Predicting Long-Term Trajectories of Connected Vehicles via the Prefix-Projection Technique

Shaojie Qiao, *Member, IEEE*, Nan Han, Junfeng Wang, *Member, IEEE*, Rong-Hua Li, *Member, IEEE*, Louis Alberto Gutierrez, and Xindong Wu, *Fellow, IEEE*

***Abstract*— The vehicle location prediction based on their spatial and temporal information is an important and difficult task in many applications. In the last few years, devices, such as connected vehicles, smart phones, GPS navigation systems, and smart home appliances, have amassed the large stores of geographic data. The task of leveraging this data by employing moving objects database techniques to predict spatio-temporal locations in an accurate and efficient fashion, comprising a complete trajectory remains an actively researched area. Existing methods for frequent sequential pattern mining tend to be limited to predicting short-term partial trajectories, at extremely high computational costs. In order to address these limitations, we designed a prefix-projection-based trajectory prediction algorithm called PrefixTP, which contains three essential phases. First, data collection, connected vehicles equipped with sensors comprise a vehicle grid and generate copious amounts of spatio-temporal data, in order to communicate and share traffic information. Second, model training, examining only the prefix subsequences, and projecting only their corresponding postfix subsequences into projected sets. Finally, trajectory matching, recursively finding postfix sequences meeting the requirement of minimum support count, and outputting the most frequent sequential pattern as the most probable trajectory. Fundamentally, PrefixTP supports three trajectory matching strategies which encompass all possibilities of prediction. Extensive experiments were conducted using real world GPS data sets, and the results show, when comparing predicted complete trajectories against partial short-term trajectories with a guarantee of real-time forecasting, that PrefixTP outperforms first-order, second-order Markov models, and Apriori-based trajectory prediction algorithm.**

Manuscript received May 13, 2017; revised July 10, 2017 and August 20, 2017; accepted September 2, 2017. Date of publication October 9, 2017; date of current version June 28, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61772091 and Grant 61100045, in part by the Planning Foundation for Humanities and Social Sciences of Ministry of Education of China under Grant 15YJAZH058, the Youth Foundation for Humanities and Social Sciences of Ministry of Education of China under Grant 14YJCZH046. The Associate Editor for this paper was R. Malekian. *(Corresponding authors: Nan Han; Junfeng Wang.)*

S. Qiao is with the School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China (e-mail: qiaoshaojie@gmail.com).

N. Han is with the School of Management, Chengdu University of Information Technology, Chengdu 610103, China (e-mail: hannan@cuit.edu.cn).

J. Wang is with the School of Aeronautics and Astronautics, Sichuan University, Chengdu 610065, China (e-mail: wangjf@scu.edu.cn).

R.-H. Li is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China.

L. A. Gutierrez is with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 USA.

X. Wu is with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA 70503 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2750075

***Index Terms*— Trajectory prediction, moving objects databases, connected vehicles, frequent sequential pattern, prefix-projection.**

I. Introduction

CONNECTED vehicles enables any vehicle, anywhere, to act as a smart node, collecting and sharing information on vehicles, roads and the surroundings [1]. It follows then, that the field of moving object databases (MOD) over trajectory data accumulated by connected vehicles has seen a surge of interest. This research, inclusive of large-scale and variable trajectory data, urgently needs innovative, intelligent, efficient and effective approaches to discover a large amount of hidden knowledge in it [2], [3]. The increasing availability of location-awareness technologies, such as GPS (Global Positioning System) and WIFI (Wireless Fidelity) etc., have resulted in huge volumes of spatio-temporal data, especially in the form of trajectories, which are represented by sequential patterns of vehicles, in a typical scenario, describing the behavior of movements: containing frequently-visited areas, users' preferences on travelling itineraries, etc.

Vehicle location prediction, also known as trajectory prediction (abbreviated as TP), has received increased attention by researchers. An illustrative TP example for connected vehicles is autonomous navigation when vehicle-to-vehicle (V2V) navigation function does not work due to network attacks or the communication signal is shielded.

Example 1: When vehicles lost the capability of V2V navigation, connected vehicles can only employ autonomous navigation to find appropriate routes. Unlike the short-term time series prediction, the long-term prediction is typically faced with growing uncertainties arising from various sources. For example, the accumulation of errors make the prediction more difficult. By discovering frequent sequential patterns of connected autonomous vehicles, an advanced autonomous vehicle system can efficiently process all the sensory data and discover appropriate paths and avoids obstacles on such paths.

The aforementioned example motivates us to propose novel, scalable and effective vehicle location prediction approaches based on massive trajectory data. Furthermore, location based services (LBS) continue to increase in popularity, and moreover, are becoming embedded in people's daily life with the widespread mobile technologies. If the LBS providers develop sound methods to infer the user's general itinerary in advance,

1524-9050 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

this would provide an opportunity to recommend the most relevant travel information so as to maximize target user's potential. For example, if a given system could effectively infer the nature of a journey, for the user, is to visit friends or family, then the system could suggest taxi-hailing APPs (applications) that may provide special offers or discount prices. Moreover, if the system could decipher the adjacent users who have similar specifications for taxis, then a triggered response for a carpooling ride sharing option can be recommended to the given subset of users. Another example is if the system could suggest optimal candidate roads, in order to avoid additional time in traffic, based on historical traffic data, time, weather and additional external factors. This recommendation would leverage trajectory prediction techniques by analyzing individuals' movement behaviors. This can help provide better LBS and contribute to users' time management as well.

There is an ever-increasing interest in trajectory databases with moving objects [4]. Trajectory prediction is a very challenging and practical problem in MOD due to the following reasons [5], [6]: (1) The uncertainty of moving objects (including connected vehicles), traditional TP methods focus on predicting short-term partial trajectories, and existing TP algorithms have relatively low prediction accuracy for long-term predictions, and even do not works well for inferring a continuous and complete trajectory. (2) Traditional distance vector based TP methods can only be applied to predict possible paths within fixed road networks. When connected vehicles remain in road junctions due to traffic jams or other complex traffic condition, they cannot provide optimal routes. (3) Existing TP approaches often perform poorly when faced with environmental and other external factors, such as traffic jams, inclement weather, etc. (4) The computational cost of classical frequent sequential patterns based TP methods is very high, which lies in generating a large volume of candidate sequences.

Mining frequently patterns from trajectory databases is a commonly-used approach to predict paths, and can find that the Apriori-like sequential pattern mining algorithm bears the following computational overhead [7]:

(1) Potentially a Huge Number of Candidate Sequences: Because it needs to enumerate all the possible permutations of items in a candidate trajectory sequence, the Apriori-based approach may generate a large set of candidate sequences. For example, if there are 10000 frequent trajectories of length-1, i.e., $\langle s_1\rangle, \langle s_2\rangle, \ldots, \langle s_{10000}\rangle$, an Apriori-like method may generate $10000 \times 10000 + \frac{10000\times 9999}{2} = 149{,}995{,}000$ candidate trajectories.

(2) Difficulties at Predicting Long-Term Trajectory Sequential Patterns: A long trajectory pattern must increase from a combination of short ones, but the number of candidate sequences is exponential to the length of trajectory patterns to be discovered. For example, suppose there is a single trajectory of length 50, $s = \langle s_1, s_2, \ldots, s_{50}\rangle$, in the trajectory database, and the minimum support threshold is set to 1, the Apriori-like method has to generate 50 length-1 candidate sequences, $50 \times 50 + \frac{50\times 49}{2} = 3725$ length-2 candidate sequences, $\binom{50}{3} = 19600$ length-3 candidate sequences,[1]$\cdots$. Obviously, the total number of candidate sequences is greater than $\sum_{i=1}^{50}\binom{50}{i} = 2^{50} - 1 \approx 10^{15}$.

(3) Multiple Scans of Trajectory Databases: Because the length of each candidate trajectory increases by one through each scan of trajectory database. For example, the Apriori-like algorithm must scan the database at least 20 times to find a sequential pattern $\{(abcd)(abcd)(abcd)(abcd)(abcd)\}$.

