Incremental Frequent Route Based Trajectory Prediction

Anja Bachmann
Karlsruhe Inst. of Technology
bachmann@teco.edu

Christian Borgelt
EU Centre for Soft Computing
christian@borgelt.net

Győző Gidófalvi
KTH Royal Inst. of Technology
gyozo.gidofalvi@abe.kth.se

ABSTRACT

Recent technological trends enable modern traffic prediction and management systems in which the analysis and prediction of movements of objects is essential. To this extent the present paper proposes IncCCFR—a novel, incremental approach for managing, mining, and predicting the incrementally evolving trajectories of moving objects. In addition to reduced mining and storage costs, a key advantage of the incremental approach is its ability to combine multiple temporally relevant mining results from the past to capture temporal and periodic regularities in movement. The approach and its variants are empirically evaluated on a large real-world data set of moving object trajectories, originating from a fleet of taxis, illustrating that detailed closed frequent routes can be efficiently discovered and used for prediction.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]:
Data mining, Spatial Databases and GIS

General Terms

Algorithms

Keywords

Spatio-Temporal Data Mining, Frequent Routes, Incremental Mining, Time Inhomogeneous Trajectory Prediction

1. INTRODUCTION

The rapid growth of demand for transportation, and high levels of car dependency caused by the urban sprawl, exceeds the slow increments in transportation infrastructure supply in many areas. This causes severe traffic congestion. In dense urban areas expanding the road network is not a sustainable solution. A more viable approach is to monitor traffic congestion, understand the causes of its formation and development, and use this knowledge in traffic management and transportation planning to mitigate the traffic congestion. Early systems for traffic prediction and management have primarily used punctuated speed and flow measurements from fixed location sensors in conjunction with traffic models to tackle the prediction and management tasks. More recently, the widespread adoption of GPS-based onboard navigation systems and location-aware mobile devices have enabled radically new possibilities.

Such systems commonly use the trajectories of the moving objects [9] as follows: vehicles periodically submit their location (and speed) to a central server, which extracts traffic/mobility patterns from the submitted information. These patterns, together with the current locations (and speeds) of the vehicles are both used in short- and long-term traffic prediction, management and planning tasks. Additionally, the current and near-future traffic conditions are sent in real-time to the vehicles likely to be affected.

Although modeling, managing, mining and predicting moving object trajectories received considerable research attention in recent years and significant contributions have been made in these areas (mostly separately), given the system and application scenarios outlined above a number of challenges remain: (1) More complex and more powerful sequential pattern based trajectory prediction approaches are difficult to adopt to capture the vital temporal and periodic variation in patterns [10]. (2) Trajectory prediction systems model and provide knowledge about the movement of the objects at a fixed level of detail, while different applications (real-time management vs. long-term planning) need different levels of detail. (3) Existing systems tend to base their predictions on either historical or current information while arguably both types of information are relevant. (4) To the best knowledge of the authors, no end-to-end system exists that provides an effective and application-relevant sliding window based stream processing framework for the management, *incremental* mining and accurate prediction of continuously evolving trajectories of moving objects.

To address the above challenges, we propose a solution that models the continuous movement of an object in space as a time-stamped, continuously evolving sequence of traversed grid cells along with respective traversal times. The system receives the continuously evolving trajectories for a set of moving objects as a single stream. Using a temporal sliding window model, two stream processing tasks are then performed simultaneously: (1) for each slide of the window the system *incrementally* mines and stores the frequent routes and the neighboring cell probabilities from the *completed* object trajectories of the window, and (2) using the current *partial* trajectories of objects and a *combination of*

ACM SIGSPATIAL IWCTS'13, Nov 05–08 2013, Orlando, FL, USA

relevant sets of historical frequent routes and neighboring cell probabilities that capture the vital temporal and periodic variation in movement [11] (e.g., Mondays between 8am and 9am), the system predicts the *near-future locations of moving objects* on the grid. Essentially, the proposed solution uses a prediction model that is a time inhomogeneous, varying order, deterministic Markov model based on frequent routes and neighboring cell probabilities.

The rest of the paper is structured as follows: Section 2 describes related work. Section 3 formalizes important concepts and the problem statement. Section 4 details the approach taken in the prediction model. Section 5 empirically evaluates the proposed methods on a real-world data set. Finally, Section 6 concludes and points to future work.

2. RELATED WORK

The following paragraphs discuss related work in frequent pattern and trajectory mining and prediction.

