Capturing Complex Behaviour for Predicting Distant Future Trajectories

Bertil Chapuis, Arielle Moro, Vaibhav Kulkarni, Benoît Garbinato
{firstname.lastname}@unil.ch
Distributed Object Programming Laboratory
University of Lausanne, Switzerland

ABSTRACT

We put forth a system, to predict distant-future positions of multiple moving entities and index the forecasted trajectories, in order to answer predictive queries involving long time horizons. Today, the proliferation of mobile devices with GPS functionality and internet connectivity has led to a rapid development of location-based services, accounting for user mobility prediction as a key paradigm. Mobility prediction is already playing a major role in traffic management, urban planning and location-based advertising, which demand accurate and long time horizon forecasting, of user movements. Existing prediction methodologies, either use motion patterns or techniques based on frequently visited places for predicting the next move. However, when it comes to distant-future, human mobility is too complex to be represented by such statistical functions. Therefore, the existing techniques are not well suited to answer distant-future queries with a satisfactory level of accuracy. To tackle this problem, we introduce a novel spatial object, *'Representative Trajectory'*, which embodies the movements of users amongst their zones of interest. We propose means to empirically evaluate the quality of this object and dynamically adapt its extraction method based on user mobility behaviour. We rely on an inverted index to store the predicted trajectories that scales well with the number of moving entities. Our evaluation results show that the technique achieves more than 70% accurate predictions with the best extraction technique. This shows that longer query time horizons do not necessarily demand complex spatial indexing schemes, which have to be rebalanced as they grow, which is a constantly experienced problem while answering predictive queries.

Categories and Subject Descriptors

H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Content Analysis and Indexing

Keywords

Mobility Prediction; Trajectory Prediction; Spatial Indexing; Moving Object Database

© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

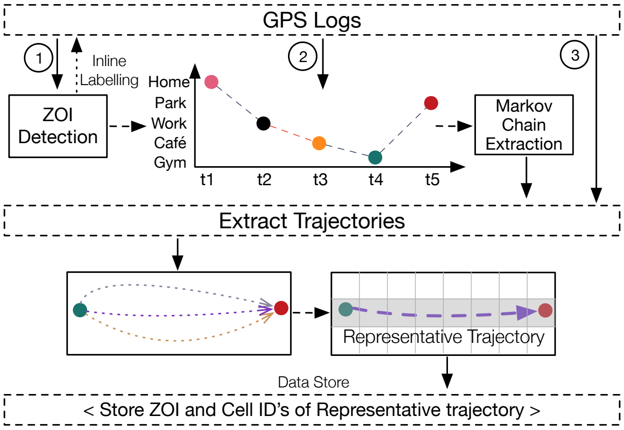


Figure 1: Overview of the System Model.

1. INTRODUCTION

The booming trend of ubiquitous computing, behavioural prediction and ease of availability of internet services, is directly impacting the way we store, retrieve, process and query data. The coming era will witness dramatic advances in the domain of positioning technologies and localisation. Therefore, location tracking and user mobility prediction is becoming increasingly important. Formulating prediction techniques to attain satisfactory accuracies at high granularities puts forth several challenges. Firstly, maintaining high accuracy is crucial for applications such as urban planning, traffic prediction and managing fleets of autonomous vehicles. Secondly, when multiple moving users are involved, it is important to have a scalable indexing technique specially while answering predictive queries.

The existing solutions to the aforementioned problems are not well adapted to solve all the requirements mentioned above. Firstly, the prediction techniques that are built on motion functions, pattern mining or temporal extrapolations, do not truly capture the complex nature of human movement. This factor is especially critical when accounting for distant future predictions, which typically lie in the order of several hours, for which such statistical means fail. The second problem is, the existing techniques, which attempt to model long time horizon predictions, collect a set of frequently visited places by the user and formulate the future place prediction within this set. Such techniques lose the information lying in between these frequently visited places, which is essential to achieve the fine granularity while predicting. An application, where such a level of prediction comes in useful, is to answer predictive queries

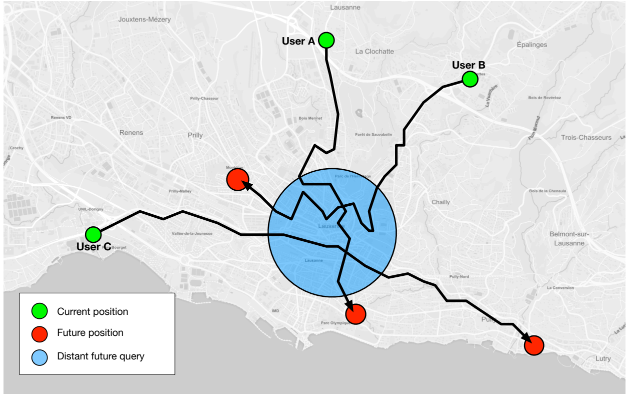


Figure 2: An example of a distant future query represented with 3 mobile users.

related to vicinity matching around some locations that are not necessarily included in the set of frequently visited places of a user, but may lie on the trajectory taken to travel from one frequent place to another. The present techniques, which index trajectories, rely on tree structures that require rebalancing when the tree is updated and thus present scaling issues specially when multiple moving points are involved.

In this paper, we put forth solutions to the above problems and propose a complete system capable to answer the kind queries as expressed below, and depicted in Figure 2. The query can be expressed as: *"Select all the users who will travel in the vicinity of a given location during their next move with a probability higher than a given threshold"*. We first extract the frequently visited places of the user, formally called Zones of Interest (ZOIs), and the transitions amongst them that will be used to train a Markov mobility model. We then extract the past trajectories in order to attain all the possible paths the user takes amongst ZOIs. We introduce a novel spatial object, "representative trajectory" that captures the substantive user mobility behaviour and adapts its extraction according to the dynamically changing user movements. The representative trajectories are then indexed when a user enters in a ZOI to answer predictive queries. The system overview is depicted in Figure 1. Our key contributions are listed hereafter.

- We put forth a complete system capable of predicting distant future trajectories of mobile users and thus answering matching queries for multiple moving users. This is made possible by taking the transitions within the individual ZOIs, thus accounting for trajectories lying within and therefore achieving higher granularity.
- Our ZOI and representative trajectory computation scheme considers dynamic user movements and adapts the computation parameters according to the mobility behaviours. We introduce a novel spatial object, called *'Representative Trajectory'*, which captures the practical nature of human mobility, by considering the fact that, users can move between two ZOIs through different paths. We further discuss and derive means to extract the best path amongst the several paths to represent the most significant trajectory of the user.
- We describe an evaluation framework that utilises precision and recall to access the quality of representative trajectories.
- Lastly, we present a practical indexing technique based on inverted index, in order to store the trajectory predictions. This technique does not demand costly rebalancing actions as opposed to existing tree structures.