With the goal of overcoming some of the challenges in TP approaches based on mining frequently occurring patterns, this research makes the following original contributions:

(1) We apply an efficient prefix-projection technique to find frequent trajectory patterns of connected vehicles, which examines only the prefix subsequences and projects only their corresponding postfix subsequences into projected sets.

(2) We propose an incremental trajectory matching approach which includes three matching strategies to recursively mine frequent sequential patterns over postfix sequences, which suits to forecast long-term and variable length of trajectories in a connected vehicle environment.

II. Related Work

Predicting long-term trajectories with uncertainty in MOD or a connected vehicles environment has recently been receiving increased attention. Existing work relevant to TP mainly focuses on discovering frequent patterns [8], [9]. Monreale *et al.* [10] extracted the frequently visited sequences of regions and detected the best matching one in the *T-pattern Tree*. But, the computational complexity of building T-pattern Tree is very high. Many of the existing prediction techniques only take into consideration the geographic features of trajectory points. Ying *et al.* [11] proposed an approach for predicting the next location of objects by geography and the semantic information of trajectories. It follows, though, that this method requires the calculation of a *Semantic Score* for each candidate path, which produces additional computational cost. Qiao *et al.* [5] proposed a three-in-one TP model, which predicts possible trajectories of moving objects with relative uncertainty, while the generation of frequent patterns tree needs multiple scans of databases. Goodall [12] proposed new real-time location prediction techniques in a connected vehicle environment, which is expected to result in reduced delays and improve traffic flow. But, the results show that the reduction in delay depends on the data quality of connected vehicles. A real-time traffic state estimation framework [13] was proposed to predict traffic density, which can improve the accuracy of autonomous navigation. But, this study does not take into account non-connected vehicle data, e.g., social media data.

Markov models are commonly used to discover frequent trajectory patterns [14], [15]. Research was contributed by

[1]It is worthwhile to note that Apriori does reduce the search space. Otherwise, the number of length-3 candidate sequences would have been $50 \times 50 \times 50 + 50 \times 50 \times 49 + \frac{50\times 49\times 48}{3\times 2} = 267,100$

Qiao *et al.* [2], which proposed a HMM based trajectory prediction algorithm, which can self-adaptively select important parameters. However, this method still cannot well handle the answer-loss problem and the state retention problem due to discontinuous chain of hidden states. Jeung *et al.* [16] used HMM to discover trajectory patterns by utilizing the effectiveness of space-partitioning methods. However, this approach cannot be applied to predict future locations of moving objects due to the spatio-temporal characteristic of trajectory data. In order to predict pedestrian movement, Asahara *et al.* [17] proposed a mixed Markov-chain model, which has an observable parameter like a HMM, but with natural variation: the unobservable parameter is fixed during the state transition. In summary, there are three main drawbacks in HMM-based trajectory prediction models: (1) Markov models do not consider the discontinuous chain of hidden states, and the state retention problem can greatly degrade the accuracy of location prediction. (2) HMM-based models do not avoid the answer-loss problem, that is, two similar trajectories are viewed to be two unrelated trajectories after partitioning the grid of geographic space, which will affect the accuracy of prediction. Lastly, (3) HMM-based prediction models depend on the training data. For the irregular spatio-temporal trajectory data, the movement rules cannot be easily represented by Markov chains due to the uncertainty movement behavior of connected vehicles, which can lead to the loss of continuous location information.

The work published in Science by Song *et al.* [18], which proved that there can be 93% potential for predictability in user mobility by measuring the entropy of each trajectory. The work motivates us to predict movement behavior of connected vehicles. Recent work in TP, such as the aforementioned, has begun to attract lots of researchers which has lead to a number of new state-of-the-art approaches as introduced in [19]–[21]. In order to support short-term traffic state prediction, Pan *et al.* [22] proposed a multivariate normal distribution-based best linear predictor. However, the proposed approach cannot capture the moving-bottleneck effect due to limitations in the proposed framework. Ding *et al.* [23] proposed a network-matched trajectory-based moving-object database mechanism. But it cannot select important parameters in a self-adaptive manner. Two efficient route planning strategies are proposed [24], where several effective techniques are employed to avoid both unnecessary calculations on large graphs and excessive re-calculations caused by updating traffic conditions. Zheng *et al.* [25] discovered the classical travel sequences among locations relative to the interests of these locations and users' travel experiences. However, the proposed method is mainly used for recommending locations of interest instead of suggesting a complete trajectory. Zhou *et al.* [26] proposed a "semi-lazy" approach to path prediction that builds prediction models using dynamically selected reference trajectories without having to tune several parameters. This method cannot be applied to a connected vehicle environment due to its limitations on spatial data.

In regard to the aforementioned research on TP, the following challenges were identified: (1) most of existing TP methods are not applicable for predicting long-term trajectories [27]; (2) it is time intensive to discover frequent sequential patterns of moving objects from massive trajectory points, because it requires the MOD to be scanned several times in order to find postfix patterns; (3) the current TP approaches mainly focus on predicting single movement patterns, and cannot be applied to infer multiple types of patterns in complex scenarios; (4) due to the uncertainty of spatial-temporal data, the prediction accuracy cannot be guaranteed, and prediction bias can be prevalent.

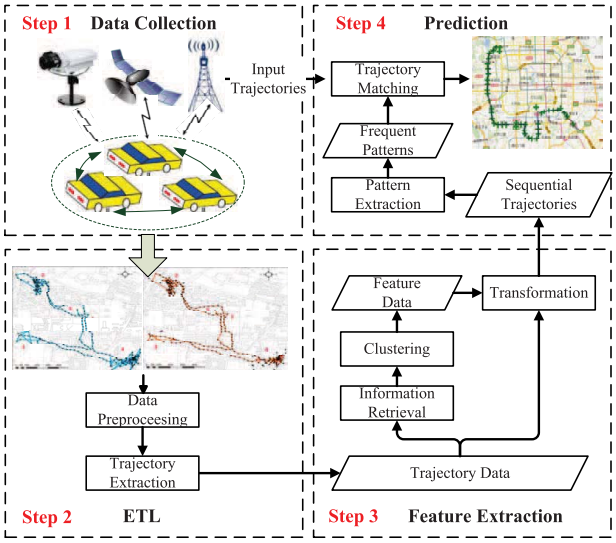


Fig. 1. TP framework based on prefix-projected patterns growth.

The privacy issue of frequent sequential pattern mining in a connected vehicle environment is very important, because we have to protect the confidential information of drivers in connected vehicle networks. In order to investigate situations where releasing frequent sequential patterns can compromise the privacy of individuals, Jin *et al.* [28] proposed two privacy protection approaches, i.e., k-anonymity and α-dissociation, which can greatly reduce privacy disclosure risk carried by frequent sequential patterns. Weber [29] introduced new security and privacy challenges in Internet of things, and emphasized the necessarily of establishing an adequate legal framework of protecting individuals privacy. Mhatre *et al.* [30] proposed a procedure to protect the privacy of data frequent sequential patterns over progressive databases. The scalability of the proposed method should be further explored from a single node system to a multi-party scenario.

III. Trajectory Prediction Framework Based on Prefix-Projection Technique

In this section, we will introduce a new trajectory prediction framework for connected vehicles based on prefix-projected pattern growth, its working mechanism is depicted in Fig. 1.

The framework contains four essential modules: (1) In the data collection module, a large amount of trajectory data can be collected by a vehicle cloud of connected devices (i.e., sensors, camera, communication base); (2) ETL (Extract-Transform-Load) techniques are applied to remove noise, extract complete trajectories comprising of a series of

spatio-temporal points, partition a complete trajectory into partial segments based on time and distance, and lastly transform raw data into an acceptable format; (3) The feature extraction module works to retrieve important feature of trajectories and group them into clusters, then transform feature data into trajectory sequential patterns; (4) The prefix-projection technique is applied in order to discover frequent sequential patterns and obtain a candidate set, then different trajectory matching strategies are applied to infer a complete path based on frequent visits.