Frequent Pattern Mining: The concept of mining association rules evolved over the last 20 years. First applied to item sets [1], then extended to sequential patterns [2], afterwards improved regarding efficiency [20]. Later on, the concept of closed patterns was introduced, including mining closed patterns from data streams [14]. In contrast to these approaches, the current paper considers the specifics of incremental trajectory mining and proposes an algorithm to efficiently extract closed frequent routes from non-overlapping windows of the stream of trajectories. The core idea is inspired by an incremental approach to graph mining [5]. Unfortunately the original approach is not applicable to the stated applications without a modification (due to a flaw), but the general idea is adaptable to the present problem.

Trajectory Mining and Prediction: Considerable research has been conducted to extract and use the regularities in object movement to predict future movement of objects.[1]

Two popular extraction and prediction methods emerged: discrete-time Markov model based [3, 4, 11, 13, 16] and sequential rule/trajectory pattern based [7, 10, 12, 19, 22–24]. The methods can also be classified based on what information is used to model the movement of objects into methods with (1) a general model for all objects [7, 10, 12, 13, 16, 19, 22], (2) a type-based model for similar (types of) objects [3,24], or (3) a specific model for each individual object or set of individual objects [4,11,16,23]. Alternatively, they may be classified according to their definition of Regions Of Interest (ROIs) for prediction and consequently their spatial and temporal scale and granularity into methods using (1) application-specific ROIs (road segment, network cell, sensors etc.) [3,10,13,16,19], (2) density-based ROIs [4,7,11, 12, 22–24], or (3) grid-based ROIs [7, 11, 19, 22]. Finally and most importantly, they can be classified according to their prediction provision into methods that provide (1) only sequential spatial predictions (location of next ROI) [3, 4, 16, 24] or (2) spatio-temporal predictions [7,10–13,19,22,23] and into methods that provide (1) time-continuous [10,12,13,16] or (2) time-punctuated [7, 11, 19, 22, 23] predictions. Other prominent approaches are [15, 17], in which predictions rely on the assumption that the observed, short-term, partial trajectory of an object is part of a (approximately [17]) shortest path to the future unknown destination of the object.

In comparison, our method (1) extracts both neighboring cell probabilities/turn statistics [13, 16] and frequent routes, (2) mines and stores only a lossless compression of the patterns (*closed* patterns) unlike most trajectory pattern based approaches (exceptions are [10, 23]), (3) restricts patterns to spatio-temporally *contiguous* ones, thus allowing spatio-temporal, time-continuous predictions, (4) adopts an incremental mining framework [5] that allows us to combine non-overlapping, temporally-relevant mining results, (5) without loss of generality, constructs a general model for all objects, (6) adopts the grid-based ROIs approach, which allows to represent, mine and predict trajectories at varying levels of detail, and lastly (7) does not assume that the objects follow a shortest path [15,17] or that the objects follow a particular movement model between ROIs [12].

[1]The following is a summary of a recent classification [11] of such methods with only some of the most relevant references.

3. PRELIMINARIES AND DEFINITIONS

Grid Based Routes of Moving Objects: Let $O = \{o_1, \ldots, o_M\}$ be a set of moving objects. Let the time domain be denoted by $\mathbb{T} \equiv \mathbb{N}_0$. Let $\mathcal{G}$ denote a *grid* with grid cells $g_1, g_2, \ldots$ with side length *glen* that uniformly partition the 2D Euclidean space . Then, the spatio-temporally scalable, *grid based movement model* describes the movement of an object on the grid with a *grid based trajectory* as follows.

Definition 1. The *grid based trajectory* of a moving object $o \in O$ is a pair $tr^o_{grid} = (ts, s_{grid})$, where $ts \in \mathbb{T}$ is the start time of the trajectory and $s_{grid} = \langle (g_1, \Delta t_1), \ldots, (g_m, \Delta t_m) \rangle$ is a *temporally annotated sequence*, i.e., a sequence of pairs of traversed grid cells $g_i \in \mathcal{G}$ and associated *traversal times* Δt_i, where Δt_i is the time it took o to traverse grid cell g_i. (In the following we omit 'grid based' and subscript 'grid'.)

Trip Trajectories of Moving Objects: Pauses in movement, which can either be explicitly signaled by the object or can be automatically inferred by spatio-temporal analysis of the trajectory, naturally subdivide the trajectory tr^o of an object $o \in O$ into a sequence of *trip trajectories* $\langle tr^o[1], \ldots, tr^o[t] \rangle$. A trip trajectory $tr^o[i]$ is modeled in the same way as an object trajectory and the term 'trip' is omitted when it is clear from the context.