2. RELATED WORK

Indexing past, current and future positions of moving entities in order to answer predictive queries is an actively research topic, due to the ubiquity of location based services. Therefore, it is important to distinguish our contribution from the plethora of existing works. At the top level, the literature can be separated, into queries related to a single point moving in one dimensional space, viz., *"find all café's around me in next hour"* [15, 18, 16] and queries accounting for multiple moving points in space, viz., *"find all users that will be in the vicinity of café X in next hour"*. Secondly, the existing work can also be separated on the basis of the sampling rate of tracking the moving object locations. Low sampling rate based approaches rely on manual checkins by the users that may range in the order of one location log a day. On the other hand, high sampling rate based approaches, track the user locations after every few seconds. Both these approaches, demand different prediction and indexing techniques and the low sampling rate based technique as presented in [2] is beyond the scope of this work. The proliferation of mobile devices with GPS functionality and uninterrupted internet services today, foster and ease the process of continually tracking the moving objects with a high sampling rate. Hence our focus lies on location logs collected at high sampling rates. Further, the time horizon of the query window is an important aspect, which can be dissected into near future and distant future queries. Majority of the published work today, focuses on near future queries in the order of next 15 minutes [15, 18, 16, 14, 1]. However, our work focuses on distant future queries in the order of several hours. The work attempting to solve distant future queries relies on motion prediction techniques by modelling the movement in terms of motion patterns, motion functions and temporal extrapolation [17]. However, according to our observation, such prediction methodologies fail to grasp and accurately represent long term user movements. Therefore, we utilise Markov models to perform distant movement predictions. We further discuss how to index such predictions for efficiently answering the queries we described in Section 1. To summarise, our work lies in answering vicinity matching queries for multiple moving points, whose location logs are tracked at a high sampling rate. We focus on distant future queries that demand accurate prediction methodology for which we depend on mobility Markov models.

Regarding the prediction techniques, a majority of existing work is focussed on predicting movements between certain points of interests [3, 20, 13, 5, 4, 12]. A domain of research also relies on cell based techniques for making predictions at the granularity of network cells [3, 12]. Such schemes completely ignore the trajectories lying in between the individual places, which is critical to answer distant future queries, involving multiple moving points with a high degree of accuracy. Additionally, the size of a typical network cell lies in the range of several kilometres, which is not adequate to answer queries related to fine grained vicinity matching. On the other hand, predictions based on map matching techniques are complex and need additional services such as network availability, which is not always feasible and is computationally expensive [9]. Existing prediction techniques considering user trajectories amongst points of interests do not store these models, which is a critical factor to answer certain queries. Further, Kalman filter based prediction approaches, involve higher complexity and thus results in higher latency as discussed in [11]. Our work consists of estimating the ZOIs in which a user spends considerable amount of time and then attain the representative trajectory in between these zones that assist to answer the queries described above. Several techniques have been demonstrated to extract points of interest of users whose central theme is based on clustering [16]. As compared to these traditional approaches, our clustering technique enables to extract the frequently visited places of a user according to the mean of the number of visits, the time spent and the distance covered in the significant location, which better represents such a place. Additionally, setting the spatiotemporal bounds based on individual mobility behaviour allows to extract places that are not necessarily found with direct clustering. Further, we follow a technique to dynamically adjust the parameters to extract the representative trajectory based on the user behaviour to consistently maintain satisfactory levels of precision and recall. Thus, unlike the methods presented in the literature, we account for the user behaviour to set parameters for both, extracting the ZOIs and representative trajectories lying in between the zones.

In traditional indexing schemes, the content is only altered when users explicitly perform updates. This is as opposed to indexing moving object locations, where the data quickly becomes outdated and continual write operations are necessary to keep the data updated. A common approach to address this issue and decrease the number of updates is to adopt an alternative model for representing the location of moving objects. In [15], Saltenis et al. present techniques to index positions of continuously moving objects. The position of the objects is modelled as linear/non-linear function of time and velocity. They present efficient techniques to index trajectories and partition R-Tree containing motion functions. However, as previously discussed, such techniques fail to accurately formulate distant future predictions. In [6], Hendawi el al. present a framework to predict answers to queries as well as queries themselves by monitoring high query rate areas. However, this technique is restricted to a single object prediction. In [7], Bao el at. propose an index structure for processing predictive queries, however with the assumption that moving objects follow shortest paths during their travel from source to the destination, which is not necessary true in practice according to our observation on real work mobility traces. In [16], Tao et al. present methods to predict and index unknown motion patterns of moving objects using recursive motion patterns to express complex trajectories. In [19], Yanagisawa et al. model the motions into 3 distinct categories including staying, moving straight and moving randomly. In [8], Jeung et al. present a hybrid prediction model for near future and distant future predictions. However, all these methods are either based only on frequently visited places and ignore encompassed trajectories or fail to model distant future predictions. The indexing techniques, discussed in [15, 18, 16], rely on tree structures that require rebalancing as the number of moving points increase and thus do not necessarily scale well. As a result, our approach is based on a simplistic and practical strategy, which is an inverted indexing model.

3. SYSTEM MODEL AND DEFINITIONS

In this section, we introduce our system model, with formal definitions and notations.

3.1 Users and Locations

We consider a set of users $U = \{u_1, \ldots, u_n\}$ moving on the surface of the earth with mobile devices that have the ability to locate themselves, typically via the Global Positioning System (GPS)[1] or some other positioning means, e.g., WiFi Positioning System (WPS).[2] The definitions presented below are from the view point of one user. The location history of the user is expressed as a sequence L of n locations, as $L = \langle loc_1, \ldots, loc_n \rangle$. Each location, loc_i contained in L, is represented by a 3-item tuple $loc_i = (\phi, \lambda, t)$. The latitude and longitude of the coordinate are represented by $\phi, \lambda \in \mathbb{R}$ respectively, and its timestamp by $t \in \mathbb{N}$.

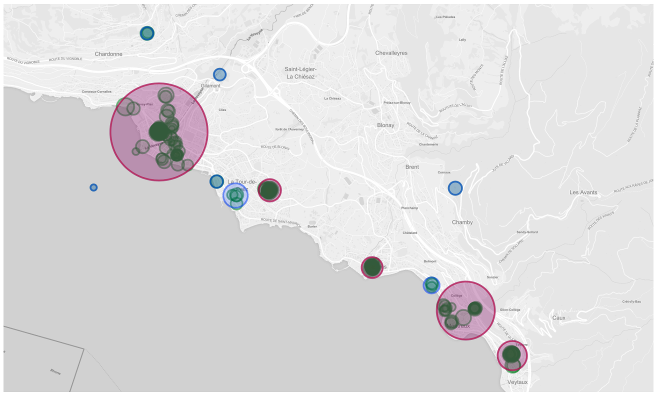


Figure 3: Clusters, Cluster Groups and Zones of Interest.

3.2 Clusters and Zones of Interest

In order to extract the Zones of Interest (ZOI) of a user based on the location history L, we must first introduce the notions of cluster and cluster group.