Fig. 1 is a generic framework and can be applied to most TP scenarios for connected as well as unequipped vehicles without sensors. In particular, the optimal path information discovered by the proposed prefix-projection technique can be shared among connected devices by communication infrastructures in the vehicle cloud.

In the phase of data collection, we employed the vehicle cloud model presented in [31]. The vehicular cloud of connected vehicles is different from the Internet cloud that is created and maintained by a cloud provider, which is temporarily created by inter-connecting resources available in the vehicles and Road Side Units [31]. In addition, resources are unlike the ones in a traditional cloud. Each vehicle has three kinds of resources, that is, sensors, data storage, and computing. The sensor is able to self-actuate as well as to detect events. Sensors are directly connected to the Internet of Vehicles, in order to be read and controlled by an external system. Data storage devices are used to record vehicle data collected from sensors and applications. It provides functions of data sharing and communication among connected vehicles.

The resources are inter-networked by purely peer-to-peer connections in the vehicular cloud [31]. Aiming to obtain the real-time road and traffic information from other vehicles, each vehicle share resource directly with others. In order to achieve data sharing and efficient communication, there is a hub vehicle with the highest value of centrality and connectivity in this cloud. Then, it is responsible for handle the process of resource sharing as well as other cloud operations. By the vehicle cloud model, each participant calculate and share the optimal route by the proposed prefix-projection-based TP model as introduced in the following sections.

IV. Preliminaries

This section works to introduce fundamental concepts which help to describe the TP problem on mining frequent trajecotry patterns based on concepts introduced in [7].

Definition 1 (Trajectory): $T = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\}$ represents a trajectory, where $\alpha_i = \langle longitude, latitude, time\rangle$, and $\alpha_i \in P$, P is a set of spatio-temporal points.

Definition 2 (Trajectory Sequence): Given a trajectory T, after feature extraction and transformation, a trajectory sequence is obtained, denoted by $\alpha = \langle \alpha_1, \alpha_2, \ldots, \alpha_i\rangle, \alpha \subseteq T$, α is an ordered list of spatio-temporal points by the timestamp i, and is called a trajectory sequence. The number of points in a trajectory sequence is called the length of the sequence. A sequence with length l is called an l-sequence.

Definition 3 (Sub-Trajectory): $\beta = \langle \beta_1, \beta_2, \ldots, \beta_p\rangle$ is a sub-trajectory of sequence $\alpha = \langle \alpha_1, \alpha_2, \ldots, \alpha_q\rangle$, $p \leq q$, if there is integers $\langle i_1, i_2, \ldots, i_p\rangle$ $(1 \leq i_1 < i_2 < \ldots < i_p)$, $\langle j_1, j_2, \ldots, j_p\rangle$ $(1 \leq j_1 < j_2 < \ldots < j_p)$, and $i_1 \leq j_1$, $\beta_{i_1} = \alpha_{j_1}, \beta_{i_2} = \alpha_{j_2}, \ldots, \beta_{i_p} = \alpha_{j_p}$, then β is called a sub-trajectory of α or α is called a super-trajectory of β, denoted by $\beta \sqsubseteq \alpha$.

Definition 4 (Frequent Trajectory Sequential Patterns): Given a positive integer ξ as the support threshold, a trajectory sequence α is called a frequent trajectory sequential pattern in trajectory database T if the sequence is contained by at least ξ tuples in the database, i.e., $supports_T(\alpha) \geq \xi$. A frequent trajectory with length l is called an length-l pattern.

Definition 5 (Prefix Sequence): Given a trajectory sequence $\alpha = \langle \alpha_1, \alpha_2, \ldots, \alpha_p\rangle$, and $\beta = \langle \beta_1, \beta_2, \ldots, \beta_q\rangle$, $p \leq q$, if and only if $\alpha_1 = \beta_1, \alpha_2 = \beta_2, \ldots, \alpha_p = \beta_p$, then α is called a prefix sequence of β.

Definition 6 (Projection): Given two trajectory sequences α and β, β is a sub-trajectory of α. A subsequence α' of α is called a projection of α w.r.t. prefix β if and only if: (1) β is the prefix sequence of α', and (2) there is no proper super-trajectory α^* of α' (i.e., $\alpha' \sqsubseteq \alpha^*$ but $\alpha' \neq \alpha^*$) such that α^* is a subsequence of α but also has prefix β.

Definition 7 (Postfix Sequence): Given a trajectory sequence $\alpha = \langle s_1, s_2, \ldots, s_n\rangle$ is the projection of $\beta = \langle s_1, s_2, \ldots, s_{m-1}, s_m\rangle$, where $m \leq n$, a trajectory sequence $\gamma = \langle s_{m+1}, \ldots, s_n\rangle$ is called the postfix sequence of α w.r.t. prefix β, denoted as $\gamma = \alpha \ominus \beta$. We denoted $\alpha = \beta \oplus \gamma$, where $\ominus$ and $\oplus$ represent subtraction and combination operations between partial trajectories, respectively.

Example 2: $\langle a\rangle$, $\langle aa\rangle$, $\langle a(ab)\rangle$ and $\langle a(ab)a\rangle$ are prefix sequences w.r.t a trajectory $\langle a(ab)(abc)(def)\rangle$, but neither $\langle ab\rangle$ nor $\langle a(b)a\rangle$ is considered to be a prefix. $\langle\ (ab)(abc)(def)\rangle$ is a postfix of the same sequence w.r.t. prefix $\langle a\rangle$, $\langle\ (\_b)(abc)(def)\rangle$ is a postfix w.r.t. prefix $\langle aa\rangle$, and $\langle\ (abc)(def)\rangle$ is a postfix w.r.t. prefix $\langle ab\rangle$.

Definition 8 (Projected Set): Given a trajectory α is a frequent sequential pattern in the trajectory database T, an **α-projected set**, denoted as $T|\alpha$, is the collection of postfix sequences in T w.r.t. prefix α.

Definition 9 (Trajectory Support Count): Let α be a sequential pattern in the trajectory database T, β be a trajectory having prefix α. The trajectory support count of β in α-projected set $T|\alpha$ is the number of trajectory sequences γ in $T|\alpha$ such that $\beta \sqsubseteq \alpha \oplus \gamma$, denoted as $support_{T|\alpha}(\beta)$.

Before matching trajectories, frequent trajectories must be mined. Discovering frequent sequential patterns is a recursive process of mining, and we can obtain the following properties [7].

Lemma 1: Assume that α is a trajectory pattern composed of l spatio-temporal points, $\langle \beta_1, \beta_2, \ldots, \beta_m\rangle$ is the set of all length-$(l+1)$ trajectory sequential patterns viewing α as its prefix sequence. The complete set of sequential patterns having prefix α, with the exception of α, can be partitioned into m different subsets. The j^{th} subset $(1 < j < m)$ is the set of trajectory patterns having prefix β_j.

Based on Lemma 1, the projection set of trajectories is partitioned as needed, which can reduce the number of traversed

points, and choose the most probable candidate items to be extended.

Lemma 2: Let α and β be two patterns in trajectory database T such that α is a prefix sequence of β, then:

1) $T|\beta = (T|\alpha)|\beta$;
2) for any sequence γ having prefix α, $support_T(\gamma) = supports_{T|\alpha}(\gamma)$, and
3) The size of α-projected set cannot exceed that of T.

Lemma 2 presents the relation between the frequent trajectory patterns α and β, having prefix α. Since $T|\beta = (T|\alpha)|\beta$, all frequent sequential patterns over β via the projection set of α can be discovered. In addition, $support_T(\gamma) = support_{T|\alpha}(\gamma)$ can guarantee that the incrementally discovered sequential pattern β, having prefix α, is a frequent trajectory.

Corollary 1: The trajectory pattern α is frequent if α meets the properties given in Lemma 1 and Lemma 2.

V. Trajectory Prediction Based on Prefix Projection

A. Working Mechanism

Based on the above preliminaries, we can define the TP problem as an inference of the continuous trajectory sequential pattern α, using the postfix sequence γ of α, where γ is viewed as the prefix sequence when forecasting the subsequent trajectory of α. The prefix-projection based TP algorithm includes the following primary steps:

(1) **Mine length-1 patterns.** Scan trajectory database T once to discover all frequent items which have support count greater than the given minimum support count. Each of these frequent items is a length-1 pattern, the set of length-1 patterns is denoted as F.

(2) **Partition the space.** The set of trajectory sequential patterns T can be divided into $|F|$ subsets, and each item in the subset corresponds to a single length-1 pattern p_1, having p_1 as its prefix sequence, where $|F|$ represents the number of items in F.

(3) **Discover frequent trajectory sequential patterns.** This step contain two operations: constructing the projected sets and discover each frequent trajectory pattern recursively.

(4) **Predict future location points via trajectory matching.** Given a projected set $T|\alpha$, the incremental item having prefix α will, with certainty, appear in $T|\alpha$. Each time a frequent sequential pattern extends to a next probable location, the following occurs: traverses the projected set $T|\alpha$, calculates the support count of the first item in a sequential pattern, and finds such an item having count greater than the support threshold of the extending item w.r.t. α, then the sequential pattern is divided, w.r.t. the extending item, into the corresponding subset. The above process is recursively performed on the projected sets.

B. Frequent Sequential Patterns Mining

The following is a detailed example of mining frequent trajectory patterns.

TABLE I
Example of Trajectory Sequential Patterns

ID	Trajectory Sequence
1	$\langle a\ b\ d\ e\rangle$
2	$\langle a\ b\ d\ c\rangle$
3	$\langle a\ c\ k\ e\rangle$
4	$\langle b\ c\ a\ d\ e\ f\rangle$
5	$\langle d\ a\ b\ c\ e\rangle$
6	$\langle e\ b\ a\ d\ e\rangle$
7	$\langle a\ a\ b\ a\ b\ c\ d\ e\ f\rangle$

Example 3: Suppose there exist the following set of trajectory patterns given in Table I with $min\_sup = 2$.

According to Table I, the item set is $\{a, b, c, d, e, f, k\}$, and we can obtain the support count of each item, i.e., $\langle a\rangle$:7, $\langle b\rangle$:6, $\langle c\rangle$:5, $\langle d\rangle$:6, $\langle e\rangle$:6, $\langle f\rangle$:2, $\langle k\rangle$:1, where the number following each item represents the support count. Given that the support count of $\langle k\rangle$ is less than min_sup, this item is discarded. The item set $\{a, b, c, d, e, f\}$ includes all length-1 sequential patterns and is employed for the next step of mining. The next step is to construct the projected set for each length-1 sequential pattern.

Consider the example $\langle a\rangle$ which shows the detail of finding length-l patterns. The $\langle a\rangle$-projected set, and the incremental mining process is presented in Table II.

As shown in Table II, in the first round of scan, $\langle a\rangle$-projected set is found consisting of six postfix sequences: $\{\langle ababcdef\rangle$, $\langle bde\rangle$, $\langle bdc\rangle$, $\langle bce\rangle$, $\langle cke\rangle$, $\langle def\rangle$, $\langle de\rangle\}$, By scanning $\langle a\rangle$-projected set once, all the length-2 patterns having prefix $\langle a\rangle$ can be found. They are given as: $\langle ab\rangle$:5, and $\langle ad\rangle$:2. Note that $\langle ac\rangle$:1 is discarded because it is less than the minimum support count.

Recursively, all trajectory sequential patterns having prefix $\langle a\rangle$ can be partitioned into two subsets: (1) those having prefix $\langle ab\rangle$, and (2) those having prefix $\langle ad\rangle$. These subsets can be mined by constructing respective projected sets, and mining each recursively.

The $\langle ab\rangle$-projected set consists of three postfix sequences: $\langle abcdef\rangle$, $\langle cdef\rangle$, $\langle ce\rangle$, $\langle de\rangle$, and $\langle dc\rangle$. By scanning $\langle ab\rangle$-projected set once, we find the length-3 patterns having prefix $\langle ab\rangle$ is $\langle abd\rangle$. The $\langle abd\rangle$-projected set can be constructed and recursively mined in a similar fashion. However, in the third round of scan, since there is no expectation of generating any frequent subsequences from a single sequence, the processing of $\langle abd\rangle$-projected set terminates.

Similarly, the $\langle ad\rangle$-projected set is comprised of two postfix sequences: $\langle ef\rangle$ and $\langle e\rangle$. By scanning $\langle ad\rangle$-projected set once, the length-3 sequential patterns having prefix $\langle ad\rangle$ is found, given as $\langle ade\rangle$. The $\langle ade\rangle$-projected set is constructed and recursively discovered. The processing of $\langle ade\rangle$-projected set terminates, because there are not any frequent subsequences in the $\langle ade\rangle$-projected set.

Finally, we can find trajectory patterns having prefix $\langle a\rangle$ consist of five trajectories including $\langle a\rangle$, $\langle ab\rangle$, $\langle ad\rangle$, $\langle abd\rangle$, and $\langle ade\rangle$.

Similarly, we can find sequential patterns having prefixes $\langle b\rangle$, $\langle d\rangle$, $\langle e\rangle$, $\langle f\rangle$ by constructing $\langle b\rangle$, $\langle d\rangle$, $\langle e\rangle$, $\langle f\rangle$-projected sets, respectively, and mining them.

TABLE II
EXAMPLE OF MINING FREQUENT SEQUENTIAL PATTERNS

First round scan		Second round scan		Third round scan	
length-1	projected set	length-2	projected set	length-3	projected set
$\langle a \rangle$	$\langle ababcdef\rangle$,$\langle bde\rangle$,$\langle bdc\rangle$,$\langle bce\rangle$ $\langle cke\rangle$,$\langle def\rangle$,$\langle de\rangle$	$\langle ab\rangle$	$\langle abcdef\rangle$,$\langle cdef\rangle$,$\langle ce\rangle$,$\langle de\rangle$,$\langle dc\rangle$	$\langle abd\rangle$	$\langle e\rangle$,$\langle c\rangle$
		$\langle ad\rangle$	$\langle ef\rangle$,$\langle e\rangle$	$\langle ade\rangle$	$\langle f\rangle$
$\langle b \rangle$	$\langle ade\rangle$,$\langle abcdef\rangle$,$\langle cadef\rangle$ $\langle ce\rangle$,$\langle cdef\rangle$,$\langle dc\rangle$,$\langle de\rangle$	$\langle bc\rangle$	$\langle adef\rangle$,$\langle def\rangle$,$\langle e\rangle$,	ϕ	ϕ
		$\langle bd\rangle$	$\langle c\rangle$,$\langle e\rangle$	ϕ	ϕ
$\langle d \rangle$	$\langle c\rangle$,$\langle e\rangle$,$\langle ef\rangle$	$\langle de\rangle$	$\langle f\rangle$	ϕ	ϕ
$\langle e \rangle$	$\langle bade\rangle$,$\langle f\rangle$	ϕ	ϕ	ϕ	ϕ
$\langle f \rangle$	ϕ	ϕ	ϕ	ϕ	ϕ

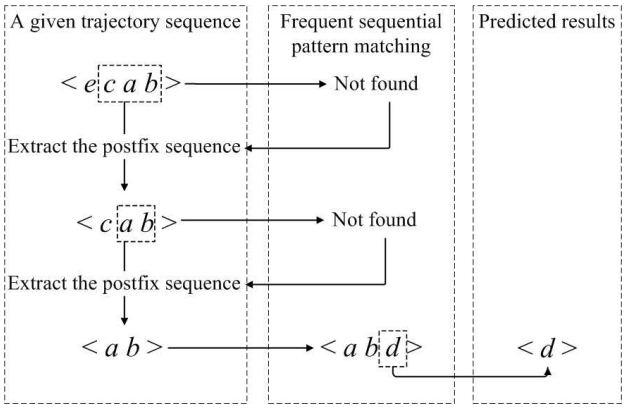


Fig. 2. Example of trajectory prediction based on frequent patterns.