Continuously Evolving Trajectories: In an online setting, as an object $o \in O$ moves, its trip trajectory $tr^o[t]$ is *evolving*, i.e., it is continuously extended at the end. A single extension of $tr^o[t]$ is referred to as a *trajectory piece*. The i-th trajectory piece is denoted by $tp^o_i[t]$ and is modeled in the same way as an object trajectory. As a trajectory piece $tp^o_i[t] = (ts_i, (g_i, \Delta t_i))$ can only be formed after the object o has completely traversed the grid cell g_i, the trajectory piece $tp^o_i[t]$ is implicitly associated with an *arrival time* $t\_arr = ts_i + \Delta t_i$. A sequence of trajectory pieces $\langle tp^o_i[t], \ldots tp^o_k[t] \rangle$ of a trip trajectory $tr^o[t]$ of object o for trip t form a *contiguous trip sub-trajectory* of object o for trip t if $\forall j$ such that $i \leq j < k$, $ts_j + \Delta t_j = ts_{j+1}$. Given a time period $t = [t_s, t_e]$ a trip trajectory that temporally intersects t and *ends* in t is called a *completed trip trajectory*. The contiguous trip sub-trajectory that is formed by the intersection of the period t and a trip trajectory that *does not end* in t is called a *partial trip trajectory*.

Frequent Route Mining: The frequent route mining of grid based trajectories is formulated identically to [10] as follows: Let $TR = \{tr_1, \ldots, tr_T\}$ be a set of trip trajectories in which tr_i represents a particular trip trajectory $tr^{o_j}[t]$ of object $o_j \in O$ for trip t. A trajectory $tr_i = (ts_i, s_i) \in TR$ *contiguously supports* a *route* r (a temporally annotated sequence), or r is a *contiguous sub-sequence* of s_i, equivalently denoted as $r \preceq_c tr_i^o$ and $r \preceq_c s_i$, respectively, iff there exists a *contiguous* index sequence $1 \leq i_1 < \ldots < i_l \leq m$ such that $\forall j; 1 \leq j < l\colon i_{j+1} - i_j = 1$ and $\forall j; 1 \leq j \leq l\colon g'_j = g_{i_j}$. We call the number of trajectories in TR that contiguously support a route r the *support* of r, $\text{supp}(r) = |\{tr \in TR \mid r \preceq_c tr\}|$. The route r is a *contiguous frequent route* iff r is *contiguously supported* by at least *min_sup* trajectories, that is, if $\text{supp}(r) \geq min\_sup$. A route r_c is a *closed contiguous frequent route* or a *pattern* for short, iff $\text{supp}(r_c) \geq min\_sup$ and there exists no contiguous frequent *extended* route r_e such that r_c is a proper subsequence of r_e, i.e., $r_c \prec r_e$, and $\text{supp}(r_c) = \text{supp}(r_e)$. For simplicity, but without loss of generality, we define the temporal annotation of the route, i.e., the traversal time of a given grid cell of the route, as the *global* average of the traversal times of the corresponding grid cell in the trajectories that support the route consisting of the single grid cell. Consequently, the constrained frequent route mining task is defined as follows:

Definition 2. Closed Contiguous Frequent Route Mining: Given a set of objects O, a set of their trip trajectories TR, and a minimum support threshold *min_sup*, find the set of closed contiguous frequent routes, *CCFR*, in TR.

Given their continuously evolving nature, trip trajectories of objects $o \in O$ are not observed as a finite *set*, but rather as a *continuous Stream of time-stamped trip Trajectory Pieces of objects*, denoted as $\mathcal{STP}$. Formally, an $\mathcal{STP}$ is an unbounded ordered sequence $\langle e_1, e_2, \ldots \rangle$ of elements, each of which is a triplet $e_i = (o_i, tp_i, t\_arr_i)$ in which tp_i represents a particular trajectory piece of object o_i (for some trip) with arrival time t_arr_i and $t\_arr_i \geq t\_arr_{i-1}$ for $i > 1$. The online processing of $\mathcal{STP}$ is facilitated by adopting a commonly used temporal sliding window model for streams:

Definition 3. Temporal Sliding Window Model: Given a stream of ordered time-stamped elements, $\mathcal{S} = \langle (e_1, t_1), (e_2, t_2), \ldots \rangle$, and temporal sliding window parameters, *window size*, $t_{wsize} \in \mathbb{N}$ and *window stride*, $t_{wstride} \in \mathbb{N}$ the Temporal Sliding Window Model (TSWM) at every *window slide* time instance, $t_{slide} = a \times t_{wstride} + t_{wsize}$ where $a \in \mathbb{N}^0$ processes (depending on the task: mine/predict) the completed/partial trip trajectories w.r.t. the time interval of the *window* $(t_{slide} - t_{wsize}, t_{slide}]$. Consequently, a TSWM is defined by the pair $SW = (t_{wsize}, t_{wstride})$.

Then, the task of mining *CCFR* online is defined as follows:

Definition 4. Online Closed Contiguous Frequent Route Mining: Given a stream of trajectory pieces $\mathcal{STP}$ of objects O, TSWM parameters SW, and minimum support threshold *min_sup*, for each window slide, or *current time* instance t_c, find *CCFR* in the completed trip trajectories.

Moving Object Location Prediction: The motivation for mining *CCFR*s is to utilize a relevant subset of extracted historical patterns in the fundamental task of predicting the near-future location of an object on the grid given its current partial trip trajectory. This task is defined as follows:

Definition 5. Moving Object Location Prediction: Given a grid $\mathcal{G}$, multiple sets of closed contiguous frequent routes $CCFR_1, \ldots, CCFR_k$ mined from a set of historical, non-overlapping windows $w_1, \ldots, w_k$, and the partial trip trajectory $\langle tp_i^o[t], \ldots tp_k^o[t] \rangle$ of object $o \in O$ for its current trip t up to the current time t_c, predict the grid cell $\widehat{g^o_{(t_p)}} \in \mathcal{G}$ that o will be located in at *prediction time* $t_p \geq t_c$ such that the Euclidean distance between the centers of $\widehat{g^o_{(t_p)}}$ and the actual grid cell $g^o_{(t_p)}$ that o is located in at t_p, or the final grid cell $g^o_\times$ of t (whichever occurs first), denoted as $dist(\widehat{g^o_{(t_p)}}, g^o_{(t_p),\times})$, is minimized.

Adopting the introduced TSWM for streams the online version of the location prediction task is defined as follows:

Definition 6. Online Moving Object Location Prediction: Given a grid $\mathcal{G}$, a stream of trajectory pieces $\mathcal{STP}$ of objects O, TSWM parameters SW, and a prediction time horizon Δt_p, for each window slide, or *current time* instance t_c, using *any* subset of the sets of CCFRs mined from a set of historical, non-overlapping windows, predict the future grid cell $\widehat{g^o_{(t_p)}}$ at the prediction time $t_p = t_c + \Delta t_p$ of every object $o \in O$ that has a partial trip trajectory in the current window, such that $\sum_{o \in O} dist(\widehat{g^o_{(t_p)}}, g^o_{(t_p),\times})$ is minimized.

4. THE IncCCFR APPROACH

The knowledge about the past movements of objects, in the form of CCFRs, form the tenet of the proposed model that aims to predict *near-future locations of moving objects* on the grid. Induced from its objective to incrementally mine CCFRs, the algorithm is named IncCCFR.

CCFR Mining: CCFR mining works by growing CCFRs (or patterns) in a depth-first fashion. The search commences with single grid cell patterns which are recursively *extended* by appending one cell in each recursion step. As a data structure, a simple flat array representation of the trajectories is used, into which references are kept to the current ends of the pattern occurrences in order to be able to quickly find and group possible extensions. The most demanding part of CCFR mining is the *closedness* check. In principle, there are two strategies: (1) Use a repository of already found (closed) frequent patterns and check whether there exists a superpattern in the repository that has the same support. If this is the case, the pattern is not closed. (2) Use a direct check of possible superpatterns and their support by generating and testing all possible extensions of a given pattern. Here, the latter was adopted, because it is simpler and faster in the case of gapless patterns, which are considered in this paper.

In order to model the stopping of objects in the extracted CCFRs, every grid cell is associated with a corresponding pseudo grid cell ('stop') and, prior to mining completed trajectories, every completed trajectory is extended by this pseudo grid cell after its last (real) grid cell.

CCFR-Based Prediction: Our prediction model relies on the notion of a *query vector* q (partial trajectory of an object) that ends in the *anchor* a (most recently traversed grid cell in q). The prediction iteratively extends q one grid cell at a time within the prediction horizon Δt_p as follows (R is the set of all contiguous closed frequent patterns):

1. Retrieve the set $R^* \subseteq R$ of patterns *best matching* the query, that is, all $r \in R^*$ contain the longest contiguous

suffix s of q occurring in any pattern $r \in R$. The support of s is $\text{supp}(s) = \max_{r \in R^*,\, s \preceq r} \text{supp}(r)$ (this holds, because we mined closed frequent patterns).

2. Let $C^* = \{c \in G \mid \exists r \in R^* : s\&c \preceq r\}$, where & denotes concatenation, be the set of grid cells that occur in the patterns in R^* directly after an occurrence of s. $\forall c \in C^* : \text{supp}(s\&c) = \max_{r \in R^*, s\&c \preceq r} \text{supp}(r)$.
3. $\forall c \in C^*$ compute the *successor probability* $p(c|s) = \text{supp}(s\&c)/\,\text{supp}(s)$ (i.e., confidence of the rule $s \rightarrow c$).
4. Retrieve the set $C = \{c \in G \mid \exists tr \in TR : a\&c \preceq tr\}$, that is, the set of grid cells occurring in the trips after the anchor a. Let $p'(c|a) = \text{supp}(a\&c)/\,\text{supp}(a)$ denote the *neighbor probability* of c given a. (These neighbor probabilities are mined in parallel with the patterns.)
5. $\forall c \in C - C^*$ let $p(c|s) = (p'(c|a)/\sum_{c \in C - C^*} p'(c|a)) \cdot (1 - \sum_{c \in C^*} p(c|s))$. This completes the successor probability distribution p over the neighbors of a.
6. Predict $c^* = \text{argmax}_{c \in C}\, p(c|s)$ as the most likely successor grid cell, extend q with it (that is, set the query $q := q\&c^*$) and reduce the remaining prediction horizon by the global average traversal time of c^*.
7. If the remaining prediction horizon is ≤ 0, stop (and return c^* as the prediction); otherwise go to step 1.

Essentially, the extension of q (the predicted trajectory) is a contiguous sequence of most likely grid cells and their predicted traversal times, resulting from interleaving choices based on CCFRs and neighboring cell probabilities. As the choices based on neighboring cell probabilities completely ignore the past movement of the object, they can easily result in predictions that are relatively unlikely, namely U-turns (when c^* is the grid cell preceding a or one of the two neighbors of a that have a Manhattan of 1 to a) and cycles (when the trajectory contains a large cycle, i.e., passes through the same cell twice). To avoid these two situations a U-turn prevention and a cycle prevention may be incorporated.

Combining Results from Different Windows: In order to combine non-overlapping sets of closed patterns the general idea of the approach for incremental frequent subgraph mining suggested in [5] is exploited. In principle, this approach is applicable, because the sequences considered here can be seen as restricted graphs (no branchings, no cycles). It relies on the idea to compute a "relative support" for the (closed) frequent patterns, such that interpreting the patterns as transactions (with the "relative support" as a transaction weight) and then mining these transactions reproduces the pattern set. That is, the combined operation of weighting the patterns and mining them is an idempotent operation. Therefore these weights are called *idempotent pattern weights* (*ipw*) rather than "relative support." The idempotent weight of a pattern r is simply its support $sup(r)$ minus the support of its proper superpatterns in the pattern set $S(r)$, that is, $ipw(r) = sup(r) - \sum_{j \in S(r)} ipw(j)$.

With such an idempotent operation, pattern sets can easily be combined: all one has to do is to compute the idempotent pattern weights for each pattern set separately, then to concatenate the weighted patterns sets, and finally to mine this concatenation (with a minimum support value that may or may not be the same as the support used to obtain the pattern sets in the first place). The result is an approximation of the pattern set that would have been obtained if the original data, from which the individual pattern sets were derived, had been pooled and mined for frequent patterns.

Note that it is necessary to define the support of a pattern as the number of all occurrences in the database sequences. That is, if a pattern occurs k times in the same sequence, this sequence contributes a count of k to the support of the pattern (and not just 1). With this support definition and for the special case of directed sequences and patterns with no gaps only, i.e., the case of CCFRs, weighting and mining the closed patterns is indeed an idempotent operation.

The adjusted form of the approach [5] is applied in the IncCCFR approach to both mine patterns incrementally and to combine multiple temporally relevant mining results from the past as follows: The stream of trajectory pieces, $\mathcal{STP}$, is traversed using a TSWM $SW = (t_{wsize}, t_{wstride})$ such that $t_{wsize} = k \times t_{wstride}$ for $1 < k \in \mathbb{N}$. Then, for each window slide w, IncCCFR (1) mines the CCFRs from the newest $t_{wstride}$-length subwindow of w, (2) computes their idempotent pattern weights, (3) stores the mined CCFRs and their idempotent pattern weights, (4) concatenates the current CCFRs with the CCFRs from the previous $k-1$ subwindows or any set of temporally relevant historical subwindows and mines their weighted combination to obtain the CCFRs for w or the temporally relevant historical subwindows. A historical subwindow is temporally relevant if for a combination of a user-defined set of temporal domain projections (hour-of-day, day-of-week, weekday-weekend, etc.) the temporal projection(s) of the historical subwindow matches that of w. Subsequently, the partial trajectories of w are predicted based on the combined mining results as described.