3.2.1 *Cluster*

A cluster represents a visit or a stay in a delimited area. It is formed from a subset of locations, sharing the same spatial and temporal characteristics. A cluster is a 4-item tuple $c = (\phi, \lambda, \Delta r, l)$, where ϕ and $\lambda \in \mathbb{R}$ are the latitude and longitude coordinates of a centroid, $\Delta r \in \mathbb{R}$ is its radius in meters and $l \in L$ is the subset of successive locations belonging to c. The centroid of the cluster is the mean of all ϕ and λ of the locations contained in l. Here, the radius corresponds to the maximum distance between the centroid of the cluster and the locations belonging to l. In order to build clusters, we introduce the constraining constants Δd_{max} and $\Delta t_{min} \in \mathbb{R}$, which correspond to the distance expressed in meters and a time duration expressed in seconds respectively. The distance between all the locations lying inside l and the centroid of the cluster must be lower than or equal to Δd_{max}. In addition, the duration between the first location and the last location must be greater than or equal to Δt_{min}. On this basis, we introduce C, the set of clusters extracted from the location history of a user as $C = \{c_1, \ldots, c_n\}$.

3.2.2 *Cluster Group*

A cluster group is an aggregation of overlapping clusters. Formally, a cluster group is as 4-item tuple $g = (\phi, \lambda, \Delta r, \{c_1, \ldots, c_n\})$. The first three items of a cluster group are the same as the ones present in a cluster. Clusters are grouped whenever they overlap. Consequently, the last item of the tuple corresponds to the set of n overlapping clusters belonging to C. The centroid of the cluster group is the mean of all the centroids of the clusters contained in g, and Δr must be computed in order enclose all the individual clusters present in g. Finally, we introduce G which contains the n cluster groups belonging to a user as $G = \{g_1, \ldots, g_n\}$.

[1] http://www.schriever.af.mil/GPS

[2] http://en.wikipedia.org/wiki/Wi-Fipositioningsystem

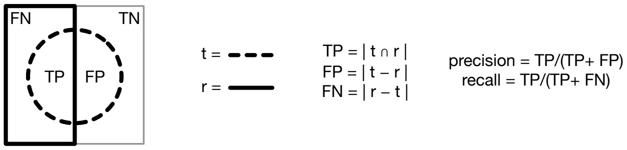


Figure 4: Binary classification in the context of trajectories

3.2.3 Zone of Interest(ZOI)

Intuitively, a ZOI is a cluster group that is frequently visited by a user. Let $v_{min} \in \mathbb{N}$ be a constant that represents a minimal number of visits. A cluster group becomes a ZOI if and only if the number of clusters in the group is greater than or equal to the constant v_{min}. However, if we only take into account this constant, it is not possible to find the ZOIs of a user from the beginning of the process, especially if a very high value is set a priori. To resolve this issue, we introduce a variable v_{mean} which is the mean of the number of visits per cluster for a user. This value acts as a reference visit threshold until reaching v_{min}. A ZOI consists of the same items as those of g, further denoted as z to distinguish the two tuples. The centroid and the radius values of z are the same as for g. In addition, we introduce a set Z containing the n ZOIs of a user represented as $Z = \{z_1, \ldots, z_n\}$.

3.3 Trajectories

A user can take multiple paths to move from one ZOI to another. Consequently, we introduce the set of trajectories $T_{i,j}$ that can be extracted from the raw set of locations L. Formally, $T_{i,j}$ is a set of n trajectories $T_{i,j} = \{l_1, \ldots, l_n\}$, where each trajectory l_i is a substring of the sequence L in which the first location is contained in z_i and the last location is located in z_j. In addition, the locations recorded between z_i and z_j in l_i do not pass over trajectories going towards any other ZOI.

3.4 Mobility Prediction Model

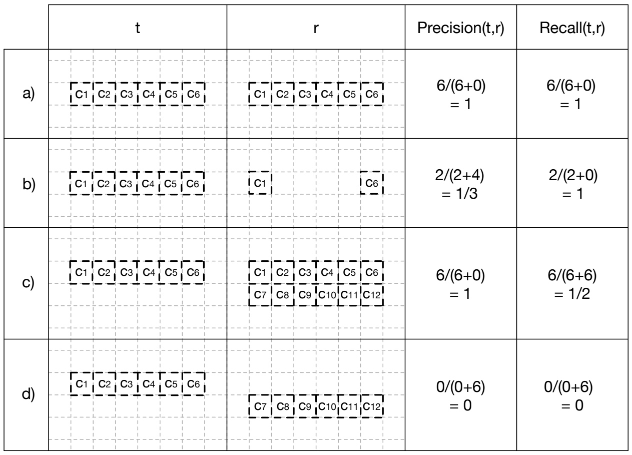
In this work, we consider a mobility prediction model following the structure of a first order Markov chain. Each user has a unique Markov chain, computed on the basis of the elements defined previously. Equation 1 depicts a matrix M containing $n \times n$ transitions probabilities, where $n = |Z|$. In other words, each ZOI is a state of the matrix M and a transition probability $p_{i,j}$ represents the probability to move from a specific z_i to another z_j. As shown in Equation 2, the transition probabilities $p_{i,j}$ of the matrix M can be computed using the cardinalities of the sets of trajectories $T_{i,j}$.

$$M = \begin{bmatrix} p_{1,1} & \cdots & p_{1,i} & \cdots \\ \vdots & \ddots & \vdots & \ddots \\ p_{j,1} & \cdots & p_{i,j} & \cdots \\ \vdots & \ddots & \vdots & p_{n,n} \end{bmatrix} \tag{1}$$

$$p_{i,j} = \frac{|T_{i,j}|}{\sum\limits_{z_k \in Z} |T_{i,k}|} \tag{2}$$

4. PREDICTING TRAJECTORIES

Given two ZOIs z_i and z_j, we consider the problem of computing the trajectory that accurately predicts the future moves from one zone to the other. In other words, the idea consists in extracting spatial objects from the actual trajectories that represent the future trajectories of a user between ZOIs. We call these spatial objects *representative trajectories* as they capture the essence of the past movements of a user amongst ZOIs. In this section, in order to evaluate the predictive capacity of a representative trajectory, we first introduce some measures that can be used to assess their accuracy. We then evaluate several strategies for building representative trajectories, some of which take the behaviour of the user into account in order to make the predictions more relevant.



	t	r	Precision(t,r)	Recall(t,r)
a)	C1 C2 C3 C4 C5 C6	C1 C2 C3 C4 C5 C6	6/(6+0) = 1	6/(6+0) = 1
b)	C1 C2 C3 C4 C5 C6	C1 C6	2/(2+4) = 1/3	2/(2+0) = 1
c)	C1 C2 C3 C4 C5 C6	C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12	6/(6+0) = 1	6/(6+6) = 1/2
d)	C1 C2 C3 C4 C5 C6	C7 C8 C9 C10 C11 C12	0/(0+6) = 0	0/(0+6) = 0

Figure 5: Precision and recall in the context of trajectories

4.1 Evaluating Representative Trajectories

The first question to answer is, *"how can the effectiveness of a representative trajectory, be measured in the context of a predictive query?"*. In information retrieval, the performance of a system is often measured in terms of precision and recall. Given the results of a query, binary classification is used to assess how many of the results are relevant (precision) and how many relevant results were selected (recall). Similarly, given the actual trajectory followed by a user and the corresponding predicted trajectory, it is possible to assess how many subparts of the trajectory are relevant according to the predicted trajectory (precision) and how many subparts of the predicted trajectory were selected (recall). To achieve this goal, we discretise space by introducing a *Grid* that allows us to perform binary classification on all the subparts of a trajectory. More formally, a *Grid* can be described as a set of uniquely identified cells, such that $Grid = \{cell_1, \ldots, cell_n\}$. Figure 4, illustrates the calculation of precision and recall in the context of discretised trajectories. Assuming t a set of cells corresponding to a actual trajectory and r a set of cells corresponding to a representative trajectory, the number of true positive cells TP can be expressed by the cardinality $|t \cap r|$, the number of false positive cells FP can be expressed by the cardinality $|t - r|$ and the number of false negative cells FN can be expressed by the cardinality $|r - t|$. Consequently, *precision* can be expressed as $TP/(TP + FP)$ and recall by $TP/(TP + FN)$. On this foundation, we formally introduce *Precision* and *Recall* which are defined in Equations 3 and 4. Finally, the measure F_β, often denoted as F-score, combines precision and recall in a single metric that can be expressed as the weighted harmonic mean described in Equation 5. Depending on the situation, one may decide to give more importance to precision or to recall by adjusting the weight factor β.

$$Precision(t,r) = \frac{|t \cap r|}{|t \cap r| + |t - r|} \tag{3}$$

$$Recall(t,r) = \frac{|t \cap r|}{|t \cap r| + |r - t|} \tag{4}$$

$$F_\beta(t,r) = (1+\beta^2) \cdot \frac{\text{Precision(t,r)} \cdot \text{Recall(t,r)}}{\beta^2 \cdot \text{Precision(t,r)} + \text{Recall(t,r)}} \tag{5}$$

Figure 5 shows how precision and recall can be calculated given the cells of an actual trajectory t and the cells of a representative trajectory r. As illustrated here, precision and recall accurately answer the questions how many subparts of the actual trajectory t are relevant according to the representative trajectory r and how many subparts of the predicted trajectory were selected. Figure 5a shows that, if t and r perfectly overlap, then precision and recall are high. Figure 5b shows that, when r is a subset of t, then precision drops because only few cells of the actual trajectory are relevant according to the representative trajectory. Figure 5c shows that if t is a subset of r, then recall drops, because only a few cells of the representative trajectory were selected. Finally, Figure 5d shows that, when t and r do not overlap, precision and recall are both low.

4.2 Building Representative Trajectories

In the previous section, we have not considered how the cells of a representative trajectory were actually selected. In order to build this representative trajectory, we first introduce the set $\tau_{i,j}$ that is a discretised version of the set of trajectories $T_{i,j}$. More formally, $\tau_{i,j}$ is a set of n discretised trajectories $\tau_{i,j} = \{t_1, \ldots, t_n\}$, where each trajectory t_k is a set of n cells such that $t_k \in \tau_{i,j} : \{cell_1, cell_2, \ldots, cell_n\}$. The importance of a cell in a *representative trajectory* is defined by its number of occurrences in $\tau_{i,j}$. Thus, we introduce the multi set $O_{i,j}$ that counts the number of occurrences of a cell in $\tau_{i,j}$ and can formally be defined as $O_{i,j} \subseteq Grid \times \mathbb{N}^*$. Finally, the cells with a number of occurrences greater than a given threshold are gathered in set $R_{i,j}$ that constitutes the representative trajectory. As a consequence, when building a representative trajectory, the main challenge consists in selecting a threshold $\theta \in \mathbb{N}^*$ that will select the most accurate and relevant cells for predicting the future moves of a user. In a more formal way, given the threshold θ, a representative trajectory $R_{i,j}$ can be obtained with the function described in Equation 6.

$$R(O_{i,j}, \theta) = \{c | (c,n) \in O_{i,j} \wedge n \geq \theta\} \tag{6}$$

In addition to these general definitions, we introduce some utility functions that can be used to select thresholds that take the behaviour of the user into account. The function $Mean(O_{i,j})$, formally defined in Equation 7, returns the mean number of cell occurrences of the multiset $O_{i,j}$. In a similar way, the functions $Min(O_{i,j})$ and $Max(O_{i,j})$, defined in Equations 8 and 9, return the minimum and maximum number of cell occurrences of $O_{i,j}$ respectively.

$$Mean(O_{i,j}) = \frac{\sum\limits_{(c,n) \in O_{i,j}} n}{|O_{i,j}|} \tag{7}$$

$$Min(O_{i,j}) = \min_{(c,n) \in O_{i,j}} n \tag{8}$$

$$Max(O_{i,j}) = \max_{(c,n) \in O_{i,j}} n \tag{9}$$

In the previous section we introduced *Precision* and *Recall* in the context of a trajectory t and a representative trajectory r. Since $\tau_{i,j}$ contains several discretised trajectories, we introduce the functions $AvgPrecision(\tau_{i,j}, r)$ and $AvgRecall(\tau_{i,j}, r)$ respectively defined in Equations 10 and 11 which measure the average precision and the average recall for multiple sets of cells.

$$AvgPrecision(\tau_{i,j}, r) = \frac{\sum\limits_{t \in \tau_{i,j}} Precision(t,r)}{|\tau_{i,j}|} \tag{10}$$

$$AvgRecall(\tau_{i,j}, r) = \frac{\sum\limits_{t \in \tau_{i,j}} Recall(t,r)}{|\tau_{i,j}|} \tag{11}$$

4.2.1 Mean Threshold

An obvious approach to select a threshold, consists in using the mean cell occurrences as illustrated in Equation 12. While this method is straightforward and computationally efficient, it does not account for the behaviour of the user within the ZOIs. For example, during the week, a user may always take the same route to go from home to work, while during the weekend he would leave home for excursions and come back at the same place. A threshold, based on the mean of the cell occurrences, is not adapted to capture this kind of behaviour.

$$\theta_{mean} = Mean(O_{i,j}) \tag{12}$$

4.2.2 F-Score Threshold

In order to build better representative trajectories, we can consider the problem of selecting the threshold as an optimisation of the F_β score introduced in Section 4.1. In other words, given the trajectories of a set $\tau_{i,j}$, the idea consists in computing the average F-Score for all the possible thresholds. Then, the threshold that gives the best average score for F_β can be considered as the best possible threshold for the representative trajectory. More formally, assuming a set of candidate representative trajectories $CR_{i,j}$ expressed in Equation 13, one can find the representative trajectory that gives the best F-Score, as described in Equations 14 and 15.

$$CR_{i,j} = \{r_\theta = R(O_{i,j}, \theta) | \theta \in [Min(O_{i,j}), Max(O_{i,j})]\} \tag{13}$$

$$\forall r_\theta \in CR_{i,j} : F_\beta^\theta = \frac{\sum\limits_{t \in \tau_{i,j}} F_\beta(t, r_\theta)}{|\tau_{i,j}|} \tag{14}$$

$$F_\beta^{max} = \max_{r^\theta \in CR_{i,j}} F_\beta^\theta \tag{15}$$

Finally, θ_{F_β} can be defined as the min θ for which $F_\beta^{max} = F_\beta^\theta$. With this approach, one can decide to give more importance to precision or recall by adjusting the β parameter. The main disadvantage of this technique lies in the fact that, F_β scores have to be computed over all the possible thresholds, which is computationally expensive.

4.2.3 Adaptative Threshold

A third approach consists in assuming that precision and recall, both contain meaningful insights, regarding the behaviour of the user, that can be used to adapt an existing threshold. First, as illustrated in Figure 5a, if the user always follows the same path, the representative trajectory is easy to build and the precision and recall are both expected to be very high. In that case, no actions

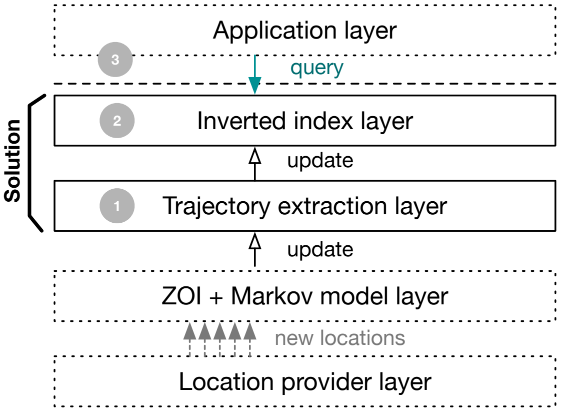


Figure 6: Layers of the Solution Architecture.

are needed. Second, as illustrated in Figure 5b, a low precision and a high recall suggests that the user takes several paths to go from one place to the other. Some regular paths are not included in the representative trajectory, which causes the precision to drop. Consequently, the threshold used to select the cells can be lowered in order to include these cells in the representative trajectory. Third, as illustrated in Figure 5c, a low recall with a high precision suggests exactly the opposite. The user follows several paths and some insignificant ones are included in the representative trajectory. Consequently, the threshold used to select the cells can be strengthened. Finally, as illustrated in Figure 5d, if the user does not have a consistent behaviour between two ZOIs, which usually happens for week-end excursions, the precision and recall both drop and can be used as insights on the poor quality of the representative trajectory. In order to account for these different scenarios, we first compute a representative trajectory r using the mean threshold, such that $r = R(O_{i,j}, \theta_{Mean})$. On this basis, Equation 16 introduces a readjusted threshold that accounts for the user behaviours as discussed.

$$\theta_A = Max(O_{i,j}) * \frac{AvgPrecision(\tau_{i,j}, r) + (1 - AvgRecall(\tau_{i,j}, r))}{2} \tag{16}$$

5. SOLUTION ARCHITECTURE

The previous sections showed how representative trajectories can be extracted from the location data of a single user. In this section, we go back to the initial requirement which consists in a predictive query involving a set of users. We present a solution architecture based on an inverted index that can be used to answer such queries. In its simplest form, an inverted index is composed of two parts: a *dictionary of terms* where each element points to a *postings list*. In the context of text retrieval, the terms of the dictionary would correspond to words and the postings would correspond to lists of documents. In our context, the terms of the dictionary are the cells of a *Grid* and the postings correspond to tuples $(u, p_{i,j})$ where u is a user identifier belonging to U and $p_{i,j}$ corresponds to the transition probability between two ZOIs, z_i and z_j belonging to M as introduced earlier. On this foundation, the architecture can be divided into three distinct procedures that act on the different layers depicted in Figure 6.

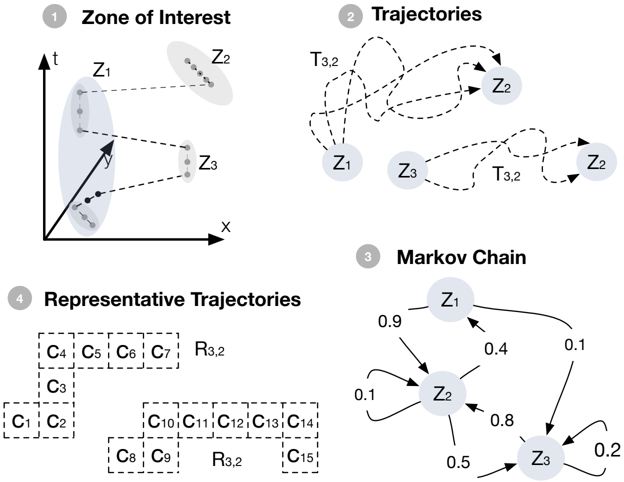


Figure 7: Extracting Representative Trajectories.

5.1 Prediction Model Extraction

The first procedure aims at extracting the representative trajectories from the location history of a user. Figure 7 recalls the four main successive steps involved in the extraction of the representative trajectories. These steps are described as below and executed for each user u belonging to the set U.

1. **Zones of Interest discovery.** As introduced in Section 3, the procedure, first uses a clustering algorithm to extract the set of ZOIs, called Z, from the location history L.

2. **Trajectories extraction.** On the basis of the discovered ZOIs, the sets of trajectories $T_{i,j}$, amongst ZOIs can be extracted by examining the location history L a second time.

3. **Markov chain computation.** The sets of trajectories $T_{i,j}$ can then be used to compute the transition probabilities $p_{i,j}$ of the Markov chain M. The Markov chain M is then stored for later use.

4. **Representative trajectory extraction.** The set of trajectories $T_{i,j}$ can also be used to compute the representative trajectories $R_{i,j}$. All the representative trajectories $R_{i,j}$ are then persisted for later use.

5.2 Inverted Index Update

Using the previously persisted items, the second procedure updates the inverted index every time a user enters in one of the ZOIs. Before presenting the procedure in detail, we must introduce variables z_i and z_j that are assumed to be the current and the predicted ZOIs of a user respectively. Figure 8 depicts the four following steps enabling to update the inverted index.

1. **Current ZOI extraction.** The procedure is triggered when a user u enters in one of the ZOIs and the identified ZOI becomes the current ZOI z_i. From that point, it is possible to find all the next possible ZOIs and their respective transition probabilities in the matrix M.

2. **Next ZOI prediction.** The predicted ZOI z_j corresponds to the ZOI with the maximal transition probability $p_{i,j}$ in the Markov chain M amongst the transitions from z_i.

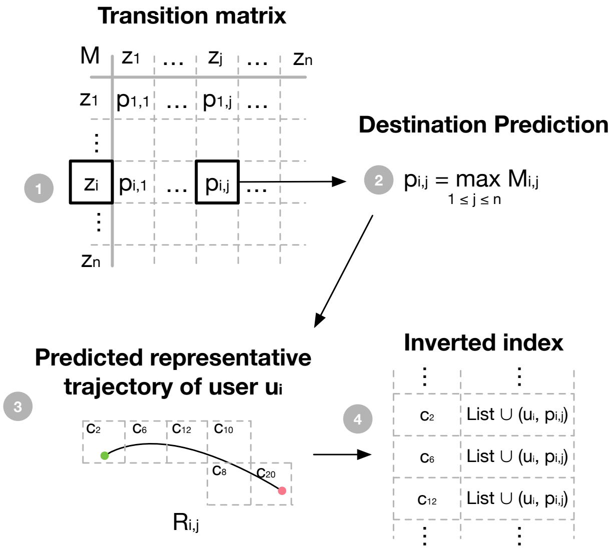


Figure 8: Updating Inverted Index.