C. Trajectory Matching

After mining frequent trajectory sequential patterns, the TP problem is simplified to only match frequent trajectory sequential patterns. Based on the discovered sequential patterns in Example 3, Fig. 2 gives an example of trajectory prediction by prefix-projected sequences.

1) Prediction: Suppose there is a given trajectory sequence $S = \langle e\ c\ a\ b\rangle$ in Fig. 2, we will predict the future most probable location w.r.t. S. First, scan the set of frequent trajectories, and find frequent sequential patterns having prefix $S = \langle e\ c\ a\ b\rangle$. However in this case, the system returns the result "Not found", and then compresses the given trajectory by extracting the postfix sequence having prefix $\langle a\rangle$, that is $S_1 = \langle c\ a\ b\rangle$. Thus, view S_1 as a new given trajectory to predict. Next, repeat the above process until the postfix sequence $\langle a\ b\rangle$ is found, and after the step of trajectory matching, we can find a frequent sequential pattern $\langle a\ b\ d\rangle$ having prefix $\langle a\ b\rangle$. Lastly, output the predicted partial trajectory $\langle d\rangle$. In the phase of trajectory matching, we adopt three strategies [32] which will be introduced in the following.

In the PrefixTP algorithm, trajectory matching is used to predict the unknown locations of moving objects after frequent trajectory patterns have been found. The given trajectory of each moving object has to be compared with movement rules generated from frequent trajectories. In this study, we apply three matching strategies for ranking a partial trajectories over the database of movement rules. Let $X = (x_1, x_2, \ldots, x_m)$ be a partial trajectory of a moving object, for which we are seeking its most probable location.

a) Complete matching: The complete matching strategy consists in finding all movement rules $R = X \oplus Y$, where X is a complete frequent sequential pattern, and $Y = (y_1, y_2, \ldots, y_n)$, which represents n continuous location points. The partial trajectory Y can be used to predict a probable trajectory of a moving object.

b) Tail matching: The tail matching strategy does not need to consider other information from the partial trajectory X except for the last visited item, x_m. The method can discover the movement rules $R = x_m \oplus Y$, where x_m is a frequent item, and Y is a frequent postfix sequence. The result of this strategy is a list of items ordered by descending values of the support count. The tail matching strategy can discover movement rules even for very short partial trajectories.

c) Longest tail matching: The longest tail matching strategy is a compromise between the above two methods. With regard to a given trajectory X, then this method finds all movement rules $R = X' \oplus Y$, where $X' \sqsubseteq X$ is a part of trajectory X, and X' and Y are frequent trajectory patterns. The method returns the movement rule weighted by the relative coverage of X.

D. Algorithm Description

Based on the aforementioned description, a new prefix-projection based TP algorithm for connected vehicles called PrefixTP is shown in Algorithm 1.

Algorithm 1 contains the following steps: (1) obtain the first point s_0 from a given trajectory s and employ the function **getPrefix** to obtain frequent sequential patterns having prefix s_0. If this fails, choose the subsequence of s, starting from the next trajectory point and give that as input for the **PrefixTP** algorithm to predict recursively (lines 1-3). (2) Each obtained frequent sequential pattern p in P is traversed (line 5), where P is a set of frequent patterns having prefix s_0, and it is determined if the length of p is larger than that of s plus n. If not, p represents an invalid sequence which does not meet the requirement needed for prediction (lines 6-7). Then, another candidate sequence from P is chosen to match s, if that fails, then the loop is exited, where i represents the timestamp of a trajectory sequence (lines 8-13). If the length of p meets the requirement, then p is added to the result set P' (lines 14-15). (3) The frequent sequential patterns in P' are sorted by the support count in descending order (line 16). If P' is empty, then the subsequence of s is chosen and

Algorithm 1 Prefix-Projection Based TP Algorithm for Connected Vehicles

Input: A given trajectory sequence s w.r.t. a vehicle, the number of prediction steps n.
Output: The most possible trajectory.

1. $P \leftarrow$getPrefix(s_0);
2. **if** $P = \emptyset$ **then**
3. **return** PrefixTP($s.postfix(1)$, n);
4. $P' \leftarrow \emptyset$;
5. **for** each $p \subset P$ **do**
6. **if** $p.len < (s.len + n)$ **then**
7. **continue**;
8. $i \leftarrow 0$;
9. **while** $i < s.len$ **do**
10. **if** $p_i \neq s_i$ **then**
11. **break**;
12. $i = i + 1$;
13. $match(p, s)$ //perform trajectory matching strategies
14. **if** $i = s.len$ **then**
15. $P'.add(p)$;
16. P'.sort();
17. **if** $P' = \emptyset$ **then**
18. **return** PrefixTP($s.postfix(1)$, n);
19. **output** $P'_0.postfix()$;

used as input for the **PrefixTP** algorithm to mine recursively (lines 17-18). (4) Lastly, the postfix sequence (having prefix s) which is ranked first (denoted by P'_0) having the maximum support count in P' (line 19) is given as output.

1) Analysis: The correctness and completeness of the algorithm can be justified by Lemma 1 and 2. We analyze the efficiency of the algorithm as follows.

(1) **No candidate sequence needs to be generated by PrefixTP.** PrefixTP only grows longer trajectory patterns from the shorter frequent ones. It does not generate any candidate sequence nonexistent in a projected set.
(2) **Projected sets keep shrinking.** A projected set is smaller than the original one because only the postfix sequences of a frequent prefix are projected into it.

It is worthwhile to note that the number of prediction steps can be specified by users. For example, if n is designated to be a large number, i.e., $n \geq 5$, this implies PrefixTP will be a long-term TP algorithm. In general, relative to traditional TP algorithms, e.g., the HMM-based TP approach [2], if we want to obtain a high accuracy of prediction, n must be less than 5. On the contrary, PrefixTP algorithm can obtain a high prediction accuracy for long-term trajectory prediction, and we will prove this point in Section VI-B3.

VI. Experiments

A. Experimental Setup

In the following experiments, trajectory datasets were generated by 33,000 GPS-enabled connected taxis in a vehicle cloud over a period of 3 months [33]. By the V2V networks of taxis, the drivers can share traffic and road information in order to improve the quality of driving services. A more detailed description of the dataset is given in Table III.

TABLE III
Description of GPS Data

Parameter	Value
Num. trajectories	$4,960,000$
Num. GPS points	$790,000,000$
Total distance (km)	$400,000,000$
Average sampling interval(min.)	3.1

In experiments, we partition GPS data into the training set and testing set, and the training set contains 80% portion of data and we randomly select different trajectories from the remaining 20% portion of testing data to predict trajectories.

All algorithms were implemented on Eclipse Juno IDE, using the Java programming language. The hardware environments included a Intel(R) Core(TM)2 Duo P8700 2.53GHz CPU, with 3.0GB RAM.

In order to demonstrate the performance of the proposed algorithm, we use the evaluation measure of prediction accuracy (abbreviated as **PA**) which is defined below [2].

Definition 10 (Hit Rate): Given a trajectory sequence $S = \{s_1, s_2, \ldots, s_k\}$, the predicted trajectory sequence $T = \{t_1, t_2, \ldots, t_n\}$ over S, where $k < n$. $dist(m, n)$ represents the Euclidean distance between points m and n, and θ is a distance threshold. Then,

The formula $dist(s_i, t_i) < \theta$ implies a single hit in prediction, the hit rate is defined as follows:

$$H(s_i, t_i) = \begin{cases} 1 & \text{if } dist(s_i, t_i) < \theta \\ 0 & \text{if } dist(s_i, t_i) > \theta \end{cases} \tag{1}$$

Definition 11 (Prediction Accuracy): Given a trajectory sequence S and a predicted trajectory sequence T, the prediction accuracy is defined to be:

$$Accuracy = \frac{\sum_{i=1}^{n} H(s_i, t_i)}{|T|}, \quad s_i \in S \tag{2}$$

where $|T|$ represents the length of points in T.