Idempotency: To show that the applied weighting is indeed idempotent, let x represent a data set and $m(x)$ the pattern mining applied to it. Let the support value for each pattern $r \in m(x)$ be denoted as $m(x).sup(r)$. Let $w(x)$ represent the weighting of the data set x that computes the idempotent pattern weights $ipw(r)$. Then, the algorithm is idempotent iff $m(w(x)).sup(r) = m(x).sup(r)$.

$$
\begin{aligned}
w(x).ipw(r) &= m(x).sup(r) - \textstyle\sum_{j \in S(r)} w(x).ipw(j) \\
m(w(x)).sup(r) &= w(x).ipw(r) + \textstyle\sum_{j \in S(r)} w(x).ipw(j) \\
&= (m(x).sup(r) - \textstyle\sum_{j \in S(r)} w(x).ipw(j)) \\
&\quad + \textstyle\sum_{j \in S(r)} w(x).ipw(j) \\
&= m(x).sup(r)
\end{aligned}
$$

Possible Extension of the Approach: IncCCFR models the frequent *regular* movement of objects and the long-term temporal and periodic variability of these regularities. A system that bases its predictions only on such long-term regularities is bound to make large prediction errors when facing rare and sudden changes due to largely unpredictable and non-periodic traffic events like accidents. The proposed model can be extended to react to such short-term conditions in two ways. One natural option is to use temporally decaying weights in the pattern combination scheme used for mining (and prediction) such that more weight is assigned to patterns that occurred more recently than to patterns that occurred less recently. Alternatively or additionally, one can combine historical patterns with the current traversal times of grid cells from the current window depending on how far ahead a given grid cell is in a given predictive pattern. So for example, if for a trajectory $tr = (ts, \langle (g_1, \Delta t_1), \ldots, (g_n, \Delta t_n) \rangle)$ the prediction is $p = \langle (g_1, \Delta t_1), \ldots, (g_n, \Delta t_n), \ldots, (g_m, \Delta t_m) \rangle$, then instead of predicting the location (speed) of the object as it follows

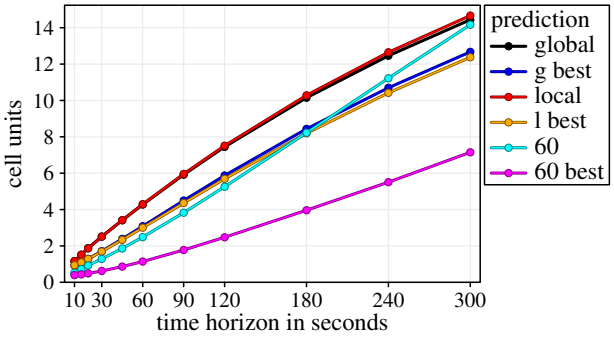

Figure 1: Absolute prediction error (i.e., average grid cell distance to the predicted and to 'best' grid cell) of different methods.

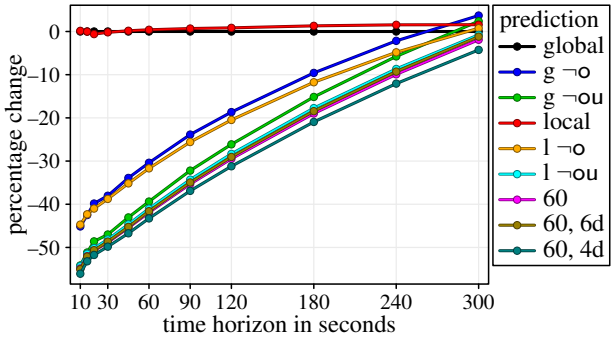

Figure 2: Relative prediction error (i.e., percentage improvement) of different methods w.r.t. the baseline predictor 'global'.

the remainder pattern $p' = \langle (g_{n+1}, \Delta t_{n+1}), \ldots, (g_m, \Delta t_m) \rangle$ only based on the historical traversal times $\Delta t_{n+1}, \ldots, \Delta t_m$, one can predict the location along p' using a (sequence) distance weighted combination of the historical and the current traversal times $\Delta t^c_{n+1}, \ldots, \Delta t^c_m$, i.e., the traversal time of the i-th grid cell in p' is predicted as $(1/i)^k * \Delta t^c_{n+i} + (1 - (1/i)^k) * \Delta t_{n+i}$ where k is a decay factor. Due to the rarity of such unpredictable events, the implementation and evaluation of this extension is left for future research.

5. EMPIRICAL EVALUATION

In this section we describe the empirical evaluation of the proposed methods. All experiments were conducted on a PC with Ubuntu 12.10 64bit and an Intel Core 2 Quad Q8400 2.66GHz processor and 4GB memory.

Real-Word Data Set: The proposed method is evaluated on a six day long (Mon, Tue, Thu, Fri, Sat, Sun) sample of the near real-time stream of raw GPS positions of around 11,000 taxis moving on the streets of Wuhan, China [18]. In this sample, positions of moving vehicles are read approximately every 20 to 60 seconds, totaling about 85 million records. The time-stamped readings include vehicle ID, location, speed and heading. After removing obvious outliers, sampling gaps of more than 120 seconds are used to identify trips in individual trajectories. To adapt the raw GPS data set to the proposed framework, two consecutive Cartesian coordinate locations within a trip are linearly interpolated by approximating the interpolating line with a sequence of contiguous grid cells and corresponding traversal times that are calculated by a modified Bresenham line algorithm [6]. After eliminating short trajectories (less than 300 seconds or 10 grid cells), approximately 2 million trips that have an average length of 1390 seconds and 94 grid cells and refer to 2 billion 100-meter grid cells have been identified.

