Inferring Distant-Time Location in Low-Sampling-Rate Trajectories

Meng-Fen Chiang\textsuperscript{\textcopyright}, Yung-Hsiang Lin\textsuperscript{\textcopyright}, Wen-Chih Peng\textsuperscript{\textcopyright} and Philip S. Yu \textsuperscript{\textcopyright}
\textsuperscript{\textcopyright}Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
\textsuperscript{\textcopyright}Department of Computer Science, University of Illinois at Chicago, Chicago, Illinois, USA
\{mfchiang.cs95g, shiang1095.cs00g, wcpeng\}@nctu.edu.tw, \textsuperscript{\textcopyright}psyu@cs.uic.edu

ABSTRACT
With the growth of location-based services and social services, low-sampling-rate trajectories from check-in data or photos with geotag information becomes ubiquitous. In general, most detailed moving information in low-sampling-rate trajectories are lost. Prior works have elaborated on distant-time location prediction in high-sampling-rate trajectories. However, existing prediction models are pattern-based and thus not applicable due to the sparsity of data points in low-sampling-rate trajectories. To address the sparsity in low-sampling-rate trajectories, we develop a Reachability-based prediction model on Time-constrained Mobility Graph (RTMG) to predict locations for distant-time queries. Specifically, we design an adaptive temporal exploration approach to extract effective supporting trajectories that are temporally close to the query time. Based on the supporting trajectories, a Time-constrained mobility Graph (TG) is constructed to capture mobility information at the given query time. In light of TG, we further derive the reachability probabilities among locations in TG. Thus, a location with maximum reachability from the current location among all possible locations in supporting trajectories is considered as the prediction result. To efficiently process queries, we proposed the index structure Sorted Interval-Tree (SOIT) to organize location records. Extensive experiments with real data demonstrated the effectiveness and efficiency of RTMG. First, RTMG with adaptive temporal exploration significantly outperforms the existing pattern-based prediction model HPM [2] over varying data sparsity in terms of higher accuracy and higher coverage. Also, the proposed index structure SOIT can efficiently speedup RTMG in large-scale trajectory dataset. In the future, we could extend RTMG by considering more factors (e.g., staying durations in locations, application usages in smart phones) to further improve the prediction accuracy.

Categories and Subject Descriptors
H.3.1 [Information Storage and Retrieval]: H.5.2 [Information Interfaces and Presentation]

General Terms
Algorithms, Design, Human Factors

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 are not made or distributed for profit or commercial advantage and 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 re-publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

1. INTRODUCTION
With the growth of location-aware technologies and location based Internet services (e.g., Foursquare and Places on Facebook), tracking or collecting a huge amount of trajectories of users becomes feasible. Given a set of trajectories, prior work in [2] has formulated a distant-time query, where given a query time, the current location and time, an estimate location of moving objects at the query time is returned. The distant-time query is useful in many applications, such as content-based delivery networks, inferring regions for tourism recommendations, and estimating the traffic status for transportation management [7].

Figure 1: Framework Overview

Time-ordered check-in records of a user becomes ubiquitous as users could easily perform check-in services (e.g., Foursquare) to note their locations with a mobile phone or people can share geotagged photos whose time-stamps and geo-locations on a photo sharing website (e.g., Flickr). Without loss of generality, the time-ordered check-in records of a user are able to be expressed as low-sampling-rate trajectories, where details of movement information are lost [8]. A considerable amount of efforts has been devoted to design location prediction models [2, 5, 3]. For example, the authors in [5] proposed a location prediction model to infers next location of a user based on collective frequent patterns discovered from previous trajectories of all users. However, [5] fails to predict distant-time future locations. A hybrid prediction model (HPM) in [2] partially address the problem of answering distant-time future locations. HPM relies on frequent moving patterns discovered from past trajectories as well as existing motion functions using the objects recent movements to support future location queries. While pattern-based prediction models over high-sampling-rate trajectory databases show promising query results, it fails to effectively pre-
dict distant-time location queries over low-sampling-rate trajectories in terms of both coverage and accuracy because HPM can fail to discover frequent moving patterns due to data sparsity.

In this paper, we address the sparsity issue in low-sampling-rate trajectories for distant-time query. Specifically, given current location at the current time point and a query time, we aim to predict the location of a user at query time. We present a Reachability-based prediction model on Time-constrained Mobility Graph (abbreviated as RTMG) that investigates user’s reachability and determines the possible candidate locations. Given a query, the core components in RTMG are as follows:

- **Adaptive Supporting Trajectory Retrieval**: By expanding investigation time interval between current time and query time, we can infer paths from region \( A \) to other regions within the investigation time interval. These trajectories are called supporting trajectories.

- **Time-Constrained Mobility Graph**: Based on the supporting trajectories, a Time-constrained mobility Graph (abbreviated as TG) that captures a user’s moving behavior within a time interval is constructed.

- **Reachability Probabilities**: In light of TG, we derive reachability probabilities of vertexes (i.e., the locations) in TG and thus determine the most likely location at the query time.

2. FRAMEWORK OVERVIEW

RTMG consists of two phases, off-line trajectory pre-processing and on-line location prediction as shown in Figure 1. The following subsections briefly describe how each module works.

2.1 Off-Line: Trajectory Pre-processing

Derive Trajectory Snapshots: In off-line phase, we derive a sequence of trajectory snapshots from raw trajectories to better capture the spatial and temporal correlations. Formally, given a trajectory database \( T^d \) of equal time interval (e.g., a day) and a time cell size \( \delta_t \), a sequence of trajectory snapshots \( \mathcal{S}(\delta_t, T) = \{S_1, \ldots, S_n\} \) is obtained by partitioning the trajectory database into equal size \( \delta_t = d \) and transforming location records into region symbols \( \mathcal{L} \) discovered from original location records. Figure 3 illustrates an example of trajectory snapshot in daily scale and the location information of records is represented by region identifications. For example, trajectory \( T_1 \) consists of six location records \( \{g_1, 1, g_1, 2, \ldots, g_1, 6\} \). Snapshot \( S_2 \) consists of two regions \( C \) and \( D \).

![Figure 3: Trajectory database](image)

Data-centric Index Structure: To improve efficiency of query processing, we design an index structure, Sorted Interval-Tree (SOIT), to structure user mobilities according to their time locality in a data-centric balanced tree. Several operators are defined to efficiently retrieve supporting trajectories and infer time-constrained mobility network on-the-fly. SOIT indexes a set of location records into a balanced tree that each leaf cell time contains similar amount of data by partitioning a timeline into a sequence of time cells and maintaining a set of location records in each time cell no more than the size of \( b \), where \( b \) is the branching factor. Figure 4 illustrates a set of location records indexed by SOIT with a branching factor of three. Centered at \( Q. c.t = 5am \), the partitions that overlap with the time point is \( N_{12} \), which consists of three location records, one locates at location \( C \) and the other two locates at location \( D \).

Each leaf time cell of SOIT is associated with a group of inverted files, where each inverted file stores a group of location records with their time-stamps covered by the leaf time cell. Each record in an inverted file that is covered by time cell \( N \) contains four entries: (1) \( N \) : location record covered by \( N \), (2) \( T \) : location record immediately after \( N \), (3) \( T \) : travel time between the end time of \( N \) and the start time of \( T \) and (4) \( SD \) : total time that a user stayed during \( T \). All leaf time cells are sorted according their start time in ascending order and are connected into a list with sibling link for efficient query processing.

![Figure 4: Indexing scheme and query processing](image)

2.2 On-Line: Location Prediction

Adaptive Supporting Trajectory Retrieval: Adaptive temporal exploration aims to dynamically determine the time interval for a query based on the temporal correlation between the query and current set of supporting trajectories. We invoke adaptive temporal exploration if we do not have sufficient and high quality supporting trajectories to develop the prediction model for a given query. Specifically, if we do not have sufficient supporting trajectories in the desired time interval, we broaden the time interval with the guidance of temporal correlation between the query and current set of supporting trajectories. Otherwise, we accomplish the extraction of supporting trajectories in the desired time interval. If the entire timeline is investigated, essentially the whole set of trajectories is used to provide more information, and thus may be more useful.

When the entire timeline is investigated (i.e., full exploration with \( k = \pi \)), essentially the entire set of trajectories will be considered as the set of supporting trajectories, where \( n \) is the number of snapshots.

Time-Constrained Mobility Graph: Inspired from the previous work [3], we model user mobility behavior as a Time-constrained mobility graph.
ability in distant-time location prediction, we use the metric, reach-
located structurally close to each other in a time-constrained mo-
dicator of closeness of the node pair. Some node pairs that are
probability of connectivity between node pairs is an important in-
than probabilities of single immediate transition or single path, the
mobility statistics of a single transition or a single path derived from
trajectories are very sparse and making prediction merely based on sparse

Definition (Reachability) Let \( A \) be the \(|V| \times |V|\) transition prob-
ability matrix of a time-constrained mobility graph \( G^Q \). Given
the restart probability \( c \in (0,1) \), the reachability from \( Q, cl \) to any
node \( v \in V \) is denoted as a vector \( RCH_{Q, cl}^v \). \( RCH_{Q, cl}^v \) can be derived by

\[
RCH_{Q, cl}^v = cE_{Q, cl} + (1 - c)RCH_{Q, cl}^{v-1}A
\]

when \( RCH_{Q, cl} \) is converged, where \( E_{Q, cl} \) is a vector, the entry
representing \( Q, cl \) is one and the rest entries are set to be zero.

Given a query \( Q \) and its TG \( G^Q \), we propose to compute the
reachability between \( Q, cl \) and \( v \in V \) in \( G^Q \) as a metric to predict
the user’s location at query time \( Q, qt \).

3. DEMONSTRATION PLAN

3.1 Demonstration Settings

We utilize Gowalla dataset to verify our prediction model. The
dataset contains 50 users, 113 decomposed trajectories in daily-
scale, and 30 distinct user-specific regions on average. The aver-
age time interval between consecutive location records is approxi-
ately 17 days. The regions were discovered by OPTICS \([1]\) with \( c \) set to be 100 meters and \( MinPts \) set to be three.

3.2 Demonstration Scenarios

From this interactive interface \(^1\), users can format several queries
to investigate particular user’s movement behavior and visualize
future locations at query time. We describe each of them as follows:
Show frequently visited regions: To visualize user’s frequently
visited regions, we can format a query by selecting a user ID, each
frequently visited region will be expressed as a red-colored rectan-
gle shown on the map. The set of regions are derived by OPTICS
algorithm which is widely used to mines dense regions from a se-
quence of location points. For example, Figure 2a shows the set of
frequently visited regions of the user (ID=48537).

Show time-constrained regions: To visualize frequently visited
regions constrained at a timepoint of interest, we can format a query
by selecting a user ID and the timepoint of interest. The set of
regions visited at specified timepoint will be displayed as purple re-
regions. This helps us to investigate the correlation between time and
locations. For example, Figure 2b illustrates two purple regions
timepoint of interest, we can format a query

\(^1\)http://carweb.cs.nctu.edu.tw/ shiang/DTLP2/index.html
Following this, we implement two functionalities to visualize location prediction results as shown in Figure 5.

**Future location at specified query time and its time-constrained mobility graph:** Given the user ID, a current location, a current time, and a query time, this functionality shows the top-3 future locations with maximum reachabilities at specified query time, where a region attached with a white label $CL$ is the user’s current location, a yellow tag with a ranking number is attached to returned locations and the actual location at the query time (i.e., ground truth) is displayed as a green region. For example, given user ID to 48537, current time 12pm at region 0, specifying query time at 5pm and pressing prediction button, the system returns top-3 candidate locations at 5pm with a yellow tag indicating their rank as illustrated in Figure 5a. In this case, the actual location at 5pm (expressed as green region) is also the top-1 predicted location.

Additionally, we also illustrate the time-constrained mobility graph connecting historical traversed regions between the current time and the query time, where nodes represent regions and edges represent the transition probabilities between two regions. As illustrated in Figure 5a, it shows the mobility graph with four nodes and three edges, where the transition probability from one region to another is recorded in INFO icon. For example, the INFO shows that the transition probability from region 10 to region 2 is 0.074.

**Future location at all possible query time:** Given the user ID, a current location, a current time, this functionality shows all future locations at all possible query time. As shown in 5b, given user ID to 48537, current time 12pm at region 0, selecting all cases at step 4 and pressing prediction button, the system will return all possible locations at varying query times. In this case, two possible locations (region 0 at 2pm and region 11 at 12pm) are returned and displayed in purple-colored rectangles on the map.

### 4. FURTHER DISCUSSION

To address the sparsity in low-sampling-rate trajectories, we develop a Reachability-based prediction model on Time-constrained Mobility Graph (RTMG) to predict locations for distant-time queries. The prototype of Reachability-based prediction model we implemented can be applied to many different scenarios, including urban planning[4], location recommendation or even location-based content delivery networks (e.g., coupons) [6] by incorporating user’s future location as a feature to intelligently deliver spatio-temporal sensitive information.

### 5. REFERENCES


