Periodicity Based Next Place Prediction

Jingjing Wang
Department of Computer Science
University of Illinois at Urbana - Champaign
Urbana, IL 61801, US
jwang112@illinois.edu

Bhaskar Prabhala
Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802, US
bup131@psu.edu

ABSTRACT

Location Prediction has attracted a significant amount of research effort. Being able to predict people's movement benefits a wide range of communication systems. It is both an interesting and challenging problem which applies to many different settings. In this paper, we describe the approaches taken for challenge of next place prediction. This task involves predicting the next destination of a user given the current context. We build a user specific model for each user that learns from his/her mobility history. We then apply the model to the current context to predict where the user will go next. We describe the algorithm, results and observations in the following sections.

1. INTRODUCTION

The challenge in this work is to predict the next location of a user based only on its current context. While we do have the historic sequence data of the users in the training phase, in the testing phase, our prediction should be based only on the single current point rather than the past sequence. Therefore, for this task, typical location prediction algorithms based on structure of sequential patterns[1, 10, 4, 9, 6] are not applicable.

Our initial observation is that user behavior exhibits strong periodic patterns. We compute user visit frequency of places and aggregate total time spent in places and observe regularity in user behavior. As shown in Figure 1, which plots the histogram of one typical user (user 8)'s visit frequency distribution over places, there are several places dominating the sequence. Visiting time distribution has similar patterns. Then we perform periodicity analysis for the top most frequent places based on *Fourier Transform* and *autocorrelation*[7]. We found strong daily and weekly periodicity for Place 3 (the most frequent one) and some weak daily or weekly periodicity for others. This motivates the use of periodicity in making predictions.

Another interesting observation is if we plot the histogram of places being visited after Place 3 (the most frequent place) and the histogram of places being visited after Place 13 (a relatively less frequent place), as shown in Figure 2. We see that after Place 3, there are still two places dominating, but after Place 13, there is only one place dominating. This observation inspires us to consider a separate strategy for different places. Since we can give only one candidate for the next place when making prediction, for some places, we may be able to determine the next place simply based on majority voting with high confidence; while for other places we may further explore temporal information to make decision.

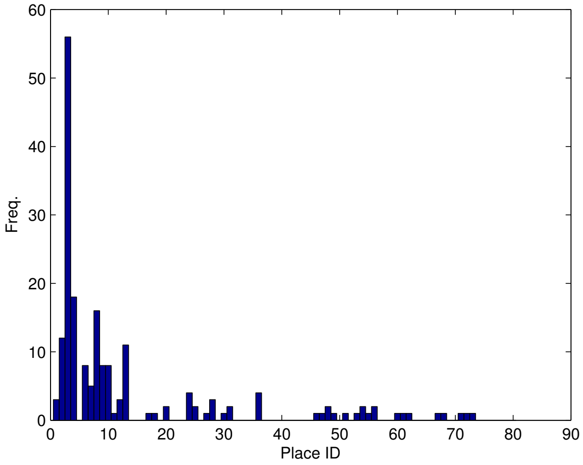

Figure 1: Visit Frequency Distribution over Places

The rest of this paper is organized as follows. Section 2 briefly describes the characteristics of the data set. Section 3 describes two main models and the algorithms we develop based on these models for prediction. Section 4 shows experimental results for different models. Section 5 results for submission, and some discussion. Section 6 summarizes our conclusions.

2. DATA SET

Details of the data set including characteristics of the data set, partition and availability of different portions for various challenge tasks are described in [5]. In this section, we briefly mention the relevant data set details for our models and algorithms.

For our task, we focus on `visit_sequence_20min.csv` files

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.
.