3. **Representative trajectory retrieval.** As soon as the current and predicted ZOIs are identified, the representative trajectory $R_{i,j}$ associated with z_i and z_j can be retrieved amongst all the representative trajectories stored for the user.

4. **Inverted index update.** After the retrieval of the representative trajectory $R_{i,j}$, the inverted index must be updated. To do so, the tuple $(u, p_{i,j})$ is added to all the postings lists pointed by the cells of the representative trajectory.

5.3 Answering the query

The third procedure is used to answer queries. We consider queries of the following nature: *"Select all the users who will travel in the vicinity of a given location during their next move with a probability higher than a given threshold"*. We assume that we know the search zone to initiate the query, which can be formally expressed as a tuple $search_{zone} = (loc, \Delta r)$ by the requestor. In addition, the probability threshold stated in the query is denoted p_{th}. In order to answer the query the $search_{zone}$ is first converted into a set of cells belonging to the *Grid* and called $search_{cells}$. As illustrated in Figure 9, assuming that the predicted representative trajectories $R_{i,j}$ have been added to the inverted index for each user belonging to U, it is now possible to retrieve all the users who will move through the cells specified in the query. In other words, users are selected if the following two conditions are met. First, when at least one cell of their representative trajectory of their next predicted move matches with one of the $search_{cells}$. Second, if their probability $p_{i,j}$ associated with their matching cell is greater than or equal to the probability threshold p_{th}.

6. EVALUATION AND DISCUSSION

We perform the evaluation of the system based on the Nokia data set [10], which consists of mobility traces collected from 188 users around lake Geneva region in Switzerland from October 2009 to March 2011. The mean duration of the participants, which mainly consisted of university students and professionals, was about 14 months consisting of more than 10 million location points. We implemented the components of our architecture, including the clustering algorithm, the Markov chain and the representative trajectory extraction algorithms in Scala. We used the Java Google S2 Library in order to discretise space [3]. The grid provided by this library comes with the guarantee that the cells will have similar areas and we configured it to produce cells which are one square kilometre on an average. In the context of this evaluation, we used a first order Markov chain to predict the movements of a user across ZOIs. This choice was motivated by the fact that, our experiments with second order Markov chains showed a gain of only 2% in terms of accuracy for predictions made with the whole dataset. We believe a much greater gain in terms of accuracy can be obtained by simply cleaning and sanitising the dataset.

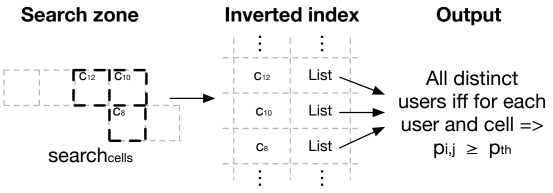
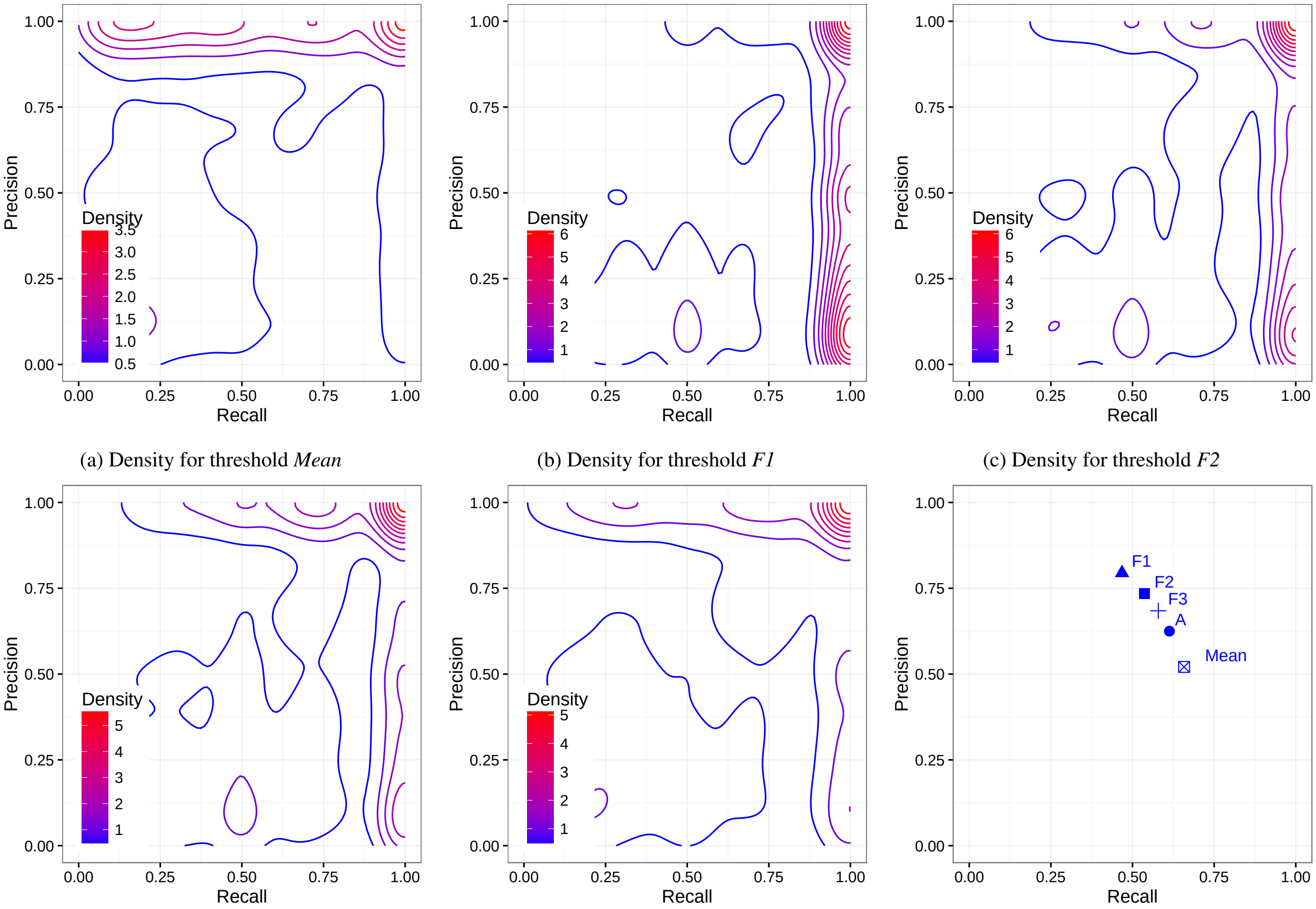


Figure 9: Answering the Query.