It is worthwhile to notice that if a matched frequent trajectory pattern S is not found, which implies we cannot find a predicted trajectory sequence T, so the given trajectory sequence S will not be taken into consideration.

1) Comparison Algorithm: In our background research we have encountered many examples in which Markov models have been used to discover frequent trajectory patterns. The state-of-the-art work is a self-adaptive parameter selection TP approach via hidden Markov models [2], which has a high accuracy of prediction with the guarantee of exceptional time performance. In this study, we implement two kinds of HMM-based TP approaches: 1st-order Markov model which implies the position of the i^{th} timestamp is only determined by the state of $(i-1)^{th}$ timestamp, 2nd-order Markov model (representing higher-order Markov model) which implies the position of the i^{th} timestamp is determined by the state of $(i-1)^{th}$ and $(i-2)^{th}$ timestamps.

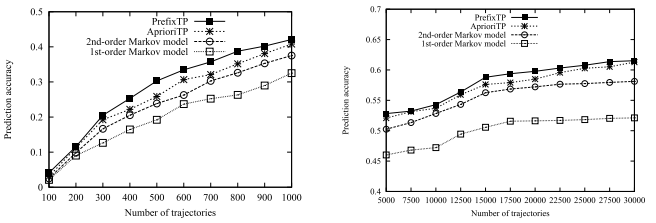


Fig. 3. Prediction accuracy comparison among different algorithms. (a) Small dataset. (b) Large-scale dataset.

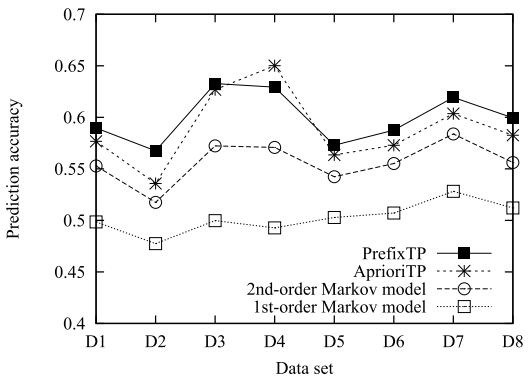


Fig. 4. Prediction accuracy comparison under different datasets.

In addition, we implement a Apriori-like TP algorithm, called AprioriTP. The difference between PrefixTP and AprioriTP lies in AprioriTP has to generate all the possible permutations of items in a trajectory sequence and needs multiple scans of trajectory databases. Because PrefixTP only grows longer sequential patterns from shorter ones, it does not generate nonexistent in a projected set which is smaller than the original one, the number of database scan by PrefixTP can be reduced when comparing to AprioriTP algorithm. In the following section, we will evaluate the effectiveness and efficiency of these TP algorithms against PrefixTP.

B. Prediction Accuracy Comparison and Analysis

1) Accuracy Comparison Under Different Training Data: In this set of experiments, we begin by first observing the TP of algorithms in a small and a large-scale trajectory dataset. The results are presented in Fig. 3, where x-axis represents the number of trajectories in the training set, and y-axis is PA.

According to Fig. 3, the following can be observed: (1) PA of different TP algorithms grow with the number of trajectories in the training set, and PrefixTP outperforms HMM-based TP approaches. PA can be, in average, improved by 49.6% and 24.5% on the small dataset, and improved by 15.9% and 4.6% on the large-scale dataset when compared to 1st-order Markov and 2nd-order Markov models, respectively. This is because 1st-order and 2nd-order Markov prediction models only take into consideration the influence of its previous one and two states, which does not utilize the complete historical behavior of moving objects. Accordingly, as the number of trajectories grow, the improvements of PA w.r.t. HMM-based TP approaches are limited. Whereas, PrefixTP discovers frequent sequential trajectory pattern with different length of trajectories, that is, all movement states in a complete trajectory are taken into consideration, so it can achieve a higher PA value. (2) The PA of PrefixTP is slightly higher than that of AprioriTP. The evidence for this claim is that AprioriTP may generate several candidate sequential patterns, and the average prediction accuracy will degrade duo to the large number of candidate trajectories. However, AprioriTP outperforms 1st-order Markov and 2nd-order Markov models. Because its working principle is similar to PrefixTP, both aim to mine frequently occurring sequential patterns.

2) Generality Evaluation: In order to validate the generality of PrefixTP, we divided the dataset from T-drive project [33] by Microsoft Research Asia into eight categories based on drivers' experiences, and observed the PA among these four algorithms. The experimental results are shown in Fig. 4.

By Fig. 4, it can be seen that PrefixTP preforms better in all categories of trajectory datasets when compared with HMM-based TP algorithms. The average gaps between PrefixTP and 1st-order as well as 2nd-order Markov prediction models are 19.5% and 7.8%, respectively, which strongly suggests that Prefix can be generalized to various TP scenarios including connected and unconnected V2V networks. The evidence for this claim is prediction performance based on the real movement behavior extracted from GPS data by mining frequent prefix-projection sequential patterns. Thus, it can be concluded with some level of certainty, that PrefixTP can generalize effectively, within diverse TP environments, and forecast short-term and long-term moving behaviors without compromising on performance. In addition, we find the PA values of PrefixTP and AprioriTP are similar, even AprioriTP outperforms PrefixTP on dataset D4. Because the trajectories in dataset D4 contain more regular movement behaviors than other datasets, AprioriTP can discover more frequent trajectory sequential patterns than PrefixTP from dataset D4, and this is a disadvantage for prefixTP algorithm. We can also find that HMM-based TP algorithms, i.e., 1st and 2nd-order Markov models, produce lower prediction accuracy by comparing with PrefixTP and AprioriTP algorithms. This is because HMM-based TP approaches employ the conditional probability equation $p_{ij}(n) = P\{X_{m+n} = j | X_m = i\}$, $(m \geq 0, n \geq 1)$ to calculate the probability of transition from one state to another one, which is not appropriate for randomly generated trajectory data having relatively few movement rules. However, PrefixTP and AprioriTP can be used to discover different kinds of trajectory data within diverse environments.

3) Estimation of n-Steps Prediction: Long-term prediction is a challenging and ongoing research problem in trajectory prediction. In this set of experiments, the PA of distinct algorithms for n-step predictions are evaluated. Essentially, this refers to predicting the next n-step locations. For this set of experiments, 1,000 trajectories were randomly selected from the testing set, and the average PA value was used to evaluate the performance of the given algorithms. The results are presented in Fig. 5, where x-axis represents the number of prediction steps, and y-axis represents PA.

As we can see from Fig. 5 that: (1) PrefixTP outperforms 1st-order Markov and 2nd-order Markov models with the

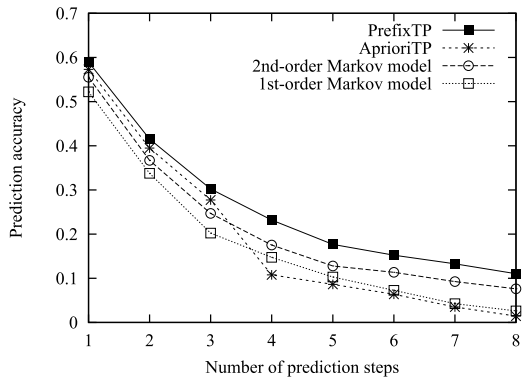


Fig. 5. Prediction accuracy comparison by n-steps among algorithms.

