Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 717095, 13 pages
http://dx.doi.org/10.1155/2015/717095

Research Article

Map Matching Based on Conditional Random Fields and Route Preference Mining for Uncertain Trajectories

Ming Xu,[1] Yiman Du,[2] Jianping Wu,[2] and Yang Zhou[2]

[1] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *School of Civil Engineering, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Yiman Du; ymducp@gmail.com

Received 14 January 2015; Revised 26 March 2015; Accepted 26 March 2015

Academic Editor: Jurgita Antucheviciene

In order to improve offline map matching accuracy of uncertain GPS trajectories, a map matching algorithm based on conditional random fields (CRF) and route preference mining is proposed. In this algorithm, road offset distance and the temporal-spatial relationship between the sampling points are used as features of GPS trajectory in a CRF model, which integrates the temporal-spatial context information flexibly. The driver route preference is also used to bolster the temporal-spatial context when a low GPS sampling rate impairs the resolving power of temporal-spatial context in CRF, allowing the map matching accuracy of uncertain GPS trajectories to get improved significantly. The experimental results show that our proposed algorithm is more accurate than existing methods, especially in the case of a low-sampling-rate.

1. Introduction

In recent years, the prevalence of GPS-enabled devices has resulted in incredible amounts of vehicle trajectory data, which records human mobility and reflects the dynamic characteristics of a city. In general, trajectory data consisting of time-stamped locations come from GPS sensors embedded vehicles or other mobile devices. However, the raw trajectory data are not completely reliable; that is, the locations indicated in such a trajectory are not the real accurate locations of vehicles or mobile devices, since the measurement errors of the device and the complex communication environment make the readings of GPS sensors frequently deviate from the actual positions. Therefore, map matching, that is, the process of adjusting the trajectory points to real road segments, is very important to some location-based services and applications and has got considerable attention by the researchers.

There are two types of map matching approaches, that is, online and offline map matching. For online map matching, real-time route navigation, for example, although accuracy is concerned, prompt speed of map matching is the most needed. For offline map matching, the most concerned is obtaining accurate matched paths which can be used for future works (e.g., knowledge mining); therefore instantaneity is not important for offline map matching while accuracy is required as much as possible.

Offline map matching is widely employed by many applications, such as pickup location recommendation [1], ridesharing services [2, 3], regional function analysis [4], urban planning analysis [5], abnormal event detection [6–8], travel time estimation [9, 10], real-time traffic flow prediction [11], and traffic guidance [12, 13]. One of the biggest challenges of offline map matching is the uncertainty caused by low-sampling-rates; that is, it is difficult to determine the best matching. In fact, low-sampling-rates are often encountered in map matching; for example, for energy consumption considerations, GPS sampling intervals of most vehicles are more than 30 seconds, and even a considerable number of trajectories' sampling intervals exceed two or three minutes. Consider the example in Figure 1, which compares two trajectories. The sampling interval of one is twenty seconds, while that of another one is three minutes. Obviously, the path of the high-sampling-rate trajectory can be identified clearly, while the path of the sparse trajectory is difficult to determine. In other words, the low-sampling-rate results in the uncertainty of map matching. Such uncertainty produces

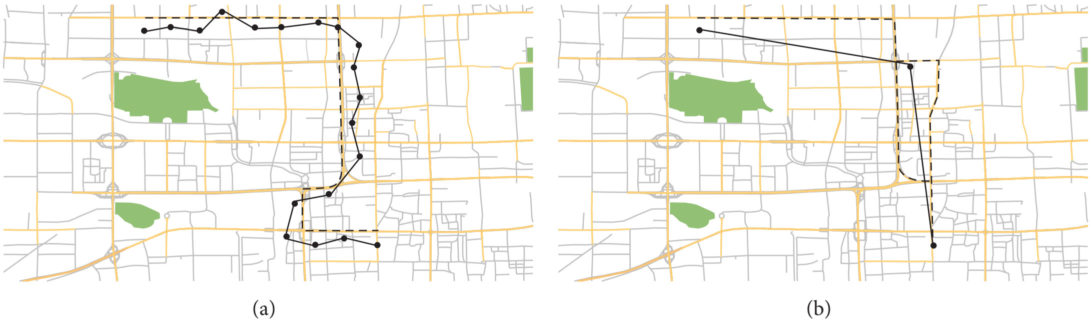


Figure 1: A comparison between high-sampling-rate and low-sampling-rate map matching.

severe effects on the abovementioned trajectory mining-based applications. However, some valuable pieces of information in the trajectories, such as temporal context and drivers' route preference, are neglected by most existing approaches of offline map matching. The instability of these approaches may be caused when dealing with low-sampling-rate trajectories.

In this paper, we explore a way that can reduce the map matching uncertainty to accurately determine a path for a low-sampling-rate trajectory. To achieve this, we make full use of available information, such as trajectory characteristics, road network topology, and driver route history. We treat offline map matching of GPS sampling points as a sequence-labeling problem [14] in machine learning and propose a conditional random field (CRF) [15] map matching algorithm employing spatial context and temporal context between sampling points. There also exist other probabilistic approaches suitable for the sequence-labeling problem, such as hidden Markov model (HMM) [16] and maximum entropy Markov model (MEMM) [17]. However, the HMM assumes that all trajectory points are conditionally independent, and the current state depends only on the previous state in the process of state transition. This assumption may lose some significant context and result in a "labeled bias" problem [18]; that is, the HMM model tends to match sampling points to long-distance road segments that have few intersections. Although the MEMM avoids the conditional independence assumptions on trajectory points by improving the structure of a probabilistic graphical model, it also suffers from the "labeled bias" problem. Therefore we select CRF to overcome the drawbacks of the HMM and the MEMM. CRF has obvious advantages in flexible integration of a variety of features and context information. In addition, we find that most drivers usually choose familiar paths to travel. Inspired by this, we employ driver route preferences to improve the quality of the algorithm. Specifically, if the GPS sampling frequency of given trajectory is too low to guarantee the effectiveness of CRF features, we extract the route preferences from historical well matched paths and superpose them on the features of CRF with appropriate weighting. We have conducted lots of experiments to verify that our algorithm can match the trajectory points to actual paths.

The contributions of our paper are threefold.

(1) We propose a CRF-based algorithm of map matching that combines spatial and temporal features effectively. Experimental results show that temporal features can boost the resolving power of existing CRF models that solely employ spatial features, especially in areas with similar spatial features, such as interaction-dense areas.

(2) We design a framework for mining the drivers' route preferences from historically matched paths to improve the effectiveness of our CRF-based algorithm, which can weaken the effect of the uncertainty caused by low-frequency-sampling.

(3) We perform extensive experiments on a real trajectory dataset collected from the physical world and labeled manually. Our algorithm is evaluated for matching accuracy. The results show that our algorithm outperforms previous methods significantly on low-sampling-rate trajectories.

The remainder of the paper is organized as follows. A review of some existing map matching methods is given in Section 2. Section 3 presents our map matching framework and problem definition. The algorithm is proposed with detailed discussion and analysis in Section 4. Section 5 presents the experimental evaluation. We conclude the paper in Section 6.