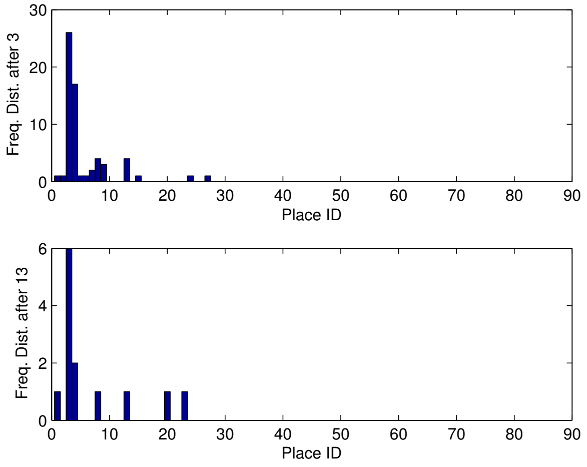

Figure 2: Freq. Dist. after Place 3 and Place 13

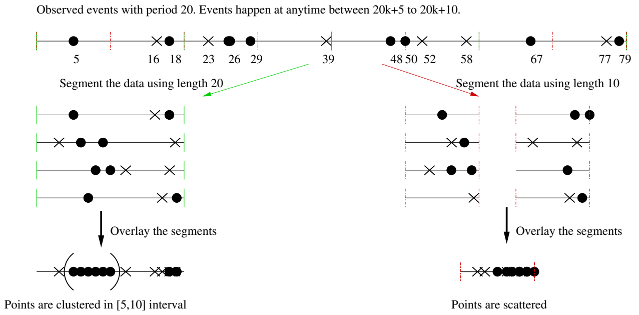
Observed events with period 20. Events happen at anytime between 20k+5 to 20k+10.
5
16
18
23
26
29
39
48
50
52
58
67
77
79
Segment the data using length 20
Segment the data using length 10
Overlay the segments
Overlay the segments
Points are clustered in [5,10] interval
Points are scattered

Figure 3: Intuition of the Periodicity Model

containing sequence of place visits which are longer than 20 minutes; besides the userID, each entry consists of current placeID,start and end (unix) times with corresponding time zones, if the visit's start, end and transition to the next location are to be trusted.

We consider two cases: all transitions and only trusted transitions in our models and algorithms.

3. MODELS AND ALGORITHMS

In this section, we describe the techniques used for our modeling of the data and the algorithms implemented for this challenge task. First we detail the periodicity based model. In section 3.2 we describe multi-class classification model. We give how the algorithms are generated accordingly in section 3.3.

3.1 Periodicity Based Model

This model is inspired by the method proposed in [8]. To illustrate, here we use the running example in that work. Suppose an event has a period 20 and we have a sequence of observations of that event. If we overlay the observations with correct period, we can see that the observations will form dense clusters along the timeline, which gives a time distribution of the periodic events, as shown in Fig. 3

Human movements are strongly determined by periodic patterns[3]. For example, if a person always goes to work from home at 9am in the morning on weekdays, given it is 9am and this person is at home in a future weekday, it will be very likely he/she will go to his/her workplace as the next destination. Based on this intuition, we directly build time information into our first prediction model. As mentioned in Section 1, we get the following observations for each user: there are some places (the number usually varies from 1 to 5) dominating the entire movement sequence in terms of both time and visit frequency; people tend to visit these places regularly, which justifies our method. In this model, we first build a profile which contains temporal information for each user. And then based on the current place, along with the profile, we predict the next place.

Since the data does not contain relationship information between users, we build user-specific models. The following discussions are for one specific user. Our basic assumption is, given the user is currently at place p_{cur}, the next place he/she is going to visit should be primarily determined by the location p_{cur} and the time interval in which he/she stays at p_{cur} . Therefore for p_{cur}, we check the conditions under which the user goes to a certain next place p_{next}. Here the condition we use is a time distribution $dis.(p_{cur} \rightarrow p_{next})$ which is computed by overlaying all the time intervals in which the user decides to "jump" to p_{next}. The period we use here is one week and the granularity is one hour. Now, if we want to predict the next place after staying at p_{test} at time interval t_{test}(this can be transformed to a time distribution $dist(t_{test})$ the same way as the above), we can use dot product $dist(t_{test}) \cdot dist(p_{cur} \rightarrow p_{next})$ as a similarity measure between the temporal condition at present and the condition of a certain transition. Then we choose the place by maximizing this similarity measure as the next location. Here we can directly use the dot product without worrying about scaling because the overlay mechanism naturally weighs the distribution by giving the time intervals which correspond to frequent places more count simply due to the more visits.

