Poster: A Mobility Prediction System
Leveraging Realtime Location Data Streams

Vaibhav Kulkarni*: vaibhav.kulkarni@unil.ch
Arielle Moro*: arielle.moro@unil.ch
Benoit Garbinato
benoit.garbinato@unil.ch
Distributed Object Programming Laboratory
University of Lausanne
CH-1015 Lausanne, Switzerland

ABSTRACT
Location-based services today, exceedingly depend on user mobility prediction, in order to push context aware services ahead of time. Existing location forecasting techniques are driven by large volumes of data to train the prediction models in a centralised server. This amounts to considerably long waiting times before the model kicks in. Disclosing highly sensitive location information to third party entities also exposes the user to several privacy risks. To address these issues, we put forth a mobility prediction system, able to provide swift realtime predictions, evading the strenuous training procedure. We enable this by constantly adapting the model to substantive user mobility behaviours that facilitate accurate predictions even on marginal time bounded movements. In comparison to existing frameworks, we utilise less volumes of data to produce satisfactory prediction accuracies. This in turn lowers the computational complexity making implementation on mobile devices feasible and a step towards privacy preservation. Here, only the predicted location can be sent to such services to maintain the utility/privacy tradeoff. Our preliminary evaluations based on real world mobility traces corroborate our hypothesis.

Categories and Subject Descriptors
H.4.2 [INFORMATION SYSTEMS APPLICATIONS]: Spatial-temporal systems

Keywords
Realtime Mobility Prediction; Mobility Behaviour; Location based Services

1. INTRODUCTION
The rapid proliferation in the number of applications offering Location-based Services (LBS), such as Google Now and Google Maps equipped with user location forecasting, makes it evident that mobility prediction is becoming a key paradigm of such services. However, numerous data breaches and malicious third party entities has casted a shadow over LBS. As can be seen, it is quite evident that their success depends on how well the user privacy is taken into account.

Existing mobility prediction techniques, utilise about 70% of the data, exclusively for model training. This results in substantial waiting times until the model is able to produce usable predictions in real deployment scenarios. We find that, a major downside with learning on a large dataset is the inherent nature of human movements. On the other hand, when this data is sent to third party servers, a malicious entity can easily infer sensitive user information such as significant places using simple heuristics. Furthermore, the algorithmic cost of making predictions on a mobile device is relatively high due to the complex ensemble techniques used, making it necessary to have a powerful server.

2. PROBLEM STATEMENT
AND CONTRIBUTIONS
The main goal of our approach is to reduce the amount of data required to produce predictions with a satisfactory accuracy to small time windows. We analyse the realtime user mobility behavioural changes to adapt this window length accordingly. We quantify user behaviours in terms of the

Figure 1: Traditional Prediction Systems vs. Our System. The process on the top, depicts the traditional mobility prediction approach. The process chain shown at the bottom gives an overview of our technique.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiCom ’16 October 03-07, 2016, New York City, NY, USA
DOI: http://dx.doi.org/10.1145/2973750.2985263
4.1 Mobility Behaviours

Frequently visited place evolution. We present the sudo code (see Algorithm 1) to obtain the user’s frequently visited places in realtime. Here, a cluster represents a unique visit in a delimited area and a cluster group represents a zone consisting of several clusters intersecting each other. Two intersecting clusters are merged. Finally, we define a

Algorithm 1 ZOI discovery algorithm

```java
Require: cluster, clusters, group, groups, zois
1: function ANALYZE(loc) → Called when a new loc is detected
2: if distance(loc, cluster.centroid) ≤ dmax then
3:     cluster.add(loc)
4: else
5:     if cluster.stayingTime() ≥ tmin then
6:         tryToMerge(groups)
7:     if tryToPut(cluster, groups) then
8:         create new group
9:         group.put(cluster)
10:        groups.add(group)
11:       clusters.add(cluster)
12:      if currentVisitNB < visitThreshold then
13:          currentVisitNB = visitNB(clusters)
14:      updateZOIs()
15:     create new cluster
16:     cluster.add(loc)
17: function UPDATEZOIs
18:     zois = {}
19:     for group ∈ groups do
20:         if group.clusterNB ≥ currentVisitNB AND time(now, group.lastVisit) ≤ tmax then
21:             zois.add(group)
```

Zone of Interest (ZOI) as a frequently and recently visited group of these clusters. Tracking these bounds over time enables to discover their evolution which in turn captures the user behavioural movement patterns.

Movement periodicity. A major challenge here is to identify periods, which do not repeat precisely at same times in addition to having multiple interlaced patterns in the non-stationary time series. Thus, standard period estimation techniques, such as autocorrelation and Fourier transform, cannot be directly applied. Further, the realtime location logs, cannot be assumed to arrive at a uniform rate. We use semivariance interpolation which conceals the incoming data stream about spatial variance at a specified distance to get a uniformly sampled stream. We further process the stream by taking the first difference and applying the log transform to get a series with constant variance. To calculate the periodicity, we calculate the power spectral density to get the candidate periods and feed them to autocorrelation estimator to rectify any false alarms resulting due to spectral leakage as shown in Figure 2.

We evaluate the system performance with Mobility Markov chain (MMC) models, simple classification, artificial neural networks, recurrent neural networks and Fourier extrapolation.

4.2 Overall System

We formulate the problem as a non-stationary time series prediction, where the model needs to be retrained according to variations in the incoming data stream. In our case, these correspond to user movements and the variations link to changing periodicities and number of ZOIs as depicted in Figure 1. We empirically determine that the model accuracies are affected for an autocorrelation index change of 0.2 and greater, which serves as a trigger for retraining. The feature vector consists of, ZOI evolution (representing the movements), starting and stay times which are fed to the predictors. In case of MMC models, their inherent structure leads to a strong dependency on the ZOI evolution pattern. The size of the training window in realtime is determined by computing the mean between each ZOI update, which sets
the threshold to limit the window length. In case the threshold is reached without detecting any updates, the training is halted and resumes with the detection of a new update. Thus, the continual tracking of ZOI updates and periodicity ensures the freshness of the predictions.

5. PRELIMINARY RESULTS AND DISCUSSION

We base our experiments on the Nokia dataset [3], consisting of mobility traces of 168 users for a period of at least 30 days. Our evaluation scheme is depicted in Figure 3. The prediction accuracy is considered as the fraction of samples for which the model correctly predicts the next location. We evaluate the entire dataset to determine the parameters of ZOI and set $d_{max}$, $t_{min}$ and visitThreshold to 60 meters, 900 seconds and 6 visits respectively. To simulate realtime incoming data, we read the data-points sequentially, according to the timestamps.

The preliminary findings of our evaluation are detailed in Table 1. As baselines, we compute the accuracies following the traditional approach of training on 70% data and evaluating on the rest. We observe that, 34% of users reach a satisfactory accuracy (>50% correct predictions for at least 30 days) with less than 100 days of data. We noticed that 2-order MMC and RNN based predictors achieve higher prediction accuracies. In case of 2-order MMC, this is due to accounting for the current state and the previous state to make the prediction. RRN blends the input vector at current state and the previous state to make the prediction. RRN is halted and resumes with the detection of a new update. Thus taking the entire history into account before making new predictions, effectively combining high level direction with low level modelling. We also saw a clear correlation between the periodicity and the accuracy of classification, neural networks and Fourier extrapolation, as these techniques weigh current state more than the past depicting linearity with the periodicity. In Figure 4, we show the running accuracy difference between our system and for all predictors against the baselines at each training model up to 60 days, with less than 100 days of data. We noticed that 2-order MMC and RNN based predictors achieve higher prediction accuracies, utilising lesser volumes of data, taking the mobility behaviour in to account.

Our future work, will attempt to have an ensemble approach to select suitable predictor in realtime according to behavioural changes, to achieve higher accuracies as we observed that certain family of predictors are better suited for particular mobility behaviours. We will also quantify the computational cost of the approach on an actual mobile device and optimise the process to have fewer model updates, which has a direct impact on the cost.

Acknowledgment. This work is partially supported by the Swiss National Science Foundation grant 146714.

7. REFERENCES