Experiments: A series of experiments was conducted to evaluate the prediction error of IncCCFR, for (1) varying *min_sup* values, (2) varying length of prediction horizons Δt_p, and (3) various mining and prediction scenarios and to compare it against a baseline predictor and its variants. The following describes the aims and settings of these experiments and Figures 1–3 show a subset of their results.

First, a natural baseline method (labeled 'global') is constructed that possesses the most information, but least intelligence, and bases its predictions solely on neighboring cell probabilities, which are derived from *all* trajectories. To examine the effects of the cycle prevention and the combined cycle *and* U-turn prevention, variants of the baseline method are evaluated (labeled 'g ¬o' and 'g ¬ou', respectively). All other methods are tested in a more realistic online setting according to a TSWM with parameters $t_{wsize} = 60$ minutes and $t_{wstride} = 5$ minutes, where predictions are based on information that is either in the current window only (labeled 'local', 'l ¬o', 'l ¬ou' for corresponding online variants of the baseline method, and '60' for the CCFR-based prediction with *min_sup* = 60[2]) or in the current window and a set of temporally relevant windows (labeled '60, 6d' for hour-of-day projected and '60, 4d' for hour-of-day *and* weekday projected versions of the CCFR-based prediction with *min_sup* = 60). In order to evaluate the accuracy of incremental mining, experiments are performed in which each 60-minute mining window is subdivided into three 20-minute subwindows that are mined, weighted and combined, mined again and used for prediction. The results of these experiments are not explicitly shown in either of the figures because the prediction error of IncCCFR under this scenario is virtually identical to the direct mining method (labeled '60'). In all experiments the CCFR-based predictors incorporate combined cycle *and* U-turn prevention.

The absolute prediction error of the methods is measured in terms of the average cell distance between the predicted grid cell and the actual grid cell of the objects at the prediction / time horizon and is shown in Figure 1. As an attempt to assess the source of the prediction error (spatial, i.e., incorrect path prediction or temporal, i.e., incorrect traversal time prediction) the prediction error is also measured in terms of the average cell distance between the actual grid and the predicted cell that is closest to it, i.e., is the 'best' prediction within the prediction horizon, and is shown in Figure 1 (labeled 'best'). Finally, Figure 2 shows relative prediction improvements compared to the baseline and Figure 3 shows relative prediction improvements for 'good' predictions (i.e., percentage point improvement for predictions where the predicted grid cell was no more than 5 grid cells away) compared to the baseline.

Figures 1 and 2 show that (1) the prediction error of all methods nearly linearly increases with Δt_p, (2) the prediction error of the local method is only slightly worse than the

[2]Lower *min_sup* yields better predictions, but due to limits of space results are shown only for the lowest value tested.

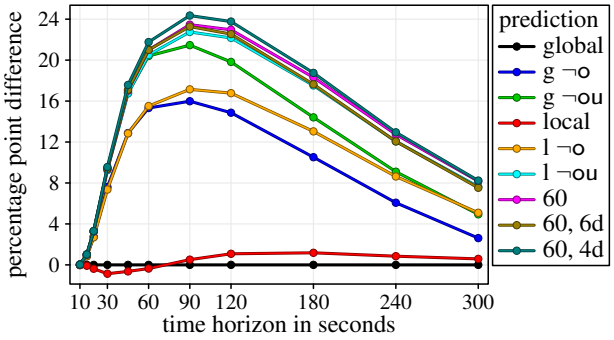

Figure 3: Relative prediction accuracy of 'good' predictions (i.e., percentage point improvement for predictions where the predicted grid cell was no more than 5 grid cells away) of different methods w.r.t. the baseline predictor 'global'.

global method, which is evidence that basing predictions on more information without regard for temporal relevance does not decrease the prediction error, (3) while the CCFR-based predictor outperforms both the baseline and its online variant ('local'), the performance gap can be closed by the combined cycle *and* U-turn prevention (label '60' vs. 'l ¬ou' in Figure 2), and (4) combining temporally relevant windows ('60, 4d') reduces the prediction error and a combination of temporal domain projections can effectively capture the temporal and periodic regularities in movement. Finally, Figure 3 shows that the absolute prediction performance gaps between the methods and the baseline peak at around 90 seconds and gradually degrade thereafter.