As mentioned in Section 1, for some places majority voting (equivalent to a *1st-Order Markov Predictor*) will be very efficient. We observed this for places that are not frequently visited. We capture this observation in the model by employing a separate strategy. If the confidence of majority voting is higher than a predefined threshold, we use it; otherwise, we consult the temporal similarity measure. The algorithm is described in Algorithm 1.

3.2 Multi-Class Classification

Giving all the current information of a user, it is natural to model this task into a multi-class classification problem. For each entry, we extract features from the current context and use the next place as the label. There are several issues to be highlighted here. First, the class labels are highly imbalanced since there are dominant places. Meanwhile, many of the minority classes have very few data samples, e.g., only 1 or 2 samples. This will cause the classifier to favor the majority classes. And since there are not enough samples for the minority classes, the accuracy on the minority classes will be very low, as we will see later in the experiments.

From the discussions in the previous sections, we see that time and location features are very discriminative in predicting the next place. Thus we only extract features from time and location. While we ignore other context which might

```
Algorithm 1: Periodicity Based Model (for a specific user)
Input: current place $p_{test}$, current time interval
       $t_{test}$,threshold, profile of the user
       $\{dist p_i \rightarrow p_j\}, p_i, p_j$ could be any places the
       user has visited
Output: next place $p_{predict}$
if there exists a place $p_n$, s.t.
$confidence(p_{test} \rightarrow p_n) > threshold$ then
    $p_{predict} = p_n$;
end
else
    $p_{predict} = \arg\max_{p_j}(sim(dist(t_{test}), dist(p_{test} \rightarrow p_j)))$;
end
return $p_{predict}$;
```

be useful, based on our assumption, we avoid possible noise from other information. The features we use here are extracted from: *start time of a visit*, *end time of a visit* and *current location*.

Features. Unix timestamp can be converted into human readable time according to the time zone. The features which we are interested in contain *day of week, hour of day, hour of week, weekend, weekday, morning, noon, afternoon, evening, midnight*. For *day of week, hour of day, hour of week*, we use 1-of-K encoding to get binary values for the features. And for *weekend, weekday, morning, noon, afternoon, evening, midnight*, they are themselves binary features corresponding to true or false. We decide whether it is *morning, noon, afternoon, evening, midnight* by dividing the day time into 5 timeslots. Each timeslot corresponds to one feature. The bit corresponding to the timeslot that contains the timestamp will be on. Current location is also used as a feature and uses 1-of-K encoding.

Up to now, the features we described do not capture any information about the sequence structure. But in fact, this information sometimes can be very useful. To illustrate, consider a person who moves to a new house during the process of data collection. After the point he moved, the places he visits will change, perhaps significantly. Therefore, we add the *normalized start_time* (in the range $[0, 1]$) as a feature. When testing, we normalize the *start_time* to the range $[0.8, 1]$.

For the classification models, we consider two sets of features. In the first set, only *day of week, hour of day, location, start_time* are used. In the second set, all features are used.

3.3 Algorithms

From above modeling we selected the following six algorithms for detailed experiments. *Period_all* and *Period_trusted* are based on the periodicity model with all transitions or only trusted transitions respectively. *SVM_allFeature_trusted*, *SVM_allFeature_all*, *SVM_original_trusted*, *SVM_original_all* are based on the multi-class classification model with different feature sets and different transition sets. Details will follow in the next section.

4. EXPERIMENTS

In this section, we first address some practical issues related to implementing the proposed algorithms and then give the experimental results of each model. Then we describe the final 5 runs of results from the models for submission.