2. Related Work

Knowledge discovery in sets of moving objects attracts many researchers, and a large number of approaches have been proposed. These studies can be divided into two categories: (1) models of moving objects, which can move freely in Euclidean space [19]; (2) models of objects, whose movements are subject to spatial constraints, such as road networks. The method proposed by this paper is related to the second research category and can be applied in transportation, urban planning, location-based services, activity recognition [20, 21], and so forth. Indeed, if we know the geometry and the topology of the network, we can represent a trajectory by a list of traversed road segments; such a process is called "map matching." Depending on the application scenario, these methods can be divided into two categories: online and offline.

Online algorithms mainly employ a greedy strategy to search for the optimal local matching from an existing solution. Greenfeld [22] proposes an incremental algorithm that considers only geometric information to evaluate each candidate edge. This information contains distance similarity and orientation similarity. Chawathe [23] proposes a segment-based method, in which a confidence score is defined and assigned to different sampling points. When a new trajectory appears, high-confidence edges are matched first, and then low-confidence edges are matched, according to already matched edges. Wenk et al. [24] propose an "adaptive clipping" approach that obtains the shortest path on a local free space graph. These kinds of methods can identify matching segments quickly due to using only a small part of the trajectory; for this reason, they are widely used for online applications, such as navigation systems. However, the accuracy of these algorithms drops sharply when sampling frequency decreases.

Offline algorithms handle the entire trajectory after completing a trip. Most studies detect the closest candidate roads from the current trajectory by means of Fréchet distance or its extended metrics; its underlying meaning is that continuity of curves is taken into account to search for corresponding paths. In the algorithm proposed by Alt et al. [25], the critical values are worked out in a parametric search process, and then the Fréchet distance is measured by finding a monotone path in the free space from the lower left corner to the upper right corner. In order to reduce the effect of anomalous sampling points in this work, Brakatsoulas et al. [26] propose an extended algorithm using average Fréchet distance; in addition, weak Fréchet distance is used in their work to reduce the time cost to $O(mn \log mn)$. Yin and Wolfson [27] model road networks using weight graphs, and their proposed algorithm is based on edit distance, which is similar to average Fréchet distance. However, these deterministic algorithms are susceptible to noise and perform worse with low-sampling-rates.

To deal with noise, low-sampling-rates, and other issues effectively, methods based on probabilities are widely used. Lou et al. [28] propose an ST-Matching algorithm that combines temporal and spatial contexts. At first, the candidate roads of each sampling point in the given trajectory are determined according to Euclidean distance between the current point and each candidate road, and then spatial and temporal analysis is used to calculate observation probability and transition probability. After accumulating probability scores, the path, which has the maximum joint probability, would be considered the matched path. However, this method does not take into account the weights of different factors and interaction between nonneighboring points, so the accuracy falls rapidly when the path is too long or when there are multiple lanes in the area. Newson and Krumm [16] propose a map matching algorithm for low-sampling-rate trajectories based on a hidden Markov model (HMM). In their work, observation probability matrix and transition probability matrix are inferred by learning from the training dataset, and then a Viterbi algorithm [15] is used to get the result. But HMM has a too strict independence assumption, ignoring the impact between points over long distances and with nonorthogonal features, so its accuracy is slightly lower than that of ST-Matching. Liao et al. [29] also suggest a CRF-based algorithm, but since it employs only the spatial context, the sampling points tend to match roads on the shortest path, and it is not suited for low-sampling-rate trajectories.

3. System Overview

3.1. Problem Definition. In this section, we give definitions of some terms used in this paper.

Definition 1. A *trajectory* θ is a sequence of GPS point which is generated by a vehicle in a trip. Formally, $\theta = o^{(1)} \rightarrow o^{(2)} \rightarrow \cdots \rightarrow o^{(T)}$, where T is the total number of sampling points in the GPS trajectory, the symbols $o^{(1)}$ and $o^{(T)}$ denote start point and end point, respectively. Each GPS point $o^{(t)}$ can be represented by a three-tuple $\langle x, y, t \rangle$, where x denotes latitude, y denotes longitude, and t denotes timestamp. Let Θ denote the collection of N trajectories.

Definition 2. A *trip* tr is an origin-destination pair (OD), which is represented by a two-tuple $\langle r_o, r_d \rangle$, where r_o denotes an origin road and r_d denotes a destination road. If the origin and destination roads of two trips, tr_A and tr_B, are the same, $\mathrm{tr}_A = \mathrm{tr}_B$ can be considered tenable.

Definition 3. A *path* γ is a road segments sequence which is traversed by a vehicle in one trip. $\gamma = r^{(1)} \rightarrow r^{(2)} \rightarrow \cdots \rightarrow r^{(T)}$, where $r^{(1)}$ denotes an origin road and $r^{(T)}$ denotes a destination road. For $r^{(t)} \in \{r_w\}_{w=1}^{W}$, the symbol w is the ID of a road segment, and W is total number of road segments in the road network. If any two neighboring road segments of a path are different and topologically connected, that path is called a *complete* path.

Map matching is equivalent to a sequence-labeled problem in machine learning. Given an observable GPS trajectory θ, a sampling points sequence o can be regarded as an observation sequence to be labeled; road set $\{r_w\}_{w=1}^{W}$ can be regarded as the label set. The objective is to find a path γ^* that is an optimal match for trajectory θ, where γ^* is essentially a maximum a posteriori probability path; that is, $\gamma^* = \mathrm{argmax}_\gamma p(\gamma \mid \theta)$.

3.2. System Framework. The framework of our proposed algorithm is presented in Figure 2. It consists of a training phase and a prediction phase. In the training phase, the parameters of the CRF model are inferred by learning from labeled trajectories. In the prediction phase, generative features and transition features are extracted from the given trajectory to establish the CRF model, and then the value of average sampling interval f is checked; if f is not lower than a given threshold, the matched path is obtained by using the CRF model directly. In detail, the conditional probability of matching a road segment given a sampling point equals

