Next Place Prediction by Understanding Mobility Patterns

Manoranjan Dash [#1], Kee Kiat Koo [#2], João Bártolo Gomes [#3], Shonali Priyadarsini Krishnaswamy [#4]
Daniel Rugeles [+5], Amy Shi-Nash [*6]

[#] *Institute for Infocomm Research, A*Star, 1 Fusionopolis Way, Singapore 138632*
[1,2,3,4] {dashm, koo-kk, bartologjp, spkrishna}@i2r.a-star.edu.sg

[+] *School of Compter Engineering, Nanyang Technological University, Singapore 639798*
[5] daniel007@ntu.edu.sg

[*] *DataSpark, Singapore Telecommunications Limited, Singapore 239732*
[6] amyshinash@singtel.com

***Abstract*—As technology to connect people across the world is advancing, there should be corresponding advancement in taking advantage of data that is generated out of such connection. To that end, next place prediction is an important problem for mobility data. In this paper we propose several models using dynamic Bayesian network (DBN). Idea behind development of these models come from typical daily mobility patterns a user have. Three features (location, day of the week (DoW), and time of the day (ToD)) and their combinations are used to develop these models. Knowing that not all models work well for all situations, we developed three combined models using least entropy, highest probability and ensemble. Extensive performance study is conducted to compare these models over two different mobility data sets: a CDR data and Nokia mobile data which is based on GPS. Results show that least entropy and highest probability DBNs perform the best.**

I. Introduction

Due to advancement in telecommunication and social network technologies, location-based services are playing a bigger role in people's lives. An important aspect of location-based services is to know the next place of a user. It helps in advertisement, product-packaging, urban planning, etc. In this paper we propose methods to predict next place of a user from CDR data, though, the same method could be applied to any mobile collected data. Our prediction takes into account some common knowledge about people's mobility behavior. A typical user follows some set patterns for some part of her mobility. For example, she goes from home to work on weekday morning, returns home from work in the evening, etc. But for the rest there may not be a set pattern. But, still there may be some common pattern. For example, a user goes to airport and returns home, goes to restaurant and returns home. How to capture these patterns? One way is to create different Dynamic Bayesian networks (DBN) [1] models to best capture the mobility behavior of a user. Some of these models should consider only location, some only time, while others should consider both location and time. When only location is used, we attempt to capture scenarios where the current location influences the next place, and time is not important. Similarly, for time-only model, time factor dominates. When both time and location are used, both time and location are important.

In the literature there are several works that use DBN to model classification tasks including speech recognition [2], [3], gesture recognition [4], medical prognostic [5], forecasting [6] among others. The specialty of this paper is next place prediction. We start observing people's mobility patterns, and then build several models to capture these patterns. By constructing different models we hope to improve the prediction accuracy of any single model. But how to select the best model?

We propose three different methods: entropy, highest probability and ensemble. Entropy is used in several recent works as a measure of mobility of a person, i.e. if the entropy is low, it means the next place is predictable with high accuracy [7]. In [8] entropy is used to understand how mobile a user is. It is not used to predict next place. In this paper we use entropy to choose the model that will possibly give the best prediction accuracy.

In most DBN models, highest probability is used to predict the next place. One may ask: what is the difference between entropy and highest probability? Entropy captures the entire distribution whereas highest probability captures only the most probable scenario. Why care for the entire distribution when we are actually interested in the highest probable prediction? Results show that performance of both methods is pretty close.

In literature there are several works where ensemble method is used [9]. Ensemble method is a populist approach where each model's prediction is taken into account and the most frequent one is chosen as the final prediction. In this paper, not only we use ensemble as a method to predict the next place by considering the outputs of all models, we try to make an entropy-ensemble joint method. We observed that entropy method is quite accurate when the selected model has low entropy, but it is not so accurate when the entropy is not-so-low. In these not-so-low entropy situations, we use ensemble method because, due to its populist approach, it is more suitable for such gamble-like situations.

An additional feature of this paper is next place prediction using CDR data. We first extract stay regions from raw CDR records of a user. Stay regions are places where a user spends significant amount of time. Sequence of stay regions of a user is extracted as spatio-temporal trajectory.

The specific contributions of this paper are:

- We build several dynamic Bayesian network (DBN)

978-1-4799-8425-1/15/$31.00 ©2015 IEEE

models considering users' mobility patterns. We also take into account a user's history.

- For each user and for each situation (location and time) we select the best model using three methods: entropy, highest probability and ensemble. Results show that each of these three methods is better than single DBNs.
- We test our method on two data sets: Nokia mobile data and CDR data of a telecommunication service provider. We used an algorithm to extract spatio-temporal sequence of stay regions for each user from her CDR records.
- For Nokia data, which is used for benchmarking purpose, experimental results show that entropy method gives the highest accuracy compared to all existing works.
- We find out factors affecting the prediction accuracy.

The structure of the rest of the paper is as follows. In the next section we discuss related work. Section III describes methodology. Experimental study is described in Section IV. The paper concludes in Section V.

II. Related Work

The related work on next place prediction can be reviewed considering two fundamental criteria, first the nature of the location data that captures user's mobility and second the methods and techniques proposed to build next place prediction models. Location prediction assumes that mobile sensor observations from wireless local network (Wi-Fi), Global Positioning System (GPS) or Global System for Mobile Communications (GSM) are available. The prediction task consists of using such data to model and understand the user's next location without knowing in advance the readings from future sensor data.

To better understand user mobility and visited locations, [10] proposed a general model for semantic trajectories, and introduced the concept of stops and moves. The locations of interest are the locations where the user stays for a period of time and the semantic trajectory represents the visiting history of semantic places (e.g., workplace, home, restaurant). Most existing research on next place prediction transforms the mobile sensor data into a sequence of visits/stays to semanatic places, similar to what is proposed in the semantic trajectory model [10].

Most related work creates a model based on frequent patterns and association rules from a history of user/collective trajectories as an ordered sequence of locations that are timestamped [11]. Other sequential learning models such as Hidden Markov Models [12], Conditional Random Fields [13] and Particle Filters [14] have been also applied to this problem. In the context of next place prediction the sequence of visits may have *Gaps* which represents a challenge for methods that required continuous sequences of a pre-defined length [12]. Here *Gaps* refers to significant time periods where the user location data is not collected (for example, when the mobile phone has run out of battery or there is no GPS/GSM network coverage). In addition, trajectories from check-in data of apps like Foursquare are usually sparse [15].

Recently the Nokia Mobile Data Challenge (MDC) released a large dataset for research and one of the dedicated tasks consisted of next place prediction [16]. From this MDC challenge, several approaches were able to predict the next place with high accuracy [17], [18]. The proposed approaches focused on learning a model for each user which captures the spatio-temporal trajectory of user visits. Significant effort was dedicated to feature engineering for each of the approaches with some reporting even millions of features [19]. However, this is possible because the rich contextual data available on smartphones nowadays, such as `accelerometer`, `bluetooth` and `call/sms` logs have been captured in the models.

A Dynamic Bayesian Network (DBN) model have been applied for Location prediction [20]. The data consisted of a handwritten log collected from 366 undergraduate students. In addition to their locations, the students also recorded their actions and routes. Using a large-scale data set of user check-ins from Foursquare [15] and generating temporal, user and general mobility features, authors compare different models and report that M5 model trees achieves noticeably high performance. Recently, in the context of indoor location prediction [21] proposed Indoor-ALPS. It uses three high level concepts to make predictions from historic location data: continuous feature selection (10 different spatio-temporal features), ensemble prediction, and incremental learning. It uses the Augsburg Indoor Location data set for evaluation.

III. Methodology

A natural and efficient method for next place prediction is Dynamic Bayesian network (DBN). DBN has been applied in different areas such as audiovisual, speech, gesture recognition [2], [3], [4], medical diagnosis [5], and stock price forecasting [6].

Here we describe a DBN with both location and time features. Note that time itself has two features: Day of Week (DoW) and Time of Day (ToD). DoW can take any of the seven days of a week, i.e., Monday, Tuesday, etc. or it can take any of two values, i.e. weekday and weekends. The choice depends on the kind of patterns we want the model to capture. For example, a typical user goes to work on weekdays and not on weekends. Here $\{$weekday, weekend$\}$ suffices. But consider the example where a user goes to a club on specific day(s) in a week. Here we should consider each day of a week separately. As far as ToD is concerned, it can take any value within the 24 hours in a day. But the model can benefit more by extracting time intervals according to mobility patterns. For example, a typical user goes to work from home on weekdays between (say) 7–9 am. Our analysis shows that best accuracy can be obtained when we group 24 hours in a day into three intervals. These are 11pm-6am, 6am-3pm, and 3pm-11pm. Given the current location $\mathcal{P}^i$ and time (dow^i, tod^i) the predicted next place is decided using Equation 1.

$$\hat{\mathcal{P}}^{i+1} = \underset{\mathcal{P}^{i+1}}{\operatorname{argmax}} P(\mathcal{P}^{i+1}|\mathcal{P}^i, dow^i, tod^i) \quad (1)$$

A DBN represents $P(\mathcal{P}^{i+1}|\mathcal{P}^i, dow^i, tod^i)$ in a compact way.

A. Other DBNs

Equation 1 takes into account all three features, i.e., location, dow, and tod. Practical observations over MDC and CDR data show that some mobility patterns can be captured using only one or two features.

1) Location Only: Using CDR data we conducted an experiment where we always predicted home. *Average accuracy over all users was 42%.* On closer observation we found that a typical user follows this kind of pattern: goes to airport and returns home, goes to shopping centre and returns home, goes to work and returns home, etc. Similarly, during working hours, one may go to a client's office and return to work place. To capture these kinds of patterns location alone suffices.

$$\hat{\mathcal{P}}^{i+1} = \underset{\mathcal{P}^{i+1}}{\operatorname{argmax}} P(\mathcal{P}^{i+1}|\mathcal{P}^{i}) \tag{2}$$

2) Time Only: There are some mobility patterns that depend only on time. Consider weekday evening time. A typical user returns home irrespective of where she may be. She may return from work to home, she may go to a restaurant, and then return home, she may go to a gym and return home. These activities are influenced more by time factor than location. As described earlier, time has two features, i.e., DoW and ToD. We can use one or both.

$$\hat{\mathcal{P}}^{i+1} = \underset{\mathcal{P}^{i+1}}{\operatorname{argmax}} P(\mathcal{P}^{i+1}|dow^{i}, tod^{i}) \tag{3}$$

Similar models can be constructed for DoW-only, ToD-only.

3) Previous Locations: Existing works have shown that previous locations also matter in next place prediction. How many previous locations should we consider? In this work we limit it to two because, in general, for more than two locations prediction accuracy does not increase. An example of a pattern with two previous locations: a user goes for a movie, eats in a restaurant and returns home. A mathematical definition with two previous locations is given below.

$$\hat{\mathcal{P}}^{i+1} = \underset{\mathcal{P}^{i+1}}{\operatorname{argmax}} P(\mathcal{P}^{i+1}|\mathcal{P}^{i-2:i}) \tag{4}$$

Similarly, we can model for one previous location. Here we consider location-only models. Arguably it is somewhat awkward to consider previous locations for time-only models.

B. Which Model to select?

At any situation in the test data, zero or more models can be applied. Which model to choose? Of course we should choose the model that gives the highest accuracy at that situation. In this section we discuss three methods for selecting the best possible model.

1) Lowest Entropy: In [8] it is shown that entropy can measure how mobile a user is. If a user is a typical home-work-home-type, who goes to his work place, and returns home, and on weekends goes to a fixed restaurant, and/or shopping mall, his entropy will be very low. On the other hand if a user roams around a lot, his entropy will be high, because it will be more uncertain to determine his next place. Consider any model $\mathcal{M}$ that we dsecribed earlier. At any situation, i.e., current and previous locations and current time, $\mathcal{M}$ will predict next place with certain probability. Given the random variable representing a possible next place $\mathcal{P}$ and denoting the sample space for $\mathcal{P}$ as $\mathcal{S}_{\mathcal{M}}$, we can find the entropy for model $\mathcal{M}$ as follows:

$$\mathcal{H}_{\mathcal{M}} = -\sum_{p \in \mathcal{S}_{\mathcal{M}}} (\mathcal{P} = p) * (log(\mathcal{P} = p)) \tag{5}$$

Out of all such models select the one with the least entropy.

$$\overset{*}{\mathcal{M}} = \underset{\mathcal{M}}{\operatorname{argmin}}\, \mathcal{H}_{\mathcal{M}} \tag{6}$$

For entropy to work properly we need a constraint. That is, each model should satisfy a minimum number of instances. Consider the example where a model is based on only one instance, i.e., at the given situation in the test data, only one instance satisfies from training data for a model $\mathcal{M}$. Then its entropy will be zero. Naturally this model will be selected over other models which are based on more number of instances but with higher entropy. So, each model must satisfy a Minimum Number Instances ($MinNumInst$) to be included in the entropy selection method.

2) Highest Probability: Consider a situation in test data. Let us say several models can be built for that situation using the training data. [1] Each of these models will have a distribution of possible next places. One of these possibilities will have the highest count, or probability. Using the highest probability criterion we choose that model having the maximum *highest probability*. Mathematical definition follows. For each model $\mathcal{M}$ we determine the highest probability.

$$\mathcal{G}_{\mathcal{M}} = \max_{p \in \mathcal{S}_{\mathcal{M}}} \frac{count(p)}{\sum_{p \in \mathcal{S}_{\mathcal{M}}} count(p)} \tag{7}$$

where $count(p)$ is the number of instances with next place p. Among all such models we select the one with the maximum *highest probability*.

$$\overset{*}{\mathcal{M}} = \underset{\mathcal{M}}{\operatorname{argmax}}\, \mathcal{G}_{\mathcal{M}} \tag{8}$$

3) Difference between entropy and highest probability: Let us assume that we have a total of six place labels. Let us say that one model found probability distribution of the next place to be [0.5, 0.1, 0.1, 0.1, 0.1, 0.1], while another model found the distribution of the next place to be [0.4, 0.4, 0.2, 0, 0, 0]. The first model has a clearer answer but due to higher entropy it is not selected. If we replace the entropy method with the model with the highest probability,

[1] Note that we do not have to build models on the fly. Using the historical data, these models can be built and stored. In a test situation, we do not re-compute these models, instead we select the ones that satisfy the given situation.

then the first model will be selected. A highest probability method at any situation will predict the place with the highest probability among all models. The basic difference between entropy and highest probability is that entropy considers the entire distribution of all possible next place choices whereas highest probability considers only the choice with the highest probability. Experimental results show that accuracy of entropy and highest probability are close.

4) Ensemble Method: Finally, we also considered ensemble method. It is a natural choice when there are several models to choose from. Each model $\mathcal{M}$ makes a prediction from the possible next places $p \in \mathcal{S}_{\mathcal{M}}$, and the place p with the highest count is the final prediction.

C. Extracting Stay Regions from Call Detail Records (CDR) Data

CDR data typically includes records of phone events (phone calls and SMS) made from a mobile phone connected to a network. Each record consists of attributes such as anonymized id of the user, timestamp, lat-lon of cell tower among others. In order to use CDR data for next place prediction we need to find out places where a user stays for sufficient length of time. A stay region r_j^l of a user u_l is defined by a spatial and temporal threshold. A spatial threshold $SpaThresh$ helps to cluster the cell towers in close proximity of each other. A temporal threshold $TimeThresh$ discriminates events within a stay region from those on transit. $r_j^l.LatLon$ denotes the center point of a stay region. Any tower within a distance of $SpaThresh$ from $r_j^l.LatLon$ belongs to the stay region. A user spends at least $TimeThresh$ amount of time in a stay region. Note that a stay region can have more than one cell tower.

The task of determining stay regions can be stated as follows: Given CDR records d_h^l, $h = 1...nd^l$ for a user u_l, determine all stay regions r_j^l such that both thresholds are met. Note that consecutive records with the same cell tower are replaced by their first and last records.

Algorithm 1 Algorithm to Determine Stay Regions

```
1:  procedure DETERMINESTAYREGIONS(CDRrecords)
2:      for each user u_l, l = 1...n do
3:          j = 1; h = 1;
4:          repeat
5:              Get first record d_h^l;
6:              h++; distance = 0; time = 0;
7:              r_j^l.LatLon = d_h^l.LatLon
8:              while distance < SpaThresh do
9:                  Get next record d_h^l;
10:                 r_j^l.LatLon = Centroid(r_j^l, d_h^l);
11:                 time+ = d_h^l.timestamp;
12:             if time > timeThresh then
13:                 Store r_j^l as a stay region;
14:                 j++;
15:         until eof for user u_l
```

Centroid method employs a weighted average technique to compute the new centroid of a stay region. For example, if three cell towers are already included in a stay region, for a fourth cell tower *Centroid* method assigns a weight of 3 to the current centroid $r_j^l.LatLon$ and a weight of 1 to the fourth cell tower lat-lon ($d_h^l.LatLon$).

After determining stay regions of a user, we create a spatio-temporal trajectory by arranging the stay regions and time duration (start and end times) in chronological order. This data is used for developing next place prediction models.

IV. Performance Study

This section begins with descriptions about two data sets that we use to compare different models. Following this, in Section IV-B we compare the next place prediction accuracy of different DBN models.

A. Two Data Sets

We use two data sets, i.e., Nokia data challenge [22] and a CDR data set from Singapore's largest telecom company.

1) Mobile Data Collection Challenge Dataset (MDC) by Nokia [22]: Mobile Data was collected from 80 persons during a period of time varying from a few weeks to two years (2009 to 2011). GPS data has been processed to generate a trajectory of visits or stays with *(user id, start time, end time, and stayID)*. Each stayID corresponds to one out of eleven possible place labels.

2) CDR Data Set: In Section III-C an algorithm was introduced to extract stay regions. By doing so, CDR data can be used for next place prediction. Before extracting stay regions CDR data was cleaned using a cell tower oscillation removal algorithm [23]. For each user a trajectory of stays is generated with *(user id, start time, end time, and stayID)*. Note that a stayID in the CDR data also has a lat-lon. A difference between stay region trajectories generated from GPS data (eg. Nokia data) and that by CDR data is that GPS data will not have any gaps whereas CDR data usually has gaps. If CDR data has location updates, these gaps may be reduced. See http://en.wikipedia.org/wiki/Mobility_management#Location_update_procedure to learn more about location updates.

B. Dynamic Bayesian Network Model

In this section we make a comparison among the DBN models. We use 0, 1 or 2 previous places as history. For 0-history (0-H), there are seven models: Loc-Only, ToD-Only, DoW-Only, Loc-ToD, Loc-DoW, ToD-DoW, and Loc-ToD-DoW. For 1-H and 2-H, time-only models do not make sense. So, 1-H and 2-H have four models each: Loc-Only, Loc-ToD, Loc-DoW, and Loc-ToD-DoW. We also compare 'least entropy', 'highest probability' and ensemble methods. So, in total we compare (7+4+4+3) 18 models.

We perform 80-20 training-testing validation (first 80% training and last 20% test) and 10-fold cross validation (CV) . Both approaches have their advantages. The 80-20 approach is stricter than 10-fold CV in the sense that only past data can be used to learn about future events (next place). But we feel 10-fold CV is not so unreasonable particularly when there is no significant drift in mobility patterns of a user, i.e., it does not matter if we learn from the past to predict future or vice-versa.

Table I presents the results. Note that 1-H combined models have 11 methods compared to 7 for 0-H. The reason is for 1-H

TABLE I: Comparison among DBN models

Model	Accuracy (CDR)		Accuracy (Nokia)	
	80-20	10-Fold CV	80-20	10-Fold CV
0-H				
Loc-DoW-ToD	**54.70**	**56.79**	**60.99**	**62.12**
Loc-ToD	53.42	55.42	58.82	59.58
Loc-DoW	51.22	52.58	58.99	58.99
Loc-Only	51.38	52.36	57.60	57.45
DoW-Only	47.28	48.10	53.11	51.92
ToD-Only	51.47	52.84	57.23	56.30
DoW-ToD	53.82	55.35	60.07	58.37
1-H				
Loc-DoW-ToD	47.19	45.89	58.68	57.81
Loc-ToD	47.80	47.06	57.91	56.92
Loc-DoW	45.94	45.41	56.50	55.86
Loc-Only	46.79	46.75	55.70	54.56
2-H				
Loc-DoW-ToD	**35.82**	**35.04**	**52.74**	51.62
Loc-ToD	38.11	47.06	54.46	52.17
Loc-DoW	36.96	36.86	53.15	51.53
Loc-Only	38.69	37.96	53.18	**50.78**
Least Entropy				
0-H (7 Models)	56.42	58.23	61.68	61.91
1-H (11 Models)	56.28	57.90	61.87	**62.46**
2-H (15 Models)	55.78	57.42	62.23	62.18
Highest Probability				
0-H (7 Models)	56.9	**58.68**	61.92	62.07
1-H (11 Models)	**56.97**	58.62	**62.38**	62.45
2-H (15 Models)	56.71	58.36	62.20	62.35
Ensemble				
0-H (7 Models)	55.68	57.13	61.67	60.64
1-H (11 Models)	56.09	57.51	61.70	61.63
2-H (15 Models)	55.99	57.42	62.08	61.63

we consider both 0-H (7 models) and 1-H (4 models). Similarly for 2-H there are 15 models.

a) Analysis of Results: Results show that 'highest probability' and 'least entropy' are the best. 2-H individual models performs the worst. Among individual DBN models, Loc-DoW-ToD performs the best. Except for 10-fold CV for Nokia data, no individual model performs better than any combined models (i.e., least entropy, highest probability and ensemble). Also, performance of combined models are very close, i.e., within 1-2 percentage points.

We want to clarify the drop in accuracy for '0-H least entropy' and '1-H least entropy.' It may seem strange that '0-H least entropy', which is based on lesser number of models (7 models) than '1-H least entropy' (11 models), is having better accuracy. The reason is a model selected from more number of models may have wrong prediction for a test case than a model selected from lesser number of models.

Among the individual models for 0-H, Loc-DoW-ToD performs the best whereas its performance is the worst for 2-H. The reason is 2-H takes into account previous two places whereas 0-H does not need it. This leads to very low number of instances satisfying a particular test case for 2-H. So much so that when there is zero instance, the system considers the prediction to be *wrong* thus leading to lower accuracy for 2-H than 0-H. This explanation holds good for the rest of the individual models as well.

In the introduction we briefly discussed about the benefits of combining entropy with ensemble. When entropy of all DBN models is not-so-low, most likely the prediction will be wrong. In such a scenario we want to replace entropy by ensemble. Note that it may be beneficial to use a populist method like ensemble. We performed an experiment on Nokia data using 80-20 validation approach. We used 'least entropy' method for 0-H (8 models). Least entropy method gives an accuracy of 61.68. Then we replaced the entropy method by ensemble method for the scenario where entropy of each individual DBN model is greater than 0.5. The accuracy increased slightly to 61.91.

C. Additional Results for Nokia Data: Benchmarking

For 11 labels, the best prediction accuracy we achieved is 62.79% using entropy method without considering previous locations. In literature many papers report lower accuracy [24], [9], [25].

For 10 labels, i.e., without 'others' label, the best prediction accuracy we achieved is 75.72% by least entropy method without considering previous locations. In literature the best accuracy reported is 73.26 [9]. So, our proposed method gives the highest accuracy. It will be interesting to find out why the entropy method performs significantly better for 10 labels compared to 11 labels.

D. Factors Influencing Next Place Prediction Accuracy

For each user we considered four factors: number of distinct stay regions (v1), number of visits or stays (v2), number of distinct cell towers (v3), and number of records (v4). We used the entire user base of approximately 3.9 million users in order to get robust trends. Result show that only number of distinct stay regions has noticeable influence on next place prediction accuracy. We computed correlation between accuracy and these four variables. The results are: -0.295, -0.037, -0.17, and -0.003 for the four variables respectively. So, the most important factor is number of distinct stay regions, and the second most influential factor is number of distinct cell towers. Both factors are somehow related, i.e., a user with high number of distinct cell towers usually has high number of distinct stay regions. Figure 1 plots the relationship between accuracy and number of distinct stay regions. As the number of stay regions increases, accuracy drops. One explanation is that with more number of stay regions, the models will be confused about the next place.

Average number of distinct stay regions is 2.73 for users with prediction accuracy in the range 95–100% and 5.32 for 0–5%. So, on an average, users with very high accuracy has smaller number of distinct stay regions than users with very low accuracy. We also considered another interesting experiment. We hypothesized that users with high accuracy have a more predictable mobility pattern, such as home-work-home kind, but users with low accuracy have a less predictable mobility pattern, such as moving between different stay regions without any dominating pattern. For example, delivery boys, taxi drivers, etc. To quantify this kind of pattern, we used some statistics. For each user we counted the number of visits for each stay region. Standard deviation is computed. For each

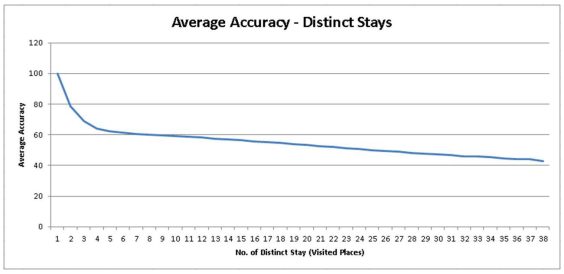

Fig. 1: Average Prediction Accuracy (0H Highest Probability) vs. No. of Distinct Stays (Visited Places)

group average standard deviation over all users is calculated. The idea is that a user who is very mobile, will have lower standard deviation than a user who follows a typical home-work-home pattern. A user who follows home-work-home pattern has most of his visit to home and work, and very few to other stay regions. Thus her standard deviation will be higher. Average standard deviation is 26.92 for the 95–100% group, but only 4.67 for the 0–5% group.

So, in summary the two most important factors influencing the next place prediction accuracy are: number of distinct stay regions and standard deviation of number of visits to different stay regions. It will be interesting to find out of the two which is more dominant. We would like to consider entropy as a factor in our future study.

V. Conclusion

We have proposed a set of DBN models considering different combinations of features, i.e., location, DoW, and ToD. Then, we proposed three methods to select the best DBN for a given user and for a given situation. Performance study shows that the three selection methods perform better than individual DBNs. In general, accuracy for 0H DBN that does not take into account previous location(s) is better than 1-H DBN and 2-H DBN. This is caused by the sparse conditional distributions that result after training (1,2)-H DBN. The results of least entropy and highest probability methods for Nokia data are better than existing methods in the literature. The two most influential factors for prediction accuracy are number of distinct stay regions and uniformity (or entropy) of mobility.

References

[1] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'98, 1998, pp. 139–147.

[2] I. Ravyse, D. Jiang, X. Jiang, G. Lv, Y. Hou, H. Sahli, and R. Zhao, "Dbn based models for audio-visual speech analysis and recognition," in *Proceedings of the 7th Pacific Rim Conference on Advances in Multimedia Information Processing*, 2006, pp. 19–30.

[3] G. Zweig and S. Russell, "Speech recognition with dynamic bayesian networks," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, 1998, pp. 173–180.

[4] J. Rett and J. Dias, "Gesture recognition using a marionette model and dynamic bayesian networks (dbns)," in *Proceedings of the Third International Conference on Image Analysis and Recognition - Volume Part II*, ser. ICIAR'06, 2006, pp. 69–80.

[5] M. A. J. van Gerven, B. G. Taal, and P. J. F. Lucas, "Dynamic bayesian networks as prognostic models for clinical patient management," *J. of Biomedical Informatics*, vol. 41, no. 4, pp. 515–529, 2008.

[6] Y. Xiang, J. Smith, and J. Kroes, "Multiagent bayesian forecasting of structural time-invariant dynamic systems with graphical models," *Int. J. Approx. Reasoning*, vol. 52, no. 7, pp. 960–977, 2011.

[7] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, "Approaching the Limit of Predictability in Human Mobility," *Nature*, vol. 3, 2013.

[8] C. Song, Z. Qu, N. Blumm, and A.-L. Barabsi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[9] R. Montoliu, A. Martnez-Uso, and J. Martnez-Sotoca, "Semantic place prediction by combining smart binary classiers," in *Mobile Data Challenge (by Nokia)*, 2012.

[10] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. De Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data & knowledge engineering*, vol. 65, no. 1, pp. 126–146, 2008.

[11] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext: a location predictor on trajectory pattern mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 637–646.

[12] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden markov models," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 911–918.

[13] R. Pan, J. Zhao, V. W. Zheng, J. J. Pan, D. Shen, S. J. Pan, and Q. Yang, "Domain-constrained semi-supervised mining of tracking models in sensor networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 1023–1027.

[14] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *Pervasive Computing, IEEE*, vol. 2, no. 3, pp. 24–33, 2003.

[15] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Proceedings of the 2012 IEEE 12th International Conference on Data mining*, 2012, pp. 1038–1043.

[16] J. 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, in conjunction with Int. Conf. on Pervasive Computing, Newcastle, UK*, 2012.

[17] V. Etter, M. Kafsi, and E. Kazemi, "Been there, done that: What your mobility traces reveal about your behavior," in *Mobile Data Challenge (by Nokia)*, 2012.

[18] J. Wang and B. Prabhala, "Periodicity based next place prediction," in *Mobile Data Challenge (by Nokia)*, 2012.

[19] Y. Zhu, E. Zhong, Z. Lu, and Q. Yang, "Feature engineering for semantic place prediction," *Pervasive Mob. Comput.*, vol. 9, no. 6, pp. 772–783, 2013.

[20] S. Lee, K. C. Lee, and H. Cho, "A dynamic bayesian network approach to location prediction in ubiquitous computing environments," in *Advances in Information Technology*. Springer, 2010, pp. 73–82.

[21] C. Koehler, N. Banovic, I. Oakley, J. Mankoff, and A. K. Dey, "Indoor-alps: an adaptive indoor location prediction system," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 171–181.

[22] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-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)*, 2012.

[23] W. Wu and et al., "Oscillation resolution for mobile phone cellular tower data to enable mobility modelling," in *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, 2014, pp. 321–328.

[24] T. M. T. Do and D. Gatica-Perez, "Contextual conditional models for smartphone-based human mobility prediction," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 163–172.

[25] V. Etter, M. Kafsi, and E. Kazemi, "Been there, done that: What your mobility traces reveal about your behavior." in *Mobile Data Challenge (by Nokia)*, 2012.