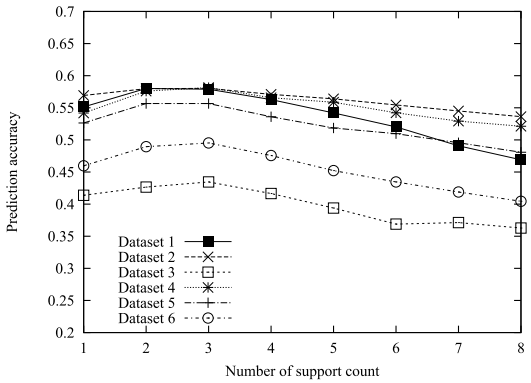


Fig. 6. Prediction accuracy of PrefixTP under different datasets.

number of prediction steps. The longer the time interval of prediction, the bigger the gap between PrefixTP and HMM-based TP approaches, with an average gap of 107.4% and 29.4%, respectively. This is primarily because PrefixTP selects candidate locations that satisfy the requirement of minimum support count, and frequent sequential patterns with variable lengths are mined based on the prefix-projection approach, which requires that the candidate postfix sequence mined in each iteration is frequent. In contrast, HMM-based TP methods only take into account the effect of fixed length trajectories in the training set, which is not appropriate for randomly generated GPS data with variable lengths. (2) As the number of prediction step $n < 4$, AprioriTP wins the HMM-based TP algorithms, however, when n is designated to be a large value, its PA drops drastically, and AporioriTP obtains the worst prediction accuracy. The evidence for this claim is AprioriTP can generate a large volume of candidate sequences when predicting long-term trajectory sequential patterns, which will greatly degrade its performance. (3) By experiments, we find there are less frequent trajectory sequential patterns of connected vehicles whose length are longer than four steps, and this phenomena agrees with the real-world situation. Because drivers usually predict one or two steps to future destination, and the length of prediction steps is no longer than three. For long-term prediction, the number of frequent trajectory sequential patters is small, which implies there is not enough knowledge for connected vehicles to forecast future locations, so for long-term predictions there is a low prediction accuracy for all algorithms.

4) Effect Analysis of Support Count on Accuracy: For the PrefixTP, the support count represents the number of a trajectory sequential patterns in a training set, and the selection of this parameter will have an effect on the accuracy of prediction. In this set of experiments, the PA of PrefixTP was observed, under different training sets, and as the support count was increased. The results are given in Fig. 6, where x-axis is the number of support counts, and the y-axis represents PA.

By Fig. 6, the PA of PrefixTP, when applied to different datasets, fluctuates as the number of support counts grow. A trend characterized by a rising initially, and then a fall, in all datasets, was observed. We can find that PrefixTP maintains a high PA value when the support count is set between two and four. Empirically, based on the given results, it can concluded that an optimal support count will have a positive effect, an increase, in the number of trajectories in a training set. If the support count is enlarged, which means the requirement of frequent trajectory patterns is more strict, and discovered trajectory sequences need to be more frequent, then the number of discovered trajectory sequences is less than the case when the support count value is small, so the prediction accuracy is lower when the support count is enlarged.

TABLE IV
PREDICTION TIME COST OF PREFIXTP

Num.	1000	2000	3000	4000	5000	6000
Time(ms)	105	293	346	520	646	778

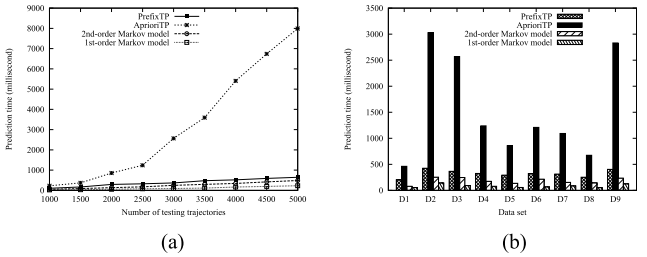


Fig. 7. Prediction time comparison of different algorithms. (a) Different number of trajectories. (b) Different data sets.

C. Time Performance Analysis

Prediction time is an important metric in real-time location tracking and forecasting. In the given series of experiments, the prediction time of various algorithms are observed as the number of trajectories increased. Results are presented in Table IV. The response time is only 0.778 seconds for forecasting 6,000 trajectories, which suggests that PrefixTP has an exceptional runtime performance, and can satisfy the requirement for real-time prediction.

Additionally, in these experiments, we have compared the performance (relative to time) of PrefixTP with AprioriTP and HMM-based TP approaches. Fig. 7(a) shows the cost in time of various relevant algorithms as the number of trajectories increase in the testing set. We also compare the efficiency of different algorithm on nine randomly selected testing datasets, and the results are shown in Fig. 7(b).

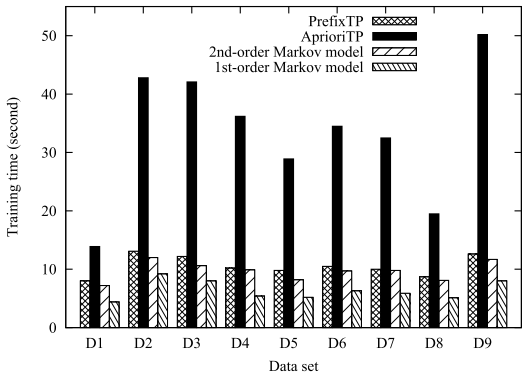


Fig. 8. Training time comparison on different datasets.

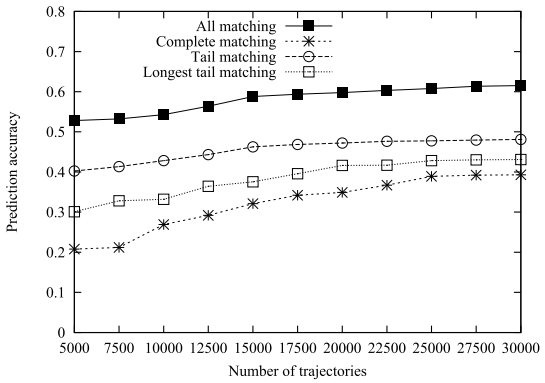


Fig. 9. Prediction accuracy comparison of different matching strategies by PrefixTP.

By Fig. 7, the time cost of PrefixTP is slightly higher than that of HMM-based TP models, but the overall time overhead of PrefixTP is maintained at the millisecond level, which can be acceptable for real-time forecasting systems. This slight increase in time can be explained by the recursion needed for mining frequent sequential patterns. However, as for AprioriTP algorithm, the time cost is very high, is about 5.65 times higher than that of PrefixTP according to Fig. 7(a). This is primarily because Apriori-like TP algorithm will generate a huge number of candidate sequences that makes the time grows rapidly for discovering frequent sequential patterns. Another reason is that AporioriTP needs to scan trajectory databases for multiple times which is time-intensive.

In order to further evaluate the time performance of these algorithm, we observe the training time of the algorithms on the previously used nine datasets given in Fig. 7(b), and the results are shown in Fig. 8.

By Fig. 8, we can see that the training time of PrefixTP is similar to that of 2nd-order Markov model, and is a little higher than that of 1st-order Markov model. This is because HMM-based prediction models spend much time on calculating the transition probability matrix of the upper layer of the trajectory chain and the transition probability matrix composed of probability values from hidden states to observable states [2], which is time consuming. However, the AprioriTP algorithm needs to spend much time on generating potentially huge set of candidate sequences.

D. Performance Evaluation of Matching Strategies

In Section V-C, we have applied three trajectory matching strategies to infer possible locations of objects, and we evaluate the prediction accuracy via different matching strategies by using the large-scale trajectory dataset in Section VI-B. The results are given in Fig. 9.

We can see that: by using all these three matching strategies (denoted by **All matching**), we can obtain the highest PA value, because PrefixTP can find all frequent sequential patterns by applying complete matching, tail matching and longest tail matching strategies. On the contrary, PrefixTP beyond complete matching strategy obtains the worst PA value, and the PA value of longest tail strategy is between complete and tail matching strategy. As aforementioned in Section V-C, this is because complete matching strategy is more strict than other two strategies, tail matching strategy can discover movement rules even for very short partial trajectory, and longest tail matching is a compromise between the above two strategies.