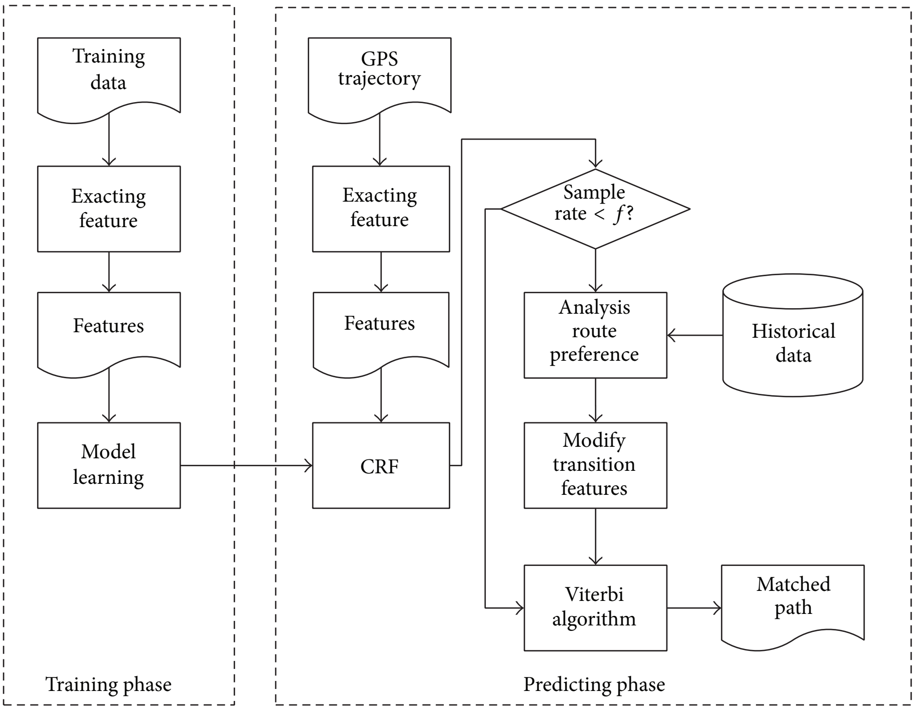


Figure 2: The framework of proposed map matching algorithm.

the sum value of all the features, and then the matched path with maximum joint conditional probability is worked out. Otherwise, the sampling rate is considered too low to capture the temporal and spatial correlation; in order to improve matching accuracy, route preference information is extracted from historical trajectories of the corresponding vehicle, and then it is superposed, with appropriate weighting, on transition features of CRF to generate new transition features.

4. Proposed Approach

4.1. CRF-Based Approach. As a special case of undirected graphical models, a conditional random field is developed on the basis of the maximum entropy model; this approach is widely used in sequence-labeled problems. It avoids the labeled bias problem in the maximum entropy model. Detailed information about CRF can be found in [18, 30].

Definition 4. Let X and Y be sets of observations and random variables, respectively, and $p(Y \mid X)$ the conditional probability of Y given X. If the set Y of random variables constitutes an undirected graph $G = (V, E)$ satisfying the Markov property, that is to say, $p(Y_v \mid X, Y_w, w \neq v) = p(Y_v \mid X, Y_w, w \sim v)$ is founded on any node v in the graph, where $w \sim v$ denotes all the nodes connecting node v in graph G, then $w \neq v$ denotes all the nodes except node v. The probability distribution $p(Y \mid X)$ is called *condition random fields*.

A sequence-labeled problem can be modeled using a linear CRF. If we denote the values assigned to observation sequence X by a vector x, the vector y assigned to labeled sequence Y has the following form:

$$p(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_{i+1\in V}\left(\sum_j \mu_j \varphi_j(y_i, x) + \sum_k \lambda_k \delta_k(y_i, y_{i+1}, x)\right)\right), \tag{1}$$

where $Z(x)$ is the normalized factor, which is used to convert $p(y \mid x)$ to a valid probability and which is defined as the sum of exponential number of sequences:

$$Z(x) = \sum_y \exp\left(\sum_{i+1\in V}\left(\sum_j \mu_j \varphi_j(y_i, x) + \sum_k \lambda_k \delta_k(y_i, y_{i+1}, x)\right)\right). \tag{2}$$

The function $p(y \mid x)$ belongs to an exponential family, which is a set of probability distribution functions of a common form. The function $\varphi(y_i, x)$ is the potential function defined for the nodes in the graph model. This function is also called a generative feature, which makes a different effect of observation sequence x on the probability that label sequence y occurs. If $\mu > 0$, the bigger the value $\varphi(y_i, x)$ is, the more the model prefers to label y_i for x_i. The function $\delta(y_i, y_{i+1}, x)$ is the potential function corresponding to the edges that link nodes and model the dependencies between two labels. It is called a *transition feature*. That is to say, in labeling y_{i+1}, both x and previously labeled y_i would be considered. The vectors μ and λ are parameters of the CRF and control the weights of their potential functions. All the potential functions and parameters can take arbitrary real values, and the entire exp() function will be nonnegative. The local information of the graph is modeled using potential functions, and such context information can be propagated through the edges linking the nodes; therefore, CRF can integrate a wide range of context information.

With respect to the map matching problems, the road segment that is matched by current sampling point of a vehicle may depend on this vehicle's previous locations. In theory, a high-order CRF that integrates long-distance context dependencies should be used in the model to achieve high accuracy. However, inferring the parameters for a high-order CRF requires complicated computation. In this paper, we use a simple first-order linear CRF model, in which only the dependency between road segments of neighboring sampling points is considered. Our experiments verify the validity of modeling in this way.

4.2. Feature Selection. In this section, we give the concrete expression of each feature function. As mentioned in Definition 1, generative feature is determined only by the current GPS point without considering the effect of other road segments. According to [16, 28, 29], error of GPS points follows Gaussian distribution $N(0,\sigma^2)$. This is in accordance with our intuition that a GPS point is more likely to match to the nearest road segment. Therefore, the formula of generative feature is given by

$$\varphi\left(r_m^{(t)}, o^{(t)}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_p^2\left(r_m^{(t)}, o^{(t)}\right)}{2\sigma^2}\right), \tag{3}$$

where $r_m^{(t)}$ denotes mth candidate road segment of GPS point $o^{(t)}$ at sampling time t and $d_p(r_m^{(t)}, o^{(t)})$ denotes the projection distance from $o^{(t)}$ to $r_m^{(t)}$. This formula is based on the assumption that the standard deviation σ is constant over the road network. This is not true in practice, due to urban canyoning effects [31] and satellite occlusions, and [32] uses geographical clustering of the regions of interest to estimate σ. However, this method is more complex. In this paper, we still consider that σ is invariant and set it to 20 meters by measuring on the real dataset. The experimental results show that the effect of the model is ideal. The generative feature ignores the relation of neighborhood points. So it is insufficient to match GPS point based only on generative feature. An example is given in Figure 3. The GPS point $o^{(t)}$ would be matched to the nearest road segment r_3 without considering its neighboring points $o^{(t-1)}$ and $o^{(t+1)}$. In fact, the road segment r_2 is correct.

The transition features model the possibility of jumping between the candidate road segments of two neighboring points. In general, drivers are unlikely to choose a detour path, and the candidate road segments of neighborhood GPS points should be adjacent or close to one another in spatial topology. Therefore, the spatial transition feature δ_1 is defined as

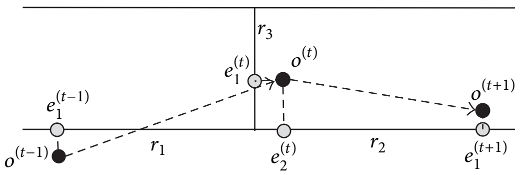


Figure 3: An example of mistake generated by merely matching the GPS points to the nearest road.

$$\delta_1\left(r_m^{(t)}, r_n^{(t+1)}\right) = \left[\frac{d\left(o^{(t)}, o^{(t+1)}\right)}{d_r\left(e_m^{(t)}, e_n^{(t+1)}\right)}\right]^2, \tag{4}$$

where $e_m^{(t)}$ is the projection point of $o^{(t)}$ on the mth candidate road. The functions $d()$ and $d_r()$ are used to calculate the Euclidean distance and path distance between two points, respectively. Obviously, $\delta_1() \in (0,1)$, and the smaller is the value of $\delta_1()$, the more roundabout is the path. The function $\delta_1()$ reflects the spatial dependency of a GPS point on its neighboring points. However, in some cases, the spatial context cannot be sufficiently reliable to determine the actual path, as when, for example, in some dense parts of a road network, the spatial features of multiple candidate road segments may be similar. In order to distinguish the actual path from other candidates, other high-resolving-power features should be integrated into our model. Consider that the speed constraint of a road segment may be different from that of others, for example, an expressway or a bypass; even with respect to the same road, the average speed can be different at different times. A GPS point is unlikely to be matched to the road segments, on which the corresponding vehicle exceeds the speed limit. We therefore introduce the temporal feature. Assume that there are k road segments in a subpath between $e_m^{(t)}$ and $e_n^{(t+1)}$; the historical average time Δt_e of traversing from $e_m^{(t)}$ to $e_n^{(t+1)}$ can be calculated by

$$\Delta t_e = \sum_k \frac{r_k \cdot l}{r_k \cdot v}, \tag{5}$$

where $r_k \cdot l$ denotes the distance of traversing on road segment k and $r_k \cdot v$ denotes the historical average speed of traversing road segment k at the same time slot. The temporal transition feature δ_2 is given as

$$\delta_2\left(r_t^m, r_{t+1}^n\right) = \left(\frac{\min\left(\Delta t, \Delta t_e\right)}{\max\left(\Delta t, \Delta t_e\right)}\right)^2. \tag{6}$$

As with $\delta_1()$, the temporal transition feature $\delta_2() \in (0,1)$. Due to taking the temporal feature into consideration, the model would give a higher probability of matching a GPS point to the candidate road whose average speed is closer to the speed of the trajectory.

In summary, the posterior probability of the matched path γ given a trajectory o can be presented as

$$p(\gamma \mid \theta) = \exp\Bigg(\sum_{t\le T} \mu\varphi\left(r_m^{(t)}, o^{(t)}\right) + \sum_{t+1\le T} \left(\lambda_1 \cdot \delta_1\left(r_m^{(t)}, r_n^{(t+1)}\right) + \lambda_2 \cdot \delta_2\left(r_m^{(t)}, r_n^{(t+1)}\right)\right)\Bigg). \tag{7}$$

The weighted coefficients μ, λ_1, and λ_2 are the model parameters, which are determined in the training phase.

4.3. Model Inference. The goal of model inference is to infer parameters $\omega = \{\mu, \lambda_1, \lambda_2\}$, which are independent of time t. A training set containing labeled trajectories is needed in this phase. Given the definition of conditional probability $p(\gamma \mid \theta)$, the optimization goal is to maximize the likelihood of the training data, and the logarithm likelihood function is given by $\ell(\omega) = \sum_l \log p(\gamma^{(l)} \mid \theta^{(l)}; \omega)$. A standard parameter learning process is to calculate the gradient of the objective function $\ell(\omega)$ and then use this gradient to search for the optimal solution. There are many algorithms for completing this task, and we use L-BFGS [33] in this paper.

In the actual process of parameter learning, if the number of candidate roads is not limited, every road in the city will be involved in the calculation, and this will lead to low efficiency. By preanalyzing on our real dataset, we find that the distance between a GPS point and its correctly matched road is unlikely more than 400 meters. Therefore, we can ignore roads that are more than 400 meters away from the GPS point so that the algorithm can be executed faster without reducing the accuracy. Maximum likelihood estimates of ω can be obtained after parameter learning; this is denoted by $\widehat{\omega}$. Map matching is the process of seeking the maximum a posteriori estimation of $p(\gamma \mid \theta, \omega)$. In this paper, we use a Viterbi algorithm [15], which can calculate a global optimal solution with low time complexity.

4.4. Map Matching Based on CRF and Route Preference Mining (RPM). The CRF model above can match a GPS trajectory to the corresponding path. However, when dealing with a low-sampling-rate trajectory, the accuracy is not satisfactory. The main reason is that the correlation between neighboring points decreases with increasing sampling intervals. As mentioned earlier, if the sampling rate is too low, the span of two neighborhood observations $o^{(t)}$ and $o^{(t+1)}$ may contain multiple candidate subpaths, whose length and speed limit are close to others, especially in the zone of dense interactions. Such an example was shown in Figure 1. To tackle this issue, some additional information is needed to supply the weakness of the resolving power of existing features. In previous research, Froehlich and Krumm [34] found that 60 percent of the paths of a driver are repeated, and most drivers select routes following their personal preferences. Motivated by this, we use personal route preference to enlarge the discrepancy in transition features. A natural way to quantify route preference is using the conditional probability $p_v(r_n^{(t+1)} \mid r_m^{(t)})$ that vehicle v traverses the nth candidate road at time $t+1$, given the condition that it traverses the mth candidate road at time t. This can be calculated as follows:

$$\begin{aligned} p_v\left(r_n^{(t+1)} \mid r_m^{(t)}\right) &= \frac{p_v\left(r_n^{(t+1)}, r_m^{(t)}\right)}{p_v\left(r_m^{(t)}\right)} \\ &= \frac{\mathrm{Count}_v\left(r_m \longrightarrow r_n\right)}{\sum_{j=1}^{I_{t+1}} \mathrm{Count}_v\left(r_m \longrightarrow r_j\right) + 1}. \end{aligned} \tag{8}$$

Here, $\mathrm{Count}_v(r_m \rightarrow r_n)$ denotes the number of trips of vehicle v, whose paths contain the road segments r_m and r_n following the order $r_m \rightarrow r_n$. The symbol I_t denotes the number of candidate roads at time t. The conditional probability $p_v(r_n^{(t+1)} \mid r_m^{(t)})$ can be obtained by counting in the database. However, after some thought, it is easy to find that $p_v(r_n^{(t+1)} \mid r_m^{(t)})$ cannot describe the personal route preference entirely, because if the historical records of a driver or a path are scarce, the value of $p_v(r_n^{(t+1)} \mid r_m^{(t)})$ is incredible. To tackle this issue, we introduce the function $f_v(r_m)$ that represents the driving experience of vehicle v on road r_m. In [12], the growth of driving experience is modeled using a sigmoid curve; this motivated us to give a similar definition:

$$f_v\left(r_m\right) = \frac{1}{1 + e^{-(a x_m^v + b)}}, \tag{9}$$

where x_m^v is the number of times that vehicle v traverses r_m, the expression $a x_m^v + b$ is the linear transformation mapping x_m^v from $[0, +\infty]$ to $[-5, +5]$, and a and b are coefficients. Obviously, the more times vehicle v traverses road r_m, the closer $f_v(r_m)$ is to 1. Based on the analysis above, the route preference $h_v(r_m^{(t)}, r_n^{(t+1)})$ of v vehicle is given as

$$h_v\left(r_m^{(t)}, r_n^{(t+1)}\right) = f_v\left(r_m\right) p_v\left(r_n^{(t+1)} \mid r_m^{(t)}\right). \tag{10}$$

The personal route preference is the reinforcement of the transition features. But the latter cannot be completely replaced by the former. Because although the resolving power of the transition features is weakened with decreases in the sampling rate, they do not completely disappear. In many cases, the transition features are still effective. Therefore, a reasonable way of using route preference is that, when matching low-sampling-rate observations, $h_v(r_m^{(t)}, r_n^{(t+1)})$ is superposed, appropriately weighted, on the transition features of CRF to generate a new transition feature, which is given by

$$\begin{aligned} &\delta'\left(r_m^{(t)}, r_n^{(t+1)}\right) \\ &\quad = \alpha \cdot h_i\left(r_m^{(t)}, r_n^{(t+1)}\right) + (1 - \alpha) \\ &\qquad \cdot \left[\lambda_1 \cdot \delta_1\left(r_m^{(t)}, r_n^{(t+1)}\right) + \lambda_2 \cdot \delta_2\left(r_m^{(t)}, r_n^{(t+1)}\right)\right], \end{aligned} \tag{11}$$

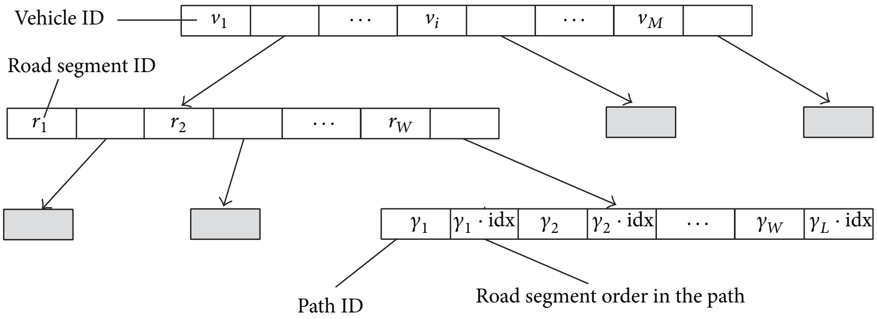


Figure 4: The structure of inverted-index table.

```
Input: a trajectory $o_{1:T}$ of vehicle $v$, CRF model, IDT of vehicles
Output: the matched path (road segment sequence)
(1) Check the average sampling interval of trajectory, if lower than threshold, **goto** (2); **otherwise goto** (9);
(2) **for** $t := 0$ to $T - 1$ **do**
(3)     **for** each candidate road $i$ in candidate road set of GPS observation of time $t$ **do**
(4)         Count the number $C_i$ of paths passing through $i$
(5)         **for** each candidate road $j$ in candidate road set of GPS observation of time $t + 1$ **do**
(6)             Count the number $C_{i \to j}$ of paths passing through $i$ to $j$ in order $i \to j$
(7)             Calculate the route preference of driver $v$ using formula (10)
(8)             Calculate new transmission features using formula (11)
(9) Calculate the maximum posterior probability using Viterbi algorithm.
```

Algorithm 1: Map matching algorithm based on CRF and route preference mining.

where α is weighting factor, which is used to control the balance between the transition features in CRF and route preference, and which is determined by experiment. Specifically, we can set α to different values and observe accuracy changes in the results of applying our algorithm and then choose the value of α that makes the algorithm achieve higher performance.

When calculating the function, there are frequent queries for the number of a certain road or a certain path passed by a vehicle, which lead to the low efficiency. To reduce the response time, we design an inverted-index table (IDT), which is implemented in three layers of nested hash structure. Its structure is showed in Figure 4. The key of the first layer hash is the vehicle ID, and its value is a pointer, which points to the second-layer hash. The key of the second-layer hash is the ID of the road segment visited by the corresponding vehicle, and its value is a pointer pointing to the third-layer hash table. The key of the third-layer hash is the ID of the path that includes the road segments traversed by the corresponding vehicle, and the value of the third-layer hash is the road segment order in the corresponding path. It is used to determine the direction of the path.

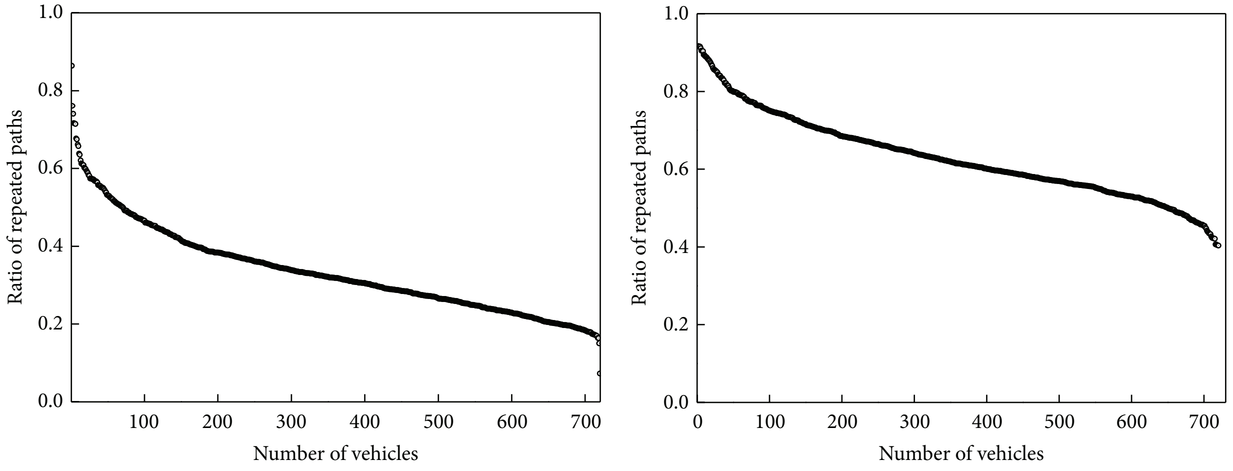
In general, drivers adopt different route strategies at different time. For example, a detour is taken to guarantee the minimum travel time in morning peak hours, and the shortest path or habitual route is selected in normal times. Therefore, in order to calculate route preference accurately, we partition a day into three time slots: morning peak, from 7:30 to 9:30, evening peak, from 17:30 to 19:30, and a normal period of nonpeak hours. Each IDT is built in each time slot. When launching a query, the request is handled in the corresponding IDT according to the sampling time. Algorithm 1 shows our proposed algorithm combining CRF and route preference.

5. Experiments

5.1. Experimental Setting and Dataset Description. For our experiments, we construct a LAN consisting of three computers, which are used to provide GIS service, database, and algorithm execution. We implement our algorithm using C#, and the GIS server is set up with ArcGIS. The digital map mainly includes road network layers with "shape" format. The attributes of the road network include ID, name, level, and length. The network traffic flow information is derived from statistics on historical trajectory data. All information, such as original GPS, traffic flow, the path of each trip, and the IDT, is stored in SQL Server.

We use real trajectory data set generated by 720 Beijing taxis over a period of six months (from March 2012 to August 2012). The sampling interval is approximately 10 seconds. The entire dataset is labeled manually. To validate the regularity of the personal route, we define the repeated path γ_c of path γ_h, which can be computed as

$$\operatorname{repeat}\left(\gamma_c \longrightarrow \gamma_h\right) = \frac{\operatorname{Card}\left(\gamma_c \cap \gamma_h\right)}{\operatorname{Card}\left(\gamma_c\right)}. \tag{12}$$



(a)

(b)

Figure 5: A comparison of the repeated paths ratio in two months (a) and five months (b) of the trajectories dataset.

When repeat($\gamma_c \rightarrow \gamma_h$) is greater than the threshold ζ, the path γ_c can be considered as a repeated path of γ_h; here ζ is set to 0.8. Note that γ_b is not necessarily the repeated path of γ_c. For example, γ_h is a path containing ten road segments, and γ_c is subpath of γ_h, which contains four road segments. The statistics of our dataset are shown in Figure 5. We find that the ratio of repeated paths increases with growth in the amount of data, and the taxi routes show the expected repetitiveness. When trajectories are accumulated over two months, an average 32.4% of paths are repeated. For 68 of these vehicles, the rate is 50%+. Only 47 vehicles have less than 20% repeated paths, since their daily trips are fewer than others. When data is gathered over five months, the average repeated paths rate grows to 63.2%, and 54 vehicles have 80%+ repeated paths, with all others reaching a minimum of 40.4% repeated paths.

We also validate that the number of repeated paths grows with more days of observation. Figure 6(a) presents the growth trend of one driver's repeated path ratio over a period of five months. At first, the ratio increases relatively slowly, and it is less than 20% when accumulating to the 40th day, which indicates that more obscure paths are generated in this period. Then the growth rate increases significantly as the number of days continues to grow; when accumulating to 90 days, the average repeated path ratio is 60%+. Thereafter, the growth rate seems to slow again; ultimately, the average of this ratio holds at approximately 70%. Furthermore, the regularity of route selection is verified. Figure 6(b) presents 387 paths derived from 6820 trips of a driver over five months. We find that the major path bears 235 trips, which is 3.4% of the total number; moreover, 1432 trips, 21% of the total, are distributed over the top 10 frequent paths. This inhomogeneity of path distribution confirms the existence of personal route preferences.

5.2. Experimental Result. The effectiveness of our proposed algorithm is evaluated using two accuracy metrics: accuracy by road segments (A_s) and accuracy by paths (A_r). These are defined as

$$A_s = \frac{\#\text{correctly matched road segments}}{\#\text{all road segments of the trajectories}}$$
$$A_r = \frac{\#\text{correctly matched paths}}{\#\text{all paths of the trajectories}}. \tag{13}$$

First, we estimate the effects of our CRF model (denoted as CRF^1) by comparing with other algorithms, such as incremental, HMM, and CRF [29] (denoted as CRF^2). We select any five months' trajectories as a training set and process the remaining one-month trajectories to generate a plurality of groups at different time intervals (by 30-second increments) for testing. Figure 7 shows the results of each algorithm.

As demonstrated in Figure 7, under the same conditions, A_r is much lower than A_s, due to the strict restriction of A_r. When the sampling interval is less than 30 seconds, all algorithms can achieve high accuracy. And as the sampling interval increases, the accuracy of all algorithms shows varying degrees of decline. All of these algorithms use the context information of the sampling points, and the effects of context information are weakened gradually with decreases in sampling frequency. Specifically, the accuracy of the incremental approach using the local geometric features drops most dramatically. Both HMM and CRF^2, which only consider the spatial context, exhibit similar performance to each other. And the accuracy of CRF^2 is slightly higher, since the CRF^2 model obtains the optimized weights of each feature by parameter learning and makes better use of the spatial context. As temporal correlation is considered, when the sampling interval is not too large, CRF^1 significantly outperforms CRF^2. We also find that, as the sampling interval continues to increase, the accuracy curve of CRF^1 drops sharply and gets close to CRF^2, which indicates that temporal correlation has

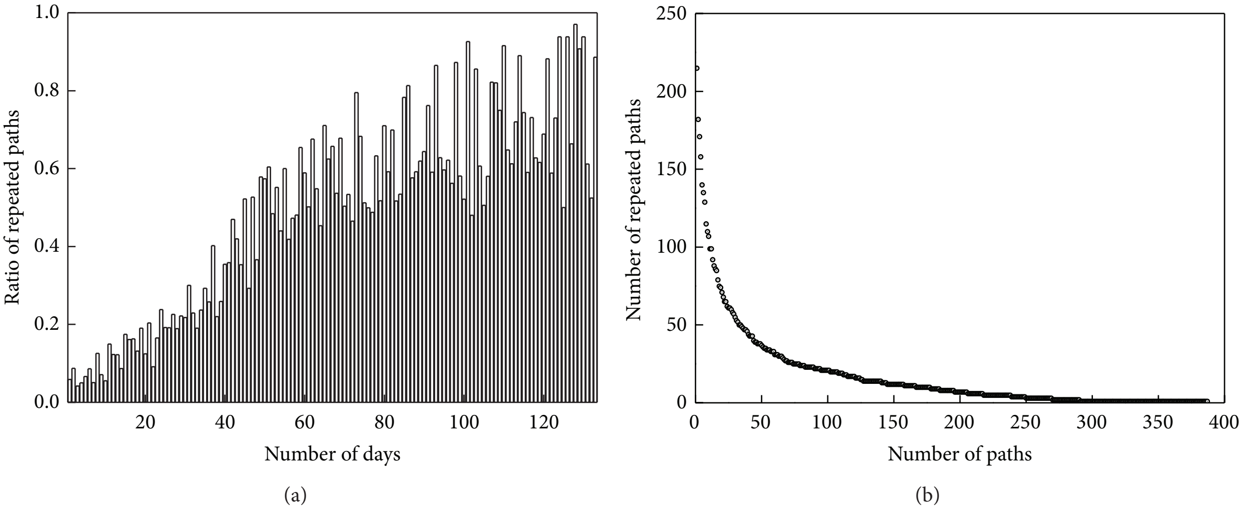


Figure 6: The relation between repeated paths ratio and days (a) and distribution of number of repeated paths (b) for a taxi.

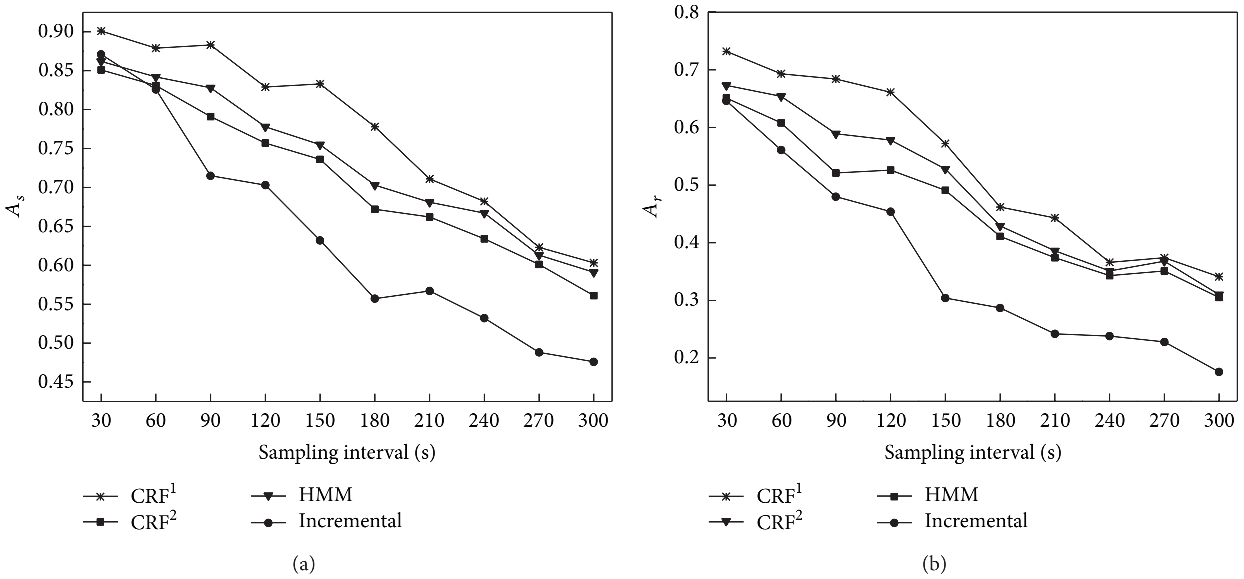
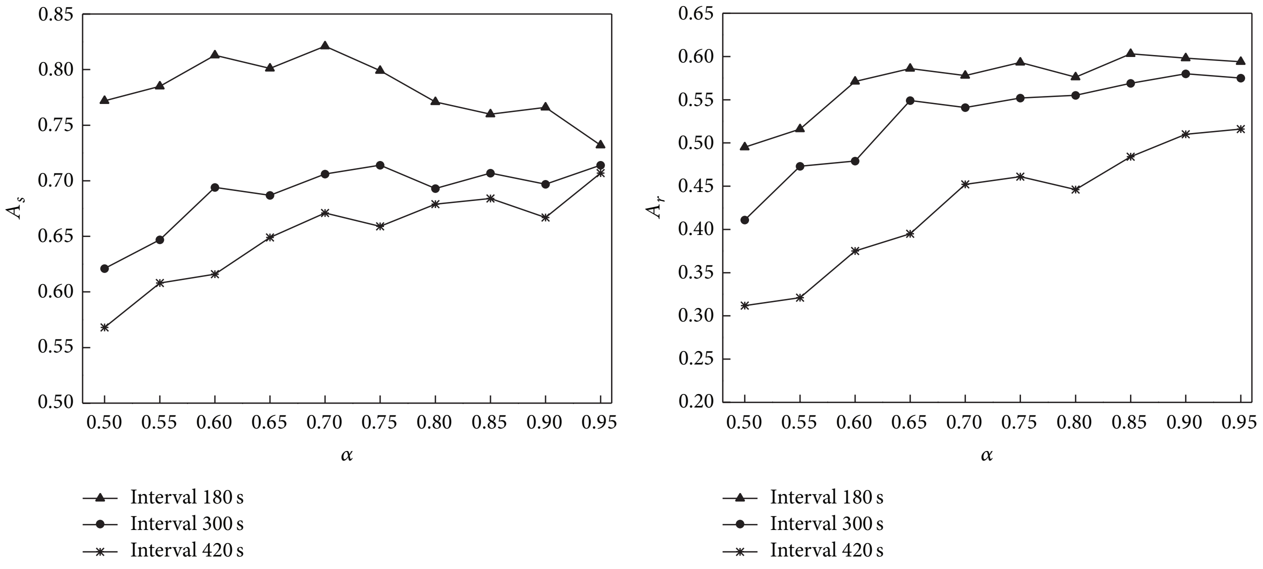


Figure 7: Comparison of HMM, incremental algorithm, CRF[1], and CRF[2].

a weaker effect when neighboring observations are far away from each other.

The parameter α in our proposed framework is used to balance the ratio of temporal-spatial constraints and route preference. We estimate α by analyzing the change in accuracy resulting from setting α to different values. We use any five months' matched trajectories to calculate route preference, and set up three groups of test sets according to different sampling intervals: 180 seconds, 300 seconds, and 420 seconds. Figure 8 shows the evaluation results. In the test set with a 180-second interval, with the value of α rising, A_s grows gradually first and reaches a maximum of 0.821 when α is 0.7; then A_s declines. The reason is that when the proportion of route preference is too high, its effect covers the effective temporal-spatial features. Although some GPS observations are matched to the correct road segments, more observations are mismatched to road segments, which the corresponding driver traverses frequently. Meanwhile, A_r exhibits a tendency toward stabilization after a time of rising. In test sets on 300-second and 420-second intervals, A_s and A_r exhibit a growth trend to different degrees, which implies that, under lower sampling frequencies, the temporal and spatial contexts have less effect, and route preference ought to dominate map matching. Considering that trajectories of higher sampling frequency are more valuable, it is necessary to ensure that the matching accuracy of these trajectories is as high as possible. According to evaluation results on different sampling intervals, the value of α is set to 0.7.

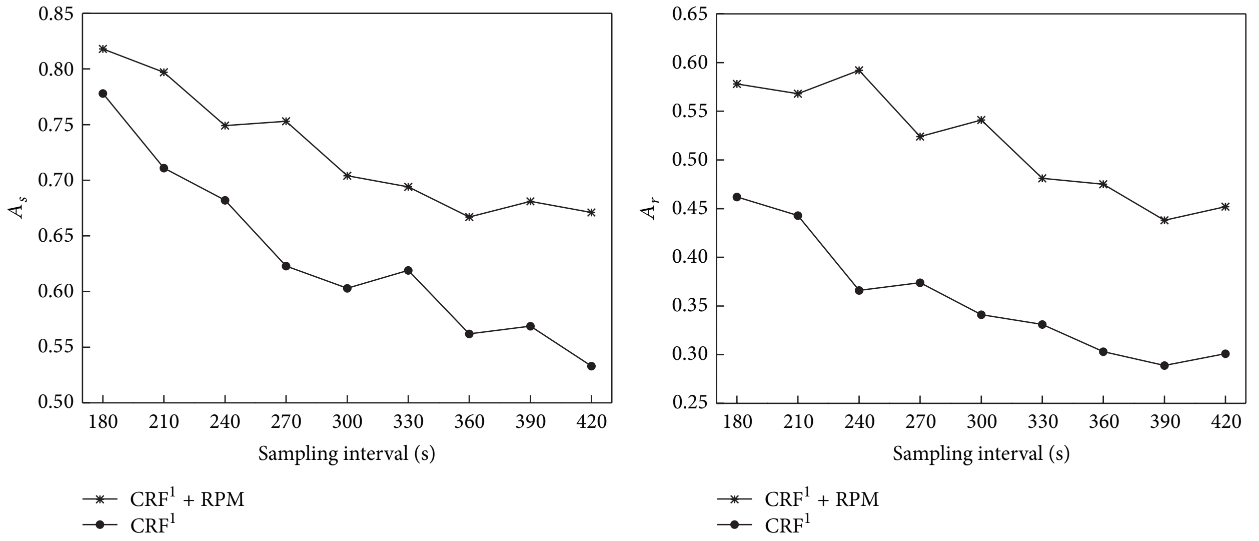
When the sampling interval exceeds 180 seconds, A_s and A_r drop lower than 80% and 50%, respectively. This



(a)

(b)

Figure 8: Evaluation of α under different sampling intervals.



(a)

(b)

Figure 9: The results of CRF^1 and CRF^1 + RPM.

implies that CRF^1, which depends on temporal-spatial context, can hardly maintain a satisfying effectiveness of map matching. So we integrate route preference into the CRF-based algorithm to provide replenishment of context. Next, we compare our algorithm combining CRF^1 and RPM (CRF^1 + RPM) with CRF^1. As presented in Figure 9, with increasing sampling interval, A_s and A_r of these two algorithms drop at different rates. However, comparing with CRF^1, A_s of CRF^1 + RPM drops slowly and tends to become stable. This means that the improvement of A_s, which is attributed to route preference mining, is more significant with an increasing sampling interval, and the growth rate of A_r is more stable and exceeds 12%. This part of the experiment indicates that CRF^1 + RPM outperforms CRF^1 for all sampling rates significantly. Figure 10 presents a comparison of matching results of CRF^1 and CRF^1 + RPM. As shown in Figure 10, a trajectory with a 300-second sampling interval, consisting of four GPS observations, is matched to a wrong path using CRF^1, while we get the correct path using CRF^1 + RPM.

Finally, we validate the influence of the accumulation of data on effectiveness. Similar to the above, our algorithm is evaluated at different sampling intervals: 180 seconds, 300 seconds, and 420 seconds, and the amount of trajectory used for route preference analysis is increased successively, with

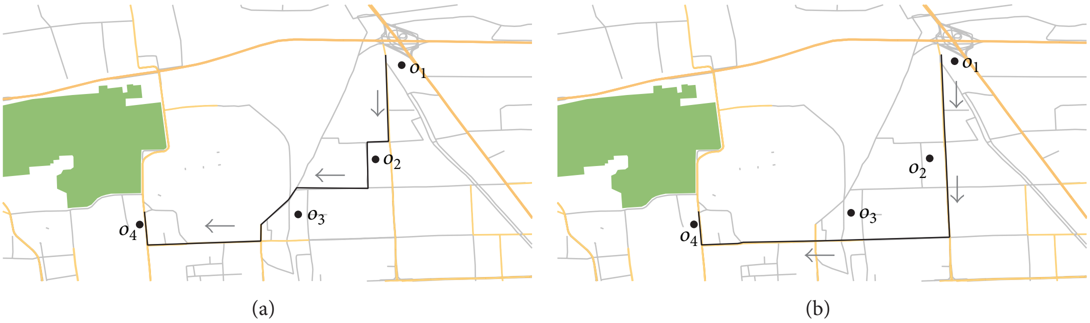


FIGURE 10: Matching results of CRF[1] before (a) and after (b) integrating route preference mining.

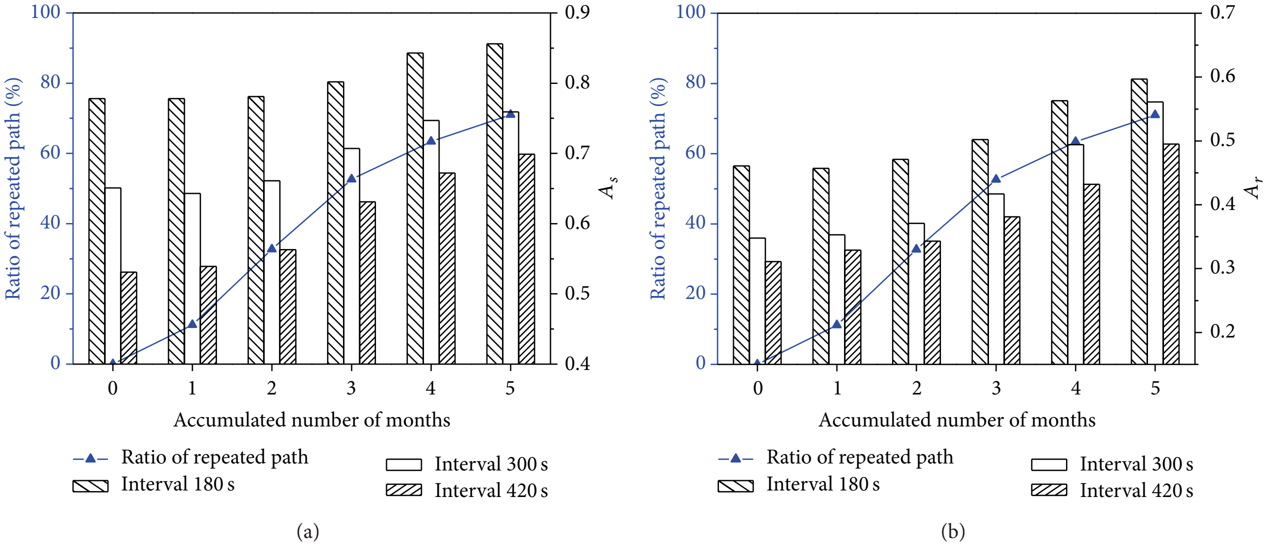


FIGURE 11: The evaluation of historical accumulation impact on accuracy.

increments of one month. Figure 11 presents the evaluation results. In the first two months, the proportion of repeated paths is so low that route preference mining cannot make a great contribution to improve performance, since this stage mainly reflects a period of generating rare paths. When three months' worth of data is accumulated, accuracy begins to grow sharply, following the rise of the repeated path ratio. And when the amount of data reaches five months, A_s and A_r are improved by nearly 10%+ and 15%+, respectively. In addition, we also find that RPM is more effective for the low-sampling-rate trajectories.

5.3. Discussion. Our proposed map matching algorithm is based on linear chain CRF, which can be regarded as the undirected graphical model version of HMM. However, compared with the HMM, CRF provides better support for modeling overlapping, nonindependent features of the output. Taking advantage of this, we introduce the temporal context of neighboring observations, which can boost the resolving power of the transition feature and integrate it with the spatial context. Experiments show that this combination achieves very good performance. In some regions of dense and diverