior (e.g., a new job). The optimizer (see Section 4.3) then considered removing the data prior to this time from the training data.

4. TRAINING AND PREDICTION

4.1 Decision Tree

To classify the test set visits, only the information from a 10-minute window of a visit is available. Thus, we cannot take advantage of the visit sequence for prediction. Instead, we have a standard classification problem where the objects to be classified are the visits of users, based on features related to their current state. Since the place IDs are different for each user, we learn a user-specific decision tree for prediction.

4.1.1 Decision Tree Building

A decision tree is a hierarchical structure for classifying objects, composed of nodes that correspond to primitive classification decisions. At the top of the tree is the root node that specifies the first dividing criterion. The root, and every non-leaf node, has two or mode child nodes, which can be thought of as classifying further all the visits of the user. Associated with each node, in addition to a dividing criterion, is a set of visits. The root node contains all the visits of the training data, while child nodes contain those all visits that match the dividing criteria along the path from the rooot to that node. In this way, the set of visits can be sub-classified into finer subgroups, where the nodes at the bottom of the tree, the leaves, contains the smallest groups.

In our approach, the features used for these decisions are simply those that were extracted from the visit sequence data since these features were the most related and useful data to our prediction task. These features are listed in Table 2, and one example tree that uses these features is shown in Figure 1. In this example, the first dividing feature is *PlaceID*, and the features along the path that is expanded are *isHoliday*, *isWeekend*, *Weekday*, then *LeavingTime* (time of day), and finally *Duration*.

To build a tree for each user, we used Weka J48 [11], an implementation of the C4.5 decision tree algorithm. This

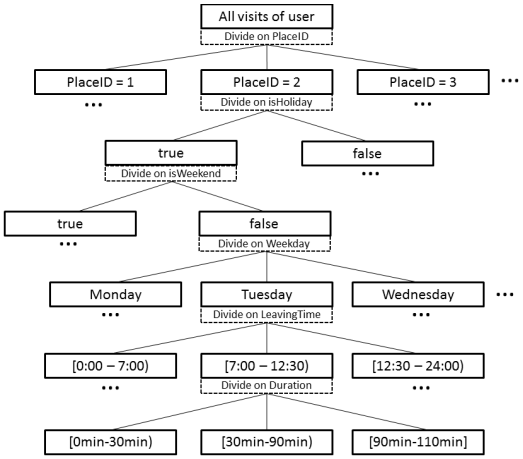

Figure 1: Decision tree example.

algorithm automatically decides upon the best dividing features to use for each node, and automatically discretizes continuous features like *LeavingTime* and *Duration*. However, because we believed that the current place was strongly associated with predicting the next place of a user, we constrained the tree so that *PlaceID* was always the root node of the tree.

4.1.2 Prediction Using the Decision Tree

In order to predict the next visit of a user, we use the features of the current visit to descend the decision tree from the root. After this process, we obtain a leaf node, which contains the set of training set visits that most closely match the current visit. In the best case, all visits in this set have the same next place ID, and this place ID is returned as the predicted next place.

Often, however, there are several distinct place IDs within the leaf node's visit set. Potentially, we could select the most frequent place ID from this set. However, this approach leads to a user's home and workplace almost always being chosen (as most probable). Instead, we use another strategy based on difference in leaving times. In particular, we select from the visit set the visit v with the smallest difference between its leaving time and the leaving time of the test set visit. If this difference is less than a user-specific threshold, then we use the next place ID of v as the prediction. If not, then we conclude that the decision tree has not been able to obtain a highly confident prediction. In this case, we always predict the estimated home or workplace of the user, based on the time and day of the current visit.

We also considered using call log data as a criteria for choosing the most similar visit from a leaf node's visit set. The intuition is that a user may often call or receive a call from one or several certain phone numbers before going to specific places. However, after analyzing the dataset, we found that there was little correlation between the call log and the next visiting place of users.

4.2 Prediction Using Calendar Data

Calendar data, which consists of a user's appointments, can be very useful for place prediction since appointments are often associated with a location. For the challenge data, however, this approach has limited applicability for two reasons. First, there were only a few users with a sufficient amount of calendar data. Second, because the calendar entries were anonymized with a hashtag, most entries appear only as once-only appointment IDs with limited use for prediction. As such, for prediction we used only the recurring appointments in a user's calendar.

Thus, appointments could be highly predictive of a user's next place, but this occurred relatively infrequently. To exploit these characteristics for prediction, we first attempted to predict the next place by matching the current visit to an appointment that was both time-relevant (discussed further below) and had been seen during the training data. If a high-confidence prediction could not be made using appointments, then we used the decision tree method instead. In total, the calendar data determined only 29 out of 5924 predictions, but showed high accuracy (i.e., 82.75%).