Algorithm	Average Accuracy
SVM_allFeature_trusted	55.69%
SVM_allFeature_all	55.06%
SVM_original_trusted	54.36%
SVM_original_all	53.29%
Period_trusted	50.44%
Period_all	48.91%

Table 1: Avg. Accuracy on all Users

4.1 Practical Issues

Cross Validation. All the parameters are tuned via 5-fold cross validation. To guarantee fair comparison between different models, the data are partitioned before training and all the models share the same partition.

Dealing with Small Dataset. There are certain users (e.g., user 151) who have very few trusted transitions in the training data. For such users, we examine the data by hand and give predictions based on our observation. The intuition is still making prediction based on spatial temporal regularities.

Dealing with unseen places. We always predict the next place to be a place which the user has visited before. Based on our model, even if the current place is never seen in the training data, we can utilize the similarity in time to make prediction.

Trusted Transitions. Although the test data used are selected from the trusted transitions, the untrusted transitions contain many useful hints. Especially when we consider the periodicity of staying at certain places, the time information is valuable. We train two versions of each model. One uses only trusted transitions and the other one uses all data points in the training set. In the testing phase, only trusted transitions are used for evaluation.

4.2 Experimental Results

Experiment Setting. We conduct all the experiments in Matlab. We implement multi-class classification via SVM with RBF kernel using the matlab interface of LIBSVM[2].

Results. We show 6 algorithms in the first column of Table 1. The second column shows the average accuracy over all users for each algorithm. With respect to this average accuracy, the classification models generally perform better than periodicity based models. Still we want to consider the periodicity based model because they explicitly emphasize the temporal similarity. If a person is highly regular, this model will outperform the classification models. The models using all features generally perform better than the models using only the absolute time features. This can be expected because all the newly added features have some discriminative power intuitively; adding them will improve performances. We can also see the trusted version generally performs better than the untrusted version. This is because the untrusted version more likely to bring noise to the data. However, when we look into accuracy for each user, the best model varies. This allows us to do model combinations and generate better algorithms.

Discussions. The following analysis and figures are generated from the classification models, but the periodicity based models have similar results.

We first examine when the models make mistakes. Take user 8 as an example. As shown in Figure 4, we plot the accuracy of prediction for each place (i.e.accuracy when this place is to be predicted as the next place) in subfigure 3 and at each place, the accuracy of predictions made from that place in subfigure 4. It is clear that the majority classes (frequent places) are usually correctly predicted while the minority classes are nearly never predicted correctly. And if the user is at a frequent place, prediction of the next place will be less accurate.

We note that there are two case for the relatively infrequent places. In the first case, the place is only relatively infrequent but still has a good number of samples. In this case, prediction from this place is more accurate. In the second case, the place is very infrequent and only appears once or twice. The prediction from this place will be similar to random guess. Actually we have thought a lot on how to improve the prediction accuracy for the places which do not have enough personal data. It is straightforward to think about utilizing the correlations between locations using human mobility as proposed by Yu Zheng et al. in [11]. However, one key issue in their method is to find stay point clusters, which they denote as locations that all users share. Then based on all users' travel experience, they detect correlations between locations. In our task, the locations (i.e. placeIDs) are user-specific. Since we do not have the raw GPS data, it is impossible for us to do any location mapping among different users, thus it is impossible to utilize other users' information to benefit the prediction. Thus we did not further refine our algorithm to tackle this issue.

Now we consider user 143 which has an accuracy around 90%. In Figure 5, we see that user 143 has a very compacted histogram of places. Thus the user is highly regular which makes prediction easier.

4.3 Results for Challenge Submission

For the challenge, we submit five results to compare with the ground truth. We generate the five results in the following way.

First three results are generated from *SVM_allFeature_trusted*, *SVM_allFeature_all*, *Period_trusted*, respectively. The last two models borrow ideas from ensemble learning. For the fourth result, for each test data point, the prediction takes the majority vote of all 6 models while predicting the next location. For the fifth result, for each test data point, we use the model that has the highest accuracy for the specific user to predict the next location.