In order to evaluate our methods for building representative trajectories, we used 70% of the dataset for creating the Markov chains and the representative trajectories. Using the remaining 30% of the dataset, we performed 4727 trajectory predictions and evaluated the quality of the outputs using the actual trajectories followed by the users. Figure 10 uses two dimensional kernel density estimate (KDE) to evaluate these predictions in terms of precision and recall. In these plots, a high density corresponds to a large concentration of predictions. The density can be greater than one as the probability is multiplied by an area of the two dimensional space. In these plots, the upper-right corner is the sweet spot. As illustrated in Figure 10a we ideally foster predictions characterised by a high precision and a high recall. The lower-right corner would typically contain predictions as the one illustrated in Figure 10b. Such predictions are symptomatic of a strong threshold that filters too many cells out of the representative trajectory. On the contrary, the upper-left corner would typically contain predictions as the one illustrated in Figure 10c. Such predictions are often synonym of a weak threshold that preserves too many cells in the representative trajectory. Finally, the lower-left corner contains results characterised by a poor precision and a poor recall as the one illustrated in Figure 10d. Such predictions can be synonymous with an inconsistent behaviour between the ZOIs or with a completely wrong result at the level of the Markov chain.

The five threshold selection methods we evaluate in order to build representative trajectories are *Mean* (θ_{Mean}), *F1* (θ_{F_1}), *F2* (θ_{F_2}) and *F3* (θ_{F_3}) and *A* (θ_A). In Figure 10a, the threshold is set to the *Mean* number of occurrences of the cells in the representative trajectory. This plot highlights a high density on the upper side, i.e, predictions are usually characterised by a high precision but recall varies a lot. As previously stated, this gives us the intuition that the threshold is too weak. When the user takes several distinct paths to go from one ZOI to the other, some irrelevant cells are included in the representative trajectories. Therefore, this method fails at accurately accounting for the behaviour of the user and may return a significant number of cells when used in practice. In Figure 10b,

[3] https://github.com/google/s2-geometry-library-java



(a) Density for threshold *Mean*

(b) Density for threshold *F1*

(c) Density for threshold *F2*

(d) Density for threshold *F3*

(e) Density for threshold *A*

(f) Average precision and recall

Figure 10: Precision-Recall Kernel Density Estimation (KDE) for various thresholds selection methods

the threshold is obtained by searching a value that gives the best F_β score when β is set to 1. In that case, the density plot highlights predictions characterised by a very high recall but a highly varying precision. This is symptomatic of a strong threshold value that filters out most of the cells of the representative trajectory. In other words, the predicted cells will be highly accurate but cover only a subpart of the trajectory followed by the user in practice. In Figure 10c and 10d, the β score is respectively set to 2 and 3. Here the tradeoff between precision and recall is clearly highlighted by the fact that, the predictions shift towards a better balance between precision and recall. The positive impact of adjusting the β weight shows that the selected thresholds produce fewer predictions at the upper-left and lower-right corners in Figure 10d. In Figure 10e, the threshold is adaptive and set by taking the behaviour of the user into account as shown in Equation 16. As highlighted in this plot, we obtained the best tradeoff between precision and recall with this method. Furthermore, the density shows that only few results are characterised by a low precision or a low recall. Finally, as previously demonstrated, this method is more efficient than the others since it is not required to compute the F_β score for all the possible thresholds.

Figure 10f highlights the average prediction and recall obtained with the five techniques. This plot clearly illustrates the tradeoff that occurs between precision and recall and confirms the well balanced results we obtain with the threshold A. Interestingly, if we compare the results in terms of F-score, as it is often the case in Information Retrieval, the combination of precision and recall would give similar values. Adjusting, the β score may help at evaluating the methods with a single measure, but, as it will be done in the next section, we advise to visually check the predictions in order asses the quality of the tradeoff between precision and recall.

We now give an overview of the predictions extracted from the dataset with a first order Markov chain and an adaptative threshold θ_A. In Figure 11, the blue cells correspond to the predicted representative trajectories. The green and red circles correspond to the starting and ending ZOIs respectively. The black line corresponds to the path followed by the user between the two ZOIs. We first notice that most predictions are visually accurate, in particular the one presented in Figure 11 a, b and e which are characterised by a high precision and a high recall. Some predictions, such as the ones highlighted in Figure 11 c and f are correct but at some point the user probably decided to take an alternative path for some unknown reasons. Since we are in the context of future movements, we can easily imagine that, in a live system, the result of the prediction could be used to create an incentive that would influence the behaviour of the user and directly impact the quality of the predictions. Some predictions, such as the one depicted in Figure 11 b, d and f, have the same starting and ending ZOIs. Such results can typically not be obtained with methods that solely rely on ZOIs and shortest paths to make predictions and clearly highlight the benefit