Matching an appointment to the user's current visit presents two problems related to time. First, for reminder purposes, some users may have the habit of setting the appointment's time before the actual time of the appointment. Second, when predicting the next place after a current visit, we have no knowledge about the time when the user will be at the next place.

We solve the first problem by doing an exhaustive search for a user-specific time offset using a simple checking on the appointments in the training dataset. In particular, we look for the case where appointments with the same tags are always followed X minutes later by an arrival at the same location. We search over a small number of possible offsets X ranging from 5 minutes to 3 hours. Thus, the computation cost is small. Given a user-specific offset X, we can then pair each training set appointment with a set of associated visits.

To solve the second problem, we perform a simple check to see if the current visit is at about the same time as a previously-seen appointment in the calendar. Of course, we have to define a threshold here, e.g., the visit may be one hour before or after the appointment. If the visit satisfies the time condition of the appointment, we then compute the set of training set visits that are associated with that appointment. To be conservative, we recognize a complete match only if the current place ID of one of these visits matches the current place ID of the test set instance that we are trying to make a prediction for. If so, then the next place ID associated with that matched training instance becomes the prediction result.

4.3 Parameter Optimizer

Our approach relies on several decisions which can vary based on the user, such as the holiday type, percentage of old data removal, time threshold for leaf-based predictions, and time thresholds for appointment-based prediction. In order to find good parameter settings, we implemented a simple optimizer.

The optimizer does a heuristic search over the set of reasonable parameter values. The objective is achieving the

User type	#user	Weka J48	J48 with Holiday	Proposed approach
(A) Users with simple movement patterns	31	61.11%	62.34%	71.85%
(B) Users with heterogeneous movement patterns	24	42.90%	44.39%	54.27%
(C) Users that change behaviour at the end	8	35.44%	34.94%	53.72%
(D) Users that change behaviour in test dataset	5	33.24%	33.82%	45.13%
(E) Users that have small data or lack trusted data	16	38.13%	39.04%	46.36%
Average accuracy		49.78%	50.86%	61.11%

Table 3: Prediction accuracy for each type of user, and averaged over all users.

highest accuracy on the cross validation part of the training data. However, the optimizer always keeps high priority for basic common sense and for reasonable values of the parameters. For example, the optimizer will not remove the old data if the prediction accuracy doesn't improve by more than 5 percent.

The parameters are based on our intuition and have a small range. In addition, the size of the visit sequence and calendar data for each user is small. Thus, the computation cost of the optimizer is acceptable.

5. RESULTS

In this section we demonstrate the experimental results of our approach on the Nokia Mobile Challenge training dataset. Since we were provided with known next places only for the "setA" training data, we must manually split the data into training and test datasets for evaluation purposes. For each of the 80 users, we create separate training and test datasets. Since there are discrepancies in the number of visits per user, we try to make the test data set contain the visits of the last three months. Also, the average number of trusted visits per normal user in the last three months is around 120 visits. Thus, we chose 120 as the maximum number of visits per user in the test dataset to avoid having the results dominated by the few users who have many recorded visits. For users with more than 120 visits in the last three months, we randomly select 120 visits. For users with less than 240 total visits, we simply take half of the data as the test dataset.

We compare our approach with a baseline method that also uses Weka J48 [3]. The baseline method is implemented as follows: First, we extract from the dataset the same features as in our models (Table 2), except for *isHoliday* (since this is a more complex feature that we need to detect); Next, we run Weka J48 to build the tree and do cross validation. Weka J48 implements the C4.5 decision tree algorithm which is considered to be a strong method for generating decision trees. Furthermore, Table 2 includes an important intermediate result: namely, the baseline enhanced with the *isHoliday* feature. To assign a specific holiday type (i.e., normal, staff, student, personal), we choose the one that leads to the best accuracy during the cross validation process. Since our proposed approach is based on decision trees with additional improvements targeted for this specific task, it is reasonable to make a comparison between our approach and these baselines.

The bottom row of Table 3 shows the average prediction accuracy of our next visit place prediction model using our approach in comparison to the baseline J48, and J48 with Holiday. The average accuracy of our proposed approach is 61.1% which is almost 12 percentage points better than the baseline method using Weka J48. Analyzing the results we found that the performance varies significantly between users, with accuracy of more than 80% for some users but less than 30% for others.

