Collective Suffix Tree-based Models For Location Prediction

Muawya Habib Sarnoub Eldaw
Department of Computer Science
and Information Systems
Birkbeck, University of London
London, WC1E 7HX, UK
eldaw@dcs.bbk.ac.uk

Mark Levene
Department of Computer Science
and Information Systems
Birkbeck, University of London
London, WC1E 7HX, UK
mark@dcs.bbk.ac.uk

George Roussos
Department of Computer Science
and Information Systems
Birkbeck, University of London
London, WC1E 7HX, UK
g.roussos@dcs.bbk.ac.uk

UbiComp '13 Adjunct, Sept 8-12, 2013, Zurich, Switzerland.
ACM 978-1-4503-2139-6/13/09...$15.00.

Abstract
Models developed for the prediction of location, where a specific individual will be present at a future time, are typically implemented using a one-model-per-user approach which cannot be employed for inferring collective or social behaviours involving other individuals. In this paper, we propose an alternative that allows for inference though a collaborative mechanism which does not require the profiling of individual users. This alternative utilises a suffix tree as its core underlying data structure, where predictions are computed over an aggregate record of behaviours of all users. We evaluate the performance of our model on the Nokia Mobile Data Collection Campaign data set and find that the collective approach performs well compared to individual user models. We also find that the commonly used Hit and Miss score on its own does not provide sufficient indication of prediction accuracy, and that employing additional metrics using the mean error may be preferable.

Author Keywords
Suffix tree, Markov model, collective model, mobility trails, location prediction.

ACM Classification Keywords
H.1.1 [Systems and Information Theory]: Information Theory.

Introduction

The ability to predict the behaviour of individual mobile users is a key ingredient for context-aware adaptation in mobile applications and systems. One approach to build a mobile recommendation system is to employ the one-model-per-user paradigm to infer immediately subsequent locations [6]. However, by setting the goal "to build a user-specific model that learns from his/her mobility history, and then apply the model to predict where the user will go next," one presents a tightly defined objective: this model must be constructed solely from personal data recorded specifically as the result of the behaviour of the particular individual observed.

This approach for building user models clearly offers distinct advantages. Indeed, such a model can potentially be developed following a variety of machine-learning and data-mining techniques and a full implementation on the user mobile device is feasible. The latter feature would also allow the specification and application of tight controls on access to the model by external systems and third-party applications. Such security constraints would inevitably limit opportunities to capitalise on the facilities afforded by this model, since adaptation, context-awareness and recommendations would be restricted to specific and specifically authorised parties defined by cumbersome to manage and adapt use policies. Thus, this one-model-per-user approach would be limited due to being over complicated and due to the need for decentralised data management to preserve robustness and reliability. Conversely, if a centralised system is used, then it would pose a considerable privacy liability, which places significant challenges for a widespread deployment. A trade-off between the two approaches is possible but recent experience with web- and location-based systems indicates that decentralised systems quickly become difficult to operate effectively in practice. Setting aside privacy-balancing issues, the one-model-per-user approach has several further limitations:

- In practice, most such models have poor performance when novel behaviours occur.
- Focusing on the individual, such models cannot be employed for inferring collective/social dynamics such as those resulting from cascading behaviour within social networks.

An alternative modus operandi to the one-model-per-user, is to infer individual behaviour by employing a single aggregate model incorporating the collective record of observed behaviours. Moreover, this alternative also mitigates the privacy concerns because individual models identifying a specific user are never constructed nor stored.

This paper makes the following contributions, backed up by empirical evidence:

- It presents a collective prediction approach and investigates its predictive limits by comparing it to the one-model-per-user approach. To this end, we study the problem of predicting the location that the user will be visiting next, taking into consideration the time and location information of where the user had been in the past.