6. CONCLUSIONS AND FUTURE WORK

To enable moving object trajectory based modern traffic prediction and management systems, the present paper proposed IncCCFR—a novel, incremental approach for managing, mining, and predicting the incrementally evolving trajectories of moving objects. The proposed prediction model is essentially a varying order, deterministic Markov model that is based on closed contiguous frequent routes and neighboring cell probabilities that are derived from the object trajectories. In addition to reduced mining and storage costs, a key advantage of the incremental approach is its ability to combine multiple temporally relevant mining results from the past to capture temporal and periodic regularities in movement, making the prediction model time inhomogeneous. This ability of the method is demonstrated in a series of experiments on a large real-world trajectory data set by comparing the prediction performance of the proposed method to the performance of the simple neighboring cell probability based predictor and its variants.

Future work is planned in several directions. First, the proposed pattern combination approach opens up new possibilities for parallelizing the mining task. Namely, one can partition the input stream into several independently mined substreams [8] from which the patterns can be trivially combined. The cost-benefit evaluation of different partitioning/parallelization schemes in terms of execution times and prediction accuracy can be evaluated. Second, the proposed extension of the method to react to rare, unpredictable, sudden changes can be implemented and evaluated.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. *Proc. of VLDB*, pp. 487-499, 1994.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. *Proc. of ICDE*, pp. 3-14, 1995.

[3] A. Asahara, A. Sato, K. Maruyama, and K. Seto. Pedestrian-movement prediction based on mixed Markov-chain model. In *Proc. of ACM-GIS*, pp. 25-33, 2011.

[4] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275-286, 2003.

[5] A. Bifet, G. Holmes, B. Pfahringer and R. Gavaldà. Mining Frequent Closed Graphs on Evolving Data Streams. *Proc. of KDD*. pp. 591-599, 2011.

[6] J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25-30, 1965.

[7] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. *Proc. of SIGKDD*, pp. 330-339, 2007.

[8] G. Gidófalvi, T.B. Pedersen, T. Risch, and E. Zeitler. Highly scalable trip grouping for large scale collective transportation systems. *Proc. of EDBT*, pp. 678-689, 2008.

[9] G. Gidófalvi and E. Saqib. From trajectories of moving objects to route-based traffic prediction and management. *Proc. of Workshop MPA*, 132-135, 2010.

[10] G. Gidófalvi, M. Kaul, C. Borgelt, and T.B. Pedersen. Frequent route based continuous moving object location- and density prediction on road networks. *Proc. of ACM SIGSPATIAL GIS*, pp. 381-384, 2011.

[11] G. Gidófalvi and F. Dong. When And Where Next: Individual mobility prediction. *Proc. of Workshop MobiGIS*, pp. 57-64, 2012.

[12] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. *Proc. of ICDE*, pp. 70-79, 2008.

[13] H. Jeung, M. Yiu, X. Zhou, and C. Jensen. Path prediction and predictive range querying in road network databases. *Proc. of VLDB*, pp. 585-602, 2010.

[14] N. Jiang and L. Gruenwald. CFI-stream: Mining closed frequent itemsets in data streams. *Proc. of SIGKDD*, pp. 592-597, 2006.

[15] H. Kriegel, M. Renz, M. Schubert, and A. Zuefle. Statistical density prediction in traffic networks. *Proc. of SDM*, 2008.

[16] J. Krumm. A Markov Model for Driver Turn Prediction. In *Proc. of SAE World Congress*, 2008.

[17] J. Krumm, R. Gruena, and D. Delling. From destination prediction to route prediction. *JLBS*, 7(2):98-120, 2013.

[18] Q. Li, T. Zhang and Y. Yu. Using cloud computing to process intensive floating car data for urban traffic surveillance. *IJGIS*, 25(8):1303-1322, 2011.

[19] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. WhereNext: a location predictor on trajectory pattern mining. *Proc. of KDD*, pp. 637-645, 2009.

[20] J. Pei, J. Han, B. Mortazavi-Asl, and H. Pinto. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. *Proc. of ICDE*, pp. 214-224, 2001.

[21] A.M.J. Md. Zubair Rahman and P. Balasubramanie. Weighted Support Association Rule Mining Using Closed Itemset Lattices in Parallel. *IJCSNS*, 9(3):247-253, 2009.

[22] F. Verhein and S. Chawla. Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. *Proc. of DASFAA*, pp. 187-201, 2006.

[23] Y. Ye, Y. Zheng, Y. Chen, J. Feng and X. Xie. Mining individual life patterns based on location history. *Proc. of MDM*, pp. 1-10, 2009.

[24] J.J.-C. Ying, W.-C. Lee, T.-C. Weng and V.S. Tseng. Semantic trajectory mining for location prediction. *Proc. of ACM-GIS*, pp. 34-43, 2011.