After detailed investigation, we found that there are some user characteristics that greatly affect the accuracy of our approach. Based on these characteristics, we classified users into five categories as shown in Table 3 (one user can be classified into several categories). As can be seen from the table, our approach works best for users who have simple and repeated movement patterns (group A), leading to an accuracy of more than 70%. The results are also fairly good for users who have heterogeneous movement patterns (group B). For users that change their behaviour at the the end of the training dataset (group C), if there are enough visits, then our approach can still manage to achieve good results; otherwise, the accuracy is much lower than the average. The most difficult users are those who change their behaviour only in the test dataset (group D), and users whose visit sequence data is small or lacks trusted data (group E). As our prediction totally depends on the training dataset, it cannot provide good predictions for these users. For such users, our method generally predicted the next place as one of the 2-3 most frequently visited places in the training dataset. As a result, the prediction accuracy for these users is poor.

6. RELATED WORKS

Even before the widespread adoption of smartphones with integrated GPS sensors, various studies have been conducted on predicting the next visited place of a mobile user [12][7]. One of the main reasons is that resource allocation in Personal Communication Systems network can be optimized when the user mobility patterns are known [5]. Most of the past and recent studies are based on user trajectories [8][9]: our work differs for the fact that predictions can be made only using the knowledge about the current state. This limited temporal informaton prevented us from usefully employing a transition model such as Markov chains [2], or nonlinear time series analysis [10].

Furthermore, Wi-Fi and GSM data has been shown to have even more potential than the GPS data when it comes to next place prediction [6]. The Wi-Fi and GSM datasets provided for the challenge are anonymized with a different seed for each user, hence we were not able to extract shared patterns among different users.

Our approach focuses on adapting well-known algorithms to our specific scenario. We argue that this choice will allow future developers to adopt complex Big Data libraries

(like Mahout[1]), and reproduce our results with just a combination of already implemented algorithms (with few if no modifications).

7. CONCLUSIONS

This report presents a classification approach using decision trees to predict the next place of mobile users. In our proposed approach, the decision tree was built using several enhancements and domain knowledge that affected both the top-level tree structure and the final predictions implied by the leaf nodes. To effectively predict the next visit place of users, the feature extraction and data pre-processing steps were carefully designed based on analysis of the data. Besides the main features which were extracted from users' visit sequences, calendar data with appointments was also used in some cases to improve the prediction. Additionally, we implemented an optimizer to find the best parameter combination for each user, since users had widely varying behavior. Finally, the performance of our approach was demonstrated by the results of the experiments on the real-life dataset of 80 mobile users provided by Nokia.

8. REFERENCES

[1] Y. Chon, H. Shin, E. Talipov, and H. Cha. Evaluating mobility models for temporal prediction with high-granularity mobility data. In S. Giordano, M. Langheinrich, and A. Schmidt, editors, *PerCom*, pages 206–212. IEEE, 2012.

[2] S. Gambs, M.-O. Killijian, and M. N. n. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, MPM '12, pages 3:1–3:6, New York, NY, USA, 2012. ACM.

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[4] D. Hand. Classifier technology and the illusion of progress. *Statistical Science*, 21(1):1–14, 2006.

[5] D. Katsaros, A. Nanopoulos, M. Karakaya, G. Yavas, Ö. Ulusoy, and Y. Manolopoulos. Clustering mobile trajectories for resource allocation in mobile environments. In M. R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, and C. Borgelt, editors, *IDA*, volume 2810 of *Lecture Notes in Computer Science*, pages 319–329. Springer, 2003.

[6] A. Kirmse, T. Udeshi, P. Bellver, and J. Shuma. Extracting patterns from location history. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 397–400. ACM, 2011.

[7] G. Liu and G. Q. M. Jr. A predictive mobility management algorithm for wireless mobile computing and communications. In *IEEE International Conference on Universal Personal Communications (ICUPC'95)*, Tokyo, Japan, Nov. 1995. http://www.it.kth.se/labs/ccs/ccs-publications.html.

[8] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, editors, *KDD*, pages 637–646. ACM, 2009.

[9] M. Morzy. Prediction of moving object location based on frequent trajectories. In A. Levi, E. Savas, H. Yenigün, S. Balcisoy, and Y. Saygin, editors, *ISCIS*, volume 4263 of *Lecture Notes in Computer Science*, pages 583–592. Springer, 2006.

[10] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. *Pervasive Computing*, pages 152–169, 2011.

[11] I. Witten, E. Frank, and M. Hall. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

[12] G. Yavas, D. Katsaros, Ö. Ulusoy, and Y. Manolopoulos. A data mining approach for location prediction in mobile environments. *Data Knowl. Eng*, 54(2):121–146, 2005.

[1] http://mahout.apache.org