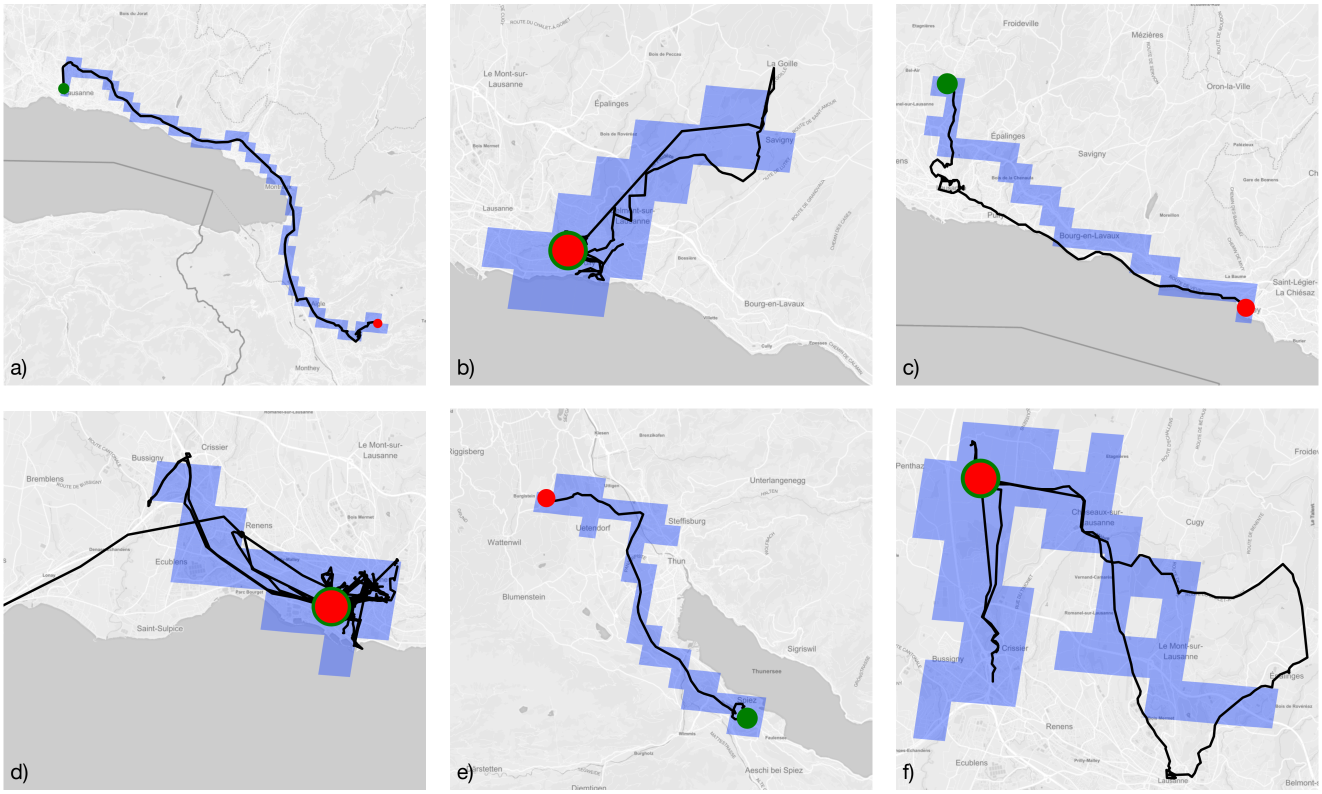


Figure 11: Predictions made with a first order Markov chain and a *A* threshold.

of using representative trajectories to predict future moves.

7. CONCLUSION

The growing ubiquity of mobile devices equipped with location aware services is opening the opportunities for novel applications. It is becoming possible to predict human mobility on a large scale today, which can be utilised to answer predictive queries. This has put forth some challenges, which the current prediction and indexing techniques are not well adapted to solve. Through this paper, we introduce an architecture that is capable to predict mobility over long time horizons, index the predicted trajectory and answer the relevant queries. Predicting mobility at distant futures allows to devise applications involving high level planning and management. To facilitate this, we present a novel spatial object, 'Representative Trajectory', that accounts for user movements within their ZOIs. Further, we propose means to empirically adapt the extraction of such objects depending on user mobility behaviour. The achieved results over real world mobility traces corroborates our solution, which achieves more than 70 % correct predictions with the best suited extraction method. Our indexing technique, based on inverted indexing, scales with the number of users unlike the tree structures proposed by existing works for predictive indexing. More importantly, we highlight the limits of mobility models, that solely rely on frequently visited places in the context of distant future predictions. Our analysis also shows that the trajectories taken in practice are often complex and as such, the user behaviour has to be taken into account for prediction over distant future. This justifies the requirement for such a spatial object and the indexing technique in order to improve the quality of distant future predictions.

8. FUTURE WORK

Our solution architecture involves several layers, each of which may be enhanced for further developments. For clarity, our initial model considered the problem in its simplest form and our future research will foster improvements. Sets of cells are used to construct representative trajectories and some useful notions may enrich them in order to predict the future movements with a higher accuracy:

1. The model initially relies on a clustering algorithm to find ZOIs. Current techniques often merge overlapping visited places to estimate them, and the resulting areas can be relatively large. Among other possibilities, adding a notion of time at this level may help at discovering ZOIs with fine granularities and thus avoiding some undesirable merges.

2. The model uses Markov chain and its transition probabilities for formulating predictions. In the future, preserving some notion of time, as well as the number of occurrences of a given cell, in the representative trajectories may help at computing these probabilities more accurately and thus making better predictions.

3. The model uses relatively large overlapping cells of one square kilometre to compute representative trajectories. We observed that when reducing the size of the cells, the model starts suffering from the lack of precision, introduced by tracking devices. Since the discretisation of space with a grid is really close from what occurs when a vector image goes through rasterisation, techniques coming from this field, such as anti-aliasing, may help at improving the granularity of the predictions.

Such additions to the model may compromise the system in terms of scalability and is bound to an exhaustive performance analysis. We observed that, the size of our dataset and other publicly available datasets such as GeoLife [21] fits in the memory and are therefore not sufficiently large enough to produce an in-depth quantitative analysis as well as relevant performance measures. A possible solution would be to generate a large synthetic dataset. While such synthetic datasets may be satisfying for performance and scalability measures, they can hardly grasp the complex nature of the human behaviours required for a quantitative analysis. A solution may be to produce synthetic traces based on the mobility models of real users with the aim to reproduce the complex behaviours. Consequently, before investigating these improvements, our priority is to find a suitable way to validate our current findings.

9. REFERENCES

[1] S. Chen, C. S. Jensen, and D. Lin. A benchmark for evaluating moving object indexes. *Proceedings of the VLDB Endowment*, 1(2):1574–1585, 2008.

[2] M.-F. Chiang, W.-Y. Zhu, W.-C. Peng, and S. Y. Philip. Distant-time location prediction in low-sampling-rate trajectories. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 117–126. IEEE, 2013.

[3] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 34–41. ACM, 2010.

[4] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Towards temporal mobility markov chains. In *1st International Workshop on Dynamicity Collocated with OPODIS 2011, Toulouse, France*, pages 2–pages, 2011.

[5] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.

[6] A. M. Hendawi, M. Ali, and M. F. Mokbel. A framework for spatial predictive query processing and visualization. In *2015 16th IEEE International Conference on Mobile Data Management*, volume 1, pages 327–330. IEEE, 2015.

[7] A. M. Hendawi, J. Bao, M. F. Mokbel, and M. Ali. Predictive tree: An efficient index for predictive queries on road networks. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1215–1226. IEEE, 2015.

[8] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *2008 IEEE 24th International Conference on Data Engineering*, pages 70–79. Ieee, 2008.

[9] D.-O. Kim, K.-J. Lee, D.-S. Hong, and K.-J. Han. *An Efficient Indexing Technique for Location Prediction of Moving Objects*, pages 1–9. Springer, 2007.

[10] 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 *Pervasive Computing*, 2012.

[11] J. J. LaViola. Double exponential smoothing: An alternative to kalman filter-based predictive tracking. EGVE '2003, pages 199–206.

[12] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson. Approaching the limit of predictability in human mobility. *Scientific reports*, 3, 2013.

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

[14] J. M. Patel, Y. Chen, and V. P. Chakka. Stripes: an efficient index for predicted trajectories. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 635–646. ACM, 2004.

[15] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. *Indexing the positions of continuously moving objects*, volume 29. ACM, 2000.

[16] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 611–622. ACM, 2004.

[17] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 611–622. ACM, 2004.

[18] Y. Tao, D. Papadias, and J. Sun. The tpr*-tree: an optimized spatio-temporal access method for predictive queries. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 790–801. VLDB Endowment, 2003.

[19] Y. Yanagisawa. Predictive indexing for position data of moving objects in the real world. In *Transactions on Computational Science VI*, pages 77–94. Springer, 2009.

[20] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM, 2011.

[21] X. X. W.-Y. M. Q. L. Yu Zheng, Hao Fu. *Geolife GPS trajectory dataset - User Guide*, July 2011.