- It examines the effect of the length of the user record of the most recent temporal locality used to make inferences, and the relative loss of accuracy when reduced data samples are provided so as to establish the exact trade-off involved.
- It describes a more comprehensive approach, than previously proposed, to evaluating the mobility model's prediction accuracy, using the *Mean Absolute Error* (MAE) and *Root Mean Squared Error* (RMSE) metrics. It also investigates the merits of the *Hit and Miss Score* (HM) in evaluating the accuracy of location prediction algorithms.

Note that we have followed a specific methodology to construct the *collective model*. In brief, we construct a variable length Markov chain model [4], which we stored in an augmented suffix tree data structure that allows for flexible and high-performance querying. This representation is supplemented by specific weighted spatio-temporal metrics of significance to estimate and rank the probability of computed inferences.

The coming sections in this paper include the Related work, Suffix tree models, Temporal models, Data set and experimental setup, Experimental results and Conclusion and future work.

Related Work

To decipher the complexity of human mobility, many approaches have been proposed for building models that can accurately predict individuals' future locations of visit. Generally, these approaches can be divided into three major categories based on the perspective from which the data is being considered: spatial, temporal, and joint spatio-temporal approaches.

Researchers have investigated the user's spatial patterns on mobile data and various prediction approaches have been proposed [12]. Other methods rely on the user's temporal patterns, as shown in [1]. In [11], Scellato et al. proposed a spatio-temporal framework, called NextPlace, which used nonlinear time series analysis of users' arrival time and residence time to predict temporal behaviour.

Prediction models of future locations of visit have been predominantly implemented using a one-model-per-user approach. For example, Krumm [9], used a Makov model for making short-term route predictions for vehicle drivers. [2] however, suggested a model in which locations are incorporated into a Markov model that can be consulted for use with a variety of applications in both single-user and collaborative scenario where multiple single-user models can be shared. Unfortunately, it is not clear how they evaluated their models apart from showing that the predictions for their single user model were compared against "random chance". Also they did not address the situations when the user has no mobility history to use for prediction. Moreover, sharing multiple single-user models inevitably raises concerns relating to the privacy of users' information being compromised, for example, by the service provider. Unlike [2] and [9], this paper, presents a collective suffix tree model (i.e. a single model for multiple of users) which is a principled and scalable implementation of a variable length markov model. It also presents various models that are capable of dealing with situations when the user has no mobility history to use for inferring future locations.

The "Hit and Miss Score (HM)" which, also known as the "Success Ratio", has been widely used to measure

the prediction accuracy in domains such as web usage mining [5] and recently in mobility prediction [6]. In the context of predicting the user's mobility, our opinion is that HM, on its own, does not give a sufficient assessment for the prediction accuracy. This is strongly supported by our tests' results. We propose that, in addition to the HM Score, MAE and RMSE should be used.

Suffix Tree Model

Overview

The cornerstone of our approach is the use of *mobility trail* which we define as the sequence of recordings, of the temporal and spatial information, of all the visits that the user makes in a day. A traditional drawback of trail analysis is that it requires considerable storage and computational resources to discover hidden patterns. To overcome this, we employ a trail-based analysis approach, which utilises suffix trees as the data structure for efficient storage, filtering and retrieval [3].

Mobility Trails

We view a user's mobility history as a directed graph, where vertices denote locations which the user visited and edges denote paths between these locations. Two locations are said to be connected if they have been visited in sequence by the observed user. In such context, a trail can be defined as a sequence of connected locations, such that the connections between locations are always directed.

Detecting the User's Mobility Patterns

We utilise the sequence of GPS recordings of the user's exact location to detect the hidden mobility patterns. To discover these patterns we apply the following procedure:

1. For every user, apply the DBScan algorithm to cluster the GPS data to identify the locations which are likely to be part of a daily pattern (Locations with number of visits below a certain threshold are ignored).
2. Compute the *centre of mass* of the locations in each discovered cluster.
3. Using a latitude/longitude grid, for each cluster, identify the grid cell(s) containing the centre of mass of the clusters.
4. Divide the day into equal time-units (We choose 20 minutes as the basic time unit).
5. Compute the duration of visit for each cluster using the time of visit associated with the GPS readings. Then identify the time-unit(s) corresponding to each visit.
6. Construct the daily trail using the grid cells and the time-units computed in step 5.

After applying above procedure we obtain sequences of visit-objects where each object contains the following information:

$<$ ***userID, time, day, location, meta-data***$>$

Suffix Trees

To efficiently store the trails and their related meta-data we use a probabilistic suffix tree data structure [3] enhanced with meta-data needed to encapsulate different information and metrics. Suffix trees can maintain all captured information in a compact format, while being able to respond to queries in linear time of the size of the trail. They have been successfully employed in a number of domains such as anti-spam filtering [10] and computational biology [3].

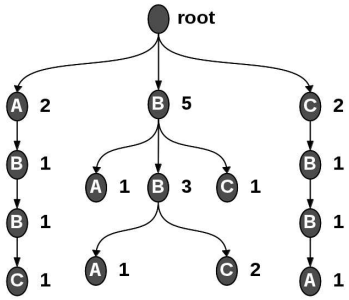

Figure 1: A suffix tree for the trails represented by the strings ”ABBC”, ”CBBA” and ”BBC”. The letters denote the individual visits in each trail and the numbers show the frequency of visit to each location. Suppose the tree represents the mobility history of a user u and the trail ”BB” gives the sequence of the his most recent visits. To predict the location that u will visit next, we present the trail ”BB” to the tree which produces the candidate locations ’A’ and ’C’ (’C’ has a higher probability as opposed to ’A’, hence most likely to be the next location of visit).

Tree Representation

For our suffix tree representation, we opted for a design in which the nodes are labelled as opposed to the edges [10]. Also, due to the nature of the task we are undertaking, our suffix tree uses a terminal character to determine the depth of the tree which, for a big number of users, can grow very large in size. Furthermore, our trees are not limited in size which is a key factor that gives the model the ability to learn as more locations are being explored by the observed users. An example of this data structure is shown in Figure 1.

The One-model-per-user

The one-per-user model is based entirely on a single user’s past mobility data which is organised into trails representing the daily sequences of visits. For each day sequence of visits, the time and location information of each visit is encoded in a string object and the objects, in turn, are grouped together to form a trail. The trails are then used to build the suffix tree. To make a prediction, let S be the suffix tree and, $\mathrm{T} = t_1 , t_2 , \ldots , t_n$, be a search-trails where t_i , $1 \leq i \leq n$, represents the locations visited by the observed user, and we wish to predict the next location in the sequence, i.e. t_{n+1} . Then we can apply *Algorithm 1*, called Predict, based on [7], to make a prediction as is shown in Figure 2.

The Collective Model

The collective suffix tree model is a joint model (i.e. a single tree) over the population of all users. It enables predicting the next location based on the past mobility data of multiple users. In this model, the data identifying specific users is completely anonymised before building the model.

Temporal Models

The motivation behind the models in this section is to address the lack of matching historical behaviour which the suffix tree requires when predicting future behaviours. These models predict the user’s future behaviour using only the current temporal context.

Time-spent Predictor (TSP)

The Time-spent Predictor (TSP) looks for the location where the user spent most of his/her time. It looks at the time-spent at each visited location and uses this as a basis for predicting the next location.If t prediction time interval, then (*TSP*) computes a time-spent ranked list of all the locations visited during t and returns the *top-k* locations. The collective version of TSP (*CTSP*) is a model built over a group of users as opposed to a single one.

Most Popular Location (MPL)

The Most Popular Location (MPL) looks for the location which the user visited most often. Given a time interval t, the (*MPL*) method ranks each user’s visited locations based on their frequency of visit. By restricting the prediction to t, MPL learns from the user temporal behaviour. As in *CTSP*, *CMPL* is the collective MPL version.