5. CONCLUSION

In this paper, we described several methods to address the problem of next place prediction. All of them are based on spatial temporal information extracted from the current context. Out of around 80 places, our model gives reasonable prediction accuracy to around 55%.

6. ACKNOWLEDGMENTS

The authors would like to thank Professor Jiawei Han, Professor Tom La Porta, Zhenhui Li, Lu-An Tang, Chi Wang and Hyungsul Kim for their discussions in this work. Research was sponsored by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053. The views

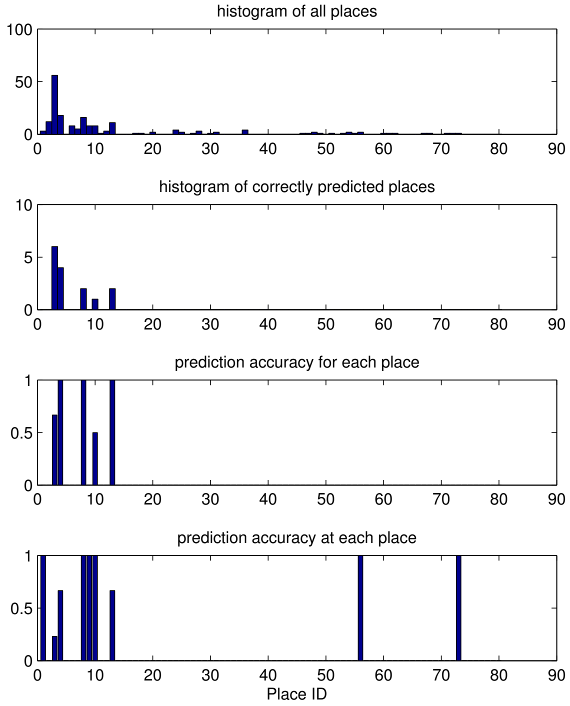

Figure 4: Error Analysis for User 8

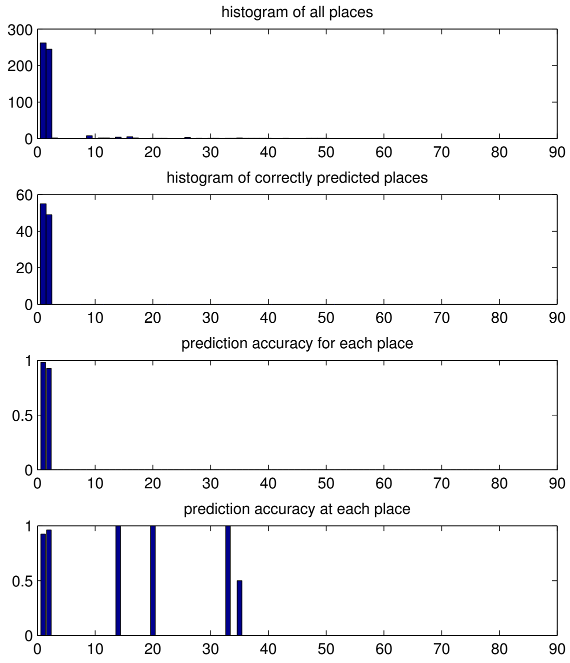

Figure 5: Error Analysis for User 143

and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, 1995.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[3] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, 2011.

[4] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *ICDE*, 2008.

[5] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen. The mobile data challenge: Big data for mobile computing research. In *Proceedings of Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf.. on Pervasive Computing*, Newcastle, June 2012.

[6] P.-R. Lei, T.-J. Shen, W.-C. Peng, and I.-J. Su. Exploring spatial-temporal trajectory model for location prediction. In *Mobile Data Management (1)*, 2011.

[7] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *KDD*, 2010.

[8] Z. Li, J. Wang, and J. Han. Period detection for sparse and incomplete event data. In *KDD*, 2012. to appear.

[9] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: A spatio-temporal prediction framework for pervasive systems. In *Pervasive*, 2011.

[10] 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.

[11] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining correlation between locations using human location history. In *GIS*, 2009.