VII. Conclusion

This study has worked to propose a novel, scalable and effective trajectory sequential mining approach. The approach includes the development of a TP algorithm for connected vehicles, called PrefixTP, which examines only the prefix subsequences and projects only their corresponding postfix subsequences into projected sets. The systematic performance study demonstrates that PrefixTP is accurate at predicting long-term trajectory sequences, and can be generalized more effectively relative to various and differing trajectory data, when compared to HMM-based and Apriori-like TP approaches. In addition, the experiments demonstrate that PrefixTP can respond in real-time.

PrefixTP represents an innovative methodology for effectively predicting sequential patterns for connected vehicles. The phase for identifying subsets of trajectory sequential patterns is a recursive mining process, which can be time intensive. In future work, we will focus on reducing the size of the projected sets, such as applying a bi-level projection scheme for improving the performance of sequential patterns mining.

References

[1] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.

[2] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[3] W. Jiang, D.-S. Huang, and S. Li, "Random walk-based solution to triple level stochastic point location problem," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1438–1451, Jun. 2016.

[4] J. D. Mazimpaka and S. Timpf, "Trajectory data mining: A review of methods and applications," *J. Spatial Inf. Sci.*, vol. 2016, no. 13, pp. 61–99, Dec. 2016.

[5] S. Qiao, N. Han, W. Zhu, and L. A. Gutierrez, "TraPlan: An effective three-in-one trajectory-prediction model in transportation networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1188–1198, Jun. 2015.

[6] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 29, May 2015.

[7] J. Pei *et al.*, "Mining sequential patterns by pattern-growth: The PrefixSpan approach," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1424–1440, Nov. 2004.

[8] S. Qiao *et al.*, "PutMode: Prediction of uncertain trajectories in moving objects databases," *Appl. Intell.*, vol. 33, no. 3, pp. 370–386, Dec. 2010.

[9] J. Ge, Y. Xia, J. Wang, C. H. Nadungodage, and S. Prabhakar, "Sequential pattern mining in databases with temporal uncertainty," *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 821–850, Jun. 2017.

[10] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "WhereNext: A location predictor on trajectory pattern mining," in *Proc. ACM SIGKDD*, Jul. 2009, pp. 637–646.

[11] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and S. V. Tseng, "Semantic trajectory mining for location prediction," in *Proc. ACM SIGSPATIAL GIS*, Nov. 2011, pp. 34–43.

[12] N. J. Goodall, "Real-time prediction of vehicle locations in a connected vehicle environment," Virginia Center Transp. Innov. Res., Charlottesville, VA, USA, Tech. Rep. FHWA/VCTIR 14-R4, Dec. 2013.

[13] S. M. Khan, K. C. Dey, and M. Chowdhury, "Real-time traffic state estimation with connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1687–1699, Jul. 2017.

[14] B. Wang, Y. Hu, G. Shou, and Z. Guo, "Trajectory prediction in campus based on Markov chains," in *Proc. BigCom*, Jul. 2016, pp. 145–154.

[15] N. Ye, Y. Zhang, R. Wang, and R. Malekian, "Vehicle trajectory prediction based on hidden Markov model," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 7, pp. 3150–3170, Jul. 2016.

[16] H. Jeung, H. Shen, and X. Zhou, "Mining trajectory patterns using hidden Markov models," in *Proc. DaWaK* vol. 4654. Sep. 2007, pp. 470–480.

[17] A. Asahara, A. Sato, K. Maruyama, and K. Seto, "Pedestrian-movement prediction based on mixed Markov-chain model," in *Proc. ACM SIGSPATIAL GIS*, Nov. 2011, pp. 25–33.

[18] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[19] Y. Sekimoto *et al.*, "Real-time people movement estimation in large disasters from several kinds of mobile phone data," in *Proc. UbiComp*, Sep. 2016, pp. 1426–1434.

[20] D. Lee, C. Liu, Y.-W. Liao, and J. K. Hedrick, "Parallel interacting multiple model-based human motion prediction for motion planning of companion robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 52–61, Jan. 2017.

[21] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F.-F. Li, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. CVPR*, Jun. 2016, pp. 961–971.

[22] T. L. Pan, A. Sumalee, R.-X. Zhong, and N. Indra-payoong, "Short-term traffic state prediction based on temporal-spatial correlation," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1242–1254, Sep. 2013.

[23] Z. Ding, B. Yang, R. H. Güting, and Y. Li, "Network-matched trajectory-based moving-object database: Models and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1918–1928, Aug. 2015.

[24] J. Xu, Y. Gao, C. Liu, L. Zhao, and Z. Ding, "Efficient route search on hierarchical dynamic road networks," *Distrib. Parallel Databases*, vol. 33, no. 2, pp. 227–252, Jun. 2015.

[25] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. WWW*, Apr. 2009, pp. 791–800

[26] J. Zhou, A. K. Tung, W. Wu, and W. S. Ng, "A 'semi-lazy' approach to probabilistic path prediction in dynamic environments," in *Proc. ACM SIGKDD*, Aug. 2013, pp. 748–756.

[27] S. Gan, S. Liang, K. Li, J. Deng, and T. Cheng, "Long-term ship speed prediction for intelligent traffic signaling," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 82–91, Jan. 2017.

[28] H. Jin, J. Chen, H. He, and C. O'Keefe, "Privacy-preserving sequential pattern release," in *Proc. PAKDD*, May 2007, pp. 547–554.

[29] R. H. Weber, "Internet of Things—New security and privacy challenges," *Comput. Law Secur. Rev.*, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[30] A. Mhatre, M. Verma, and D. Toshniwal, "Privacy preserving sequential pattern mining in progressive databases using noisy data," in *Proc. 13th Int. Conf. Inf. Vis.*, Jul. 2009, pp. 456–460.

[31] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 241–246.

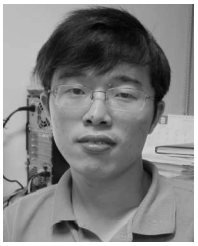
[32] M. Morzy, "Mining frequent trajectories of moving objects for location prediction," in *Proc. MLDM*, vol. 4571. Jul. 2007, pp. 667–680.

[33] J. Yuan *et al.*, "T-drive: Driving directions based on taxi trajectories," in *Proc. ACM SIGSPATIAL GIS*, Nov. 2010, pp. 99–108.

Shaojie Qiao received the B.S. and Ph.D. degrees from Sichuan University, Chengdu, China, in 2004 and 2009, respectively. From 2007 to 2008, he was a Visiting Scholar with the School of Computing, National University of Singapore. He is currently a Professor with the School of Cybersecurity, Chengdu University of Information Technology, Chengdu. He has led several research projects in moving objects databases and trajectory data mining. He authored over 40 high-quality papers and co-authored over 90 papers. His research interests include trajectory prediction and intelligent transportation systems.

Nan Han received the M.S. and Ph.D. degrees from the Chengdu University of Traditional Chinese Medicine, Chengdu, China. She is currently a Lecturer with the School of Management, Chengdu University of Information Technology, Chengdu. Her research interests include trajectory prediction and data mining. She has authored 20 papers. She participated in several projects supported by the National Natural Science Foundation of China.

Junfeng Wang received the M.S. degree in computer application technology from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2001, and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2004. He is currently a Professor with the School of Aeronautics and Astronautics, Sichuan University, Chengdu. His current research interests include data mining and spatial information networks.



Rong-Hua Li received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2013. He is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His research interests include social network analysis, graph data management, and sequence data management and mining.

Louis Alberto Gutierrez received the Ph.D. degree in computer science from the Rensselaer Polytechnic Institute, Troy, USA, in 2014. He was a National Science Foundation GK-12 Fellow, a Mickey Leland Energy Fellow, and a CHCI 2012 Scholar. His research areas include big data noise reduction, social computing, and mobile technologies.

Xindong Wu (F'11) received the Ph.D. degree in artificial intelligence from The University of Edinburgh, in 1993. He is currently a Professor of computer science with the University of Louisiana at Lafayette, Lafayette, USA. His research interests include data mining and knowledge-based systems. He is a fellow of the AAAS. He is the Editor-in-Chief of the *Knowledge and Information Systems*, and the *Advanced Information and Knowledge Processing*.