Data Set and Experimental Setup

Data Set

We tested our approach on the data set which Nokia released for it’s mobile data competition in 2012 [8]. We used only a subset of the data set (originally used for the Open Challenge task). This subset consisted of data collected from the mobile phones of 38 users and had the actual raw location data including GPS recordings and WLAN data for all the users. It was rich

Algorithm 1 *Predict(T, S, k)*

```
Begin
    let s be the longest
    suffix of T in S
    if s is empty then
        return the top-k
        popular locations;
        % these will be
        % children of the root
    else
        return a ranked
        list of the k most
        popular locations directly
        reachable from s;
        % these will be children
        % of the last location
        % in s, where s is a path
        % from the root
    end if
End
```

Figure 2: *Algorithm 1 - Predict(...)*

Property	Open challenge data set
Number of users	38
Number of user-days	8154
Avg number of locations per user	89

Table 1: Properties of the Nokia MDC Open Challenge data set.

and ideal for testing the models proposed in this paper particularly the comparison between the collective intelligence and the one-model-per-user approaches. A summary of the properties of the data is in Table 1.

Experiments

To test the proposed models, we organise the data into a sequence of days based on the time in which the user visited each location. We then implement the following steps:

1. For each user, we divide the daily trails into two sets: a training set containing the first α % of the total number of trails, and a test set containing the remaining *(100-*α*)%* (our choice of $alpha$ was 90).

2. For the one-per-user-model, we create a suffix tree for each user using the data given in the training set. For the collective model, we use the data from the union of the different users' training sets to create a single suffix tree.

3. For each daily trail data from the test set, we compute search-trails, of a maximum length n, using the following sliding window technique:

   Let T be the daily trail to be tested and assume a window of size n, we extract the n first locations from the trail T, match against the suffix tree and try to predict the $(n+1)^{th}$ location in T. We then slide the window to include the locations from the 2^{nd} to $(n+1)^{th}$ and attempt to predict the $(n+2)^{th}$ location and so on. The locations contained in the sliding window make, what we call, a search-trail (the maximum length we used was 5).

Performance Measurement

Error Measurement

To evaluate the accuracy of the proposed models, we use two well known metrics: the *Mean Absolute Error* (MAE) and the *Root Mean Square Error* (RMSE) [5], which are standard methods for measuring the average inaccuracy associated with a set of model-produced predictions. In the context of next location prediction, we have slightly different interpretation to MAE and RMSE as opposed to how they are normally interpreted in other contexts. MAE can be interpreted as the mean rank of the correct results while RMSE can be viewed as the square root of the mean of the square ranks of the correct results. By rounding up MAE or RMSE value to nearest whole number, either MAE or RMSE can be used to determine the size of the list of predicted locations which includes, on average, the correct next location. For example, if the MAE value 1.5, one can safely suggest the top 2 predicted locations as they are most likely to include, on average, the correct next location.

In our proposed approach, the possible next locations are ranked 1 to r according to their probability of visit. Assuming that the highest ranked location was the one followed (noting that ties are broken arbitrarily), then we compute the absolute error score for an individual prediction as *(r-1)*. For *n* predictions the MAE, as shown in Equation 1, is the average of the *n* individual error scores:

$$MAE = n^{-1}\sum_{i} r_i - 1. \quad (1)$$

As a special case, if the location that was followed had probability zero in the suffix tree, i.e. it does not appear as a next location, then we take it to be in the last position, *r*. In such a situation, we assume that the list

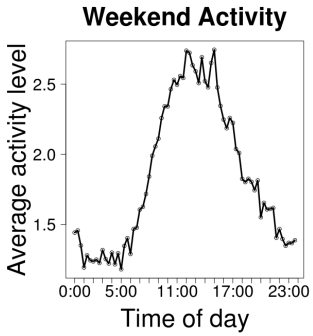

Figure 3: *Weekend Activity*

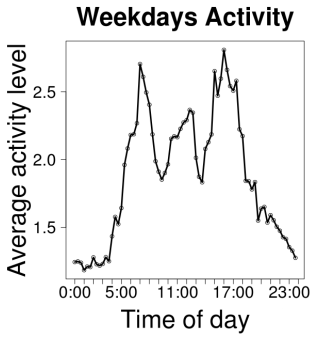

Figure 4: *Weekdays Activity*

of suggested locations has the length equal to the maximum branching factor of the suffix tree (i.e. the maximum number of leaves per branch).

To compute the RMSE, the errors are squared before they are averaged. If the squared error score for an individual prediction is *(r-1)*^{2} then for n predictions, the RMSE, as shown in Equation 2, is the square root of the average of the n squared error scores. Therefore, RMSE gives a relatively high weight to large errors, and is thus most useful when large errors are particularly undesirable. This implies that, in general, extending the list of suggested locations on the basis of the RMSE value, is likely to give a better chance for the correct next location to be included in the list, as opposed to when MAE is used.

$$RMSE = \left[n^{-1} \sum_{i} (r_i - 1)^2 \right]^{-\frac{1}{2}}. \quad (2)$$

Hit and Miss Score (HM)
We view a correct prediction as a (*hits*) and an incorrect one as a (*miss*). To compute the HM score, we divide the number of hits (*h*) by the total number of attempts made (*n*), as shown in Equation 3. In the current context, HM can be interpreted as the probability of guessing that the next location visited by the user was the one with the maximum probability.

$$HM = n^{-1}h. \quad (3)$$

Experimental Results

To develop our proposed approach, we benefited from an exploratory investigation in which we discovered a significant variance in the average number and the places visited by the users during the weekend period compared to that of the weekdays period, as shown in Figure 3 and 4. This motivated us to build separate models for each class of data.

Discussion
To benchmark for the performances of the proposed models, we compare against the performance of MPL. The idea is that for a good prediction performance, MAE and RMSE should be significantly smaller, whilst the HM score should be greater, compared to the MPL results. If suffix tree results are significantly close to the MPL results, the system should make the predictions on the basis of the most popular visited locations, and ignore the computationally expensive suffix tree models.

Target Locations with Visiting History:
Table 2, shows the results for target locations with history visits. The best HM score was produced by the collective suffix tree (CST_{seq}) model in which the sequential order of the visits was considered but the time of visit was ignored. Ignoring the time of visit increases the number of overlaps between the locations and, hence there is a greater chance for the correct locations to be predicted. The most exciting result in this experiment was the one achieved by *CST*, specially when compared to the results of its rival *ST*. The (*CST*) had a slightly higher RMSE and MAE compared to *ST*, which had the best MAE and RMSE results in the experiment. A key contributing cause to this difference in the MAE and RMSE results was the fact that (*CST*) utilises data from multiple users, and hence, has a higher branching factor.

Model	MAE	RMSE	HM
MPL	7.8425	13.4803	0.6189
TSP	10.0476	14.9909	0.5000
ST	**1.1099**	**2.5131**	**0.7596**
ST_{seq}	**7.0333**	**16.3389**	**0.7077**
CMPL	231.6909	252.2456	0.1225
CTSP	247.9568	260.5844	0.0717
CST	**1.6975**	**3.8453**	**0.7547**
CST_{seq}	**7.6994**	**19.2091**	**0.7733**

Table 2: MAE, RMSE and HM (In this experiment, all target locations given in the test set have visiting history in the training data set).

Record length	number of trails	%
0	555	25.63
1	33	1.52
2	29	1.34
3	25	1.15
> 3	1523	70.34

Table 3: Details of historical records (search-trails) used for querying the models

Target Locations with No Visiting History:
The results shown in Table 4, were the product of the TSP and MPL temporal models which, unlike the suffix tree models, they do not require the most recent history data to make genuine predictions. The collective models scored very poorly here, particularly the CTSP and the CMPL.

Collective MPL and TSP Models:
To understand the reason behaind the poor performance of *CMPL*, we compared the variance between the numbers of users choosing a particular landmark as their most popular location. The idea is that for a given time period of the day, *CMPL* will have a good prediction performance, if a large number of users share a particular landmark or set of landmarks as their most popular location(s). This means that for each time interval, in order to achieve good performance, we would expect the variance across the different landmarks to be high. The same idea can be applied to the *CTSP* model. After testing, we found that the highest variances across the different time intervals, for MPL and TSP models were 1.16 and 5.7431, respectively. One approach to improve the performances of CMPL and CTSP, would be to cluster users according to location or distance and then individually predict the mobility in each cluster using CMPL or CTSP.

Query Length:
We examined the effect of the length of the historical data used for querying the models to determine the relative loss of accuracy when this data is reduced. It is clear from the results shown in Table 5 that, on average, the more history the search query contained the better the prediction result, except for the search queries with history data length equal to 2. It is also clear that for queries length greater than 3, which account for 70.34% of total number of queries, the *ST* achieved only a very small improvement over the *CST* HM score. This a strong indication that the performances of the two models are very similar.

The Collective Model versus the One-model-per-user
The collective model has a number of advantages over the one-model-per-user:

- *Support for privacy:* The collective suffix tree model mitigates the privacy concerns by eliminating the need to build and store individual models which identify specific users. Moreover, the prediction is carried out using a short sliding window, which only reveals the more recent mobility history of the user and is far too narrow to form the basis of model building.
- *Social prediction:* Like other data mining methods, the one-per-user model can produce accurate predictions but only about previously observed behaviours. However, it has relatively poor performance when novel behaviours occur. On the other hand, the collective model can show better performance in such situations because it incorporates a range of behaviours from multiple users.
- *Serendipity:* As a recommendation method, the one-model-per-user would always predict places that had been previously seen by the user. Whilst this leads to making safe recommendations, it does not help the user to discover new places. Because it enables prediction using places seen by other users, the collective model can recommend new places that the observed user has not experienced in the past.

Model	MAE	RMSE	HM
MPL	15.7422	18.3290	0.1523
TSP	15.1946	17.9626	0.1514
CMPL	232.0867	243.1919	0.0222
CTSP	234.0202	244.1505	0.0161

Table 4: MAE, RMSE and HM (In this experiment, all target locations given in the test set have no visiting history in the training data set).

- *Less sensitivity to cold start situations:* When users are newly added to the system, they normally have no mobility history that can be employed to predict their next behaviour, a condition is known as cold-start. Because the collective model is based on multiple users, it is less sensitive to such situations compared to the one-model-per-user.
- *Cheaper to build:* The collective model costs less to build and maintain.

Despite their appeal, both models share a few shortfalls which we summarise as follows:

1. When there is no mobility history to consider for predicting the next location of visit (i.e. the user visited a location for the first time), the models predict the top-k most visited locations.
2. If the user has very low predictability (i.e. the user very often visits new places that he/she never seen in the past), there is high probability that the model would make an incorrect prediction.
3. For some users, matching the highest ranked search-trail does not necessarily lead to a correct prediction.

To address these problems we propose the following respective solutions:

1. The immediate solution, not neccessarily the most effective, is use TSP or MPL when the observed user has no mobility history to consider. An alternative approach would be to cluster landmarks based on their distance and build a collective model for each cluster.
2. Because it has better coverage, a collective model may be more suitable to predict users with low predictability.
3. In many cases, using shorter search-trails gives correct predictions. Therefore, for each prediction, we query the tree using the longest search-trail and all its shortened versions.

MAE and RMSE versus HM

To have a better perspective of the model accuracy, it is important to know, not only, whether or not the system is making correct predictions but also "how close" the prediction to matching the correct target location when system incorrectly predicts the user's next place of visit. Since the HM score is more focused on the hits as opposed to the misses, using it on its own, gives an imbalanced assessment of the prediction accuracy. Also, in many application areas, unless the HM score is very high which is very hard to achieve, the predictions cannot be reliable and most probably not very useful.

A sensible alternative would be for the predictor to present a short list of landmarks that are most likely to be of interest to the user. Showing, the user, a list of landmarks to choose from would, in many cases, be preferable to acting on a single landmark prediction. In the current context of next location prediction, MAE and RMSE could be employed very effectively, to determine the length of the list of suggested landmarks which would, most likely, include the correct next location to be visited by the observed user.

The main criticism is that, using one metric on its own may not give sufficient assessment of the prediction accuracy. A thorough evaluation of the results suggests that algorithms optimised for maximising HM

Length of record used for querying	ST			CST		
	MAE	RMSE	HM	MAE	RMSE	HM
1	1.4242	2.8551	0.6364	1.7576	3.7172	0.6970
2	1.6552	3.0850	0.6207	2.7931	4.8672	0.5862
3	1.2800	2.6533	0.6800	1.9200	4.0987	0.7200
> 3	1.0900	2.4907	0.7663	1.6717	3.8217	0.7597

Table 5: MAE, RMSE and HM, for ST and CST models, computed for different visiting history lengths (In this experiment, only target locations that have visiting history in the ST training data set were used).

score do not necessarily perform similarly when measured with the MAE and RMSE. Hence, striking a balance between the values of the three metrics is a key element in getting a good evaluation.

Conclusion and Further Work

We presented an alternative approach that allows for collaborative prediction and has the potential to overcome the one-model-per-user's weaknesses. We showed how the two approaches have very comparable performances particularly when previously seen behaviours are available. We also showed that only a short record of mobility history is required in order to make relatively accurate predictions. We investigated the merits of HM in evaluating the prediction performance and argued that it is insufficient when used on its own. We demonstrated that using the three metrics: MAE, RMSE and HM together for evaluation, gives a better view of the model's accuracy.

For our future work, we seek to improve the accuracy of the collective model though clustering the users on the basis of distance or through clustering of locations into different semantic categories.

References

[1] An, N. T., and Phuong, T. M. A gaussian mixture model for mobile location prediction. In *The 9th International Conference on Advanced Communication Technology*, vol. 2 (Feb. 2007), 914 –919.

[2] Ashbrook, D., and Starner, T. Learning significant locations and predicting user movement with GPS. In *Sixth International Symposium on Wearable Computers, 2002. (ISWC 2002). Proceedings* (2002), 101–108.

[3] Bejerano, G., and Yona, G. Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics 17*, 1 (Jan. 2001), 23–43.

[4] Borges, J., and Levene, M. Evaluating variable-length markov chain models for analysis of user web navigation sessions. *Knowledge and Data Engineering, IEEE Transactions on 19*, 4 (2007), 441452.

[5] Borges, J., and Levene, M. A comparison of scoring metrics for predicting the next navigation step. *November 3*, 2010 (2010), 15.

[6] Etter, V., Kafsi, M., and Kazemi, E. Been there, done that: What your mobility traces reveal about your behavior. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, vol. 2 (2012).

[7] Jacquet, P., Szpankowski, W., and Apostol, I. A universal predictor based on pattern matching. *Information Theory, IEEE Transactions on 48*, 6 (2002), 14621472.

[8] Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., and Laurila, J. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin* (2010).

[9] Krumm, J. A markov model for driver turn prediction. *SAE SP 2193*, 1 (2008).

[10] Pampapathi, R., Mirkin, B., and Levene, M. A suffix tree approach to anti-spam email filtering. *Machine Learning 65*, 1 (2006), 309338.

[11] Scellato, S., Musolesi, M., Mascolo, C., Latora, V., and Campbell, A. Nextplace: A spatio-temporal prediction framework for pervasive systems. *Pervasive Computing* (2011), 152169.

[12] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W. Y. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing* (2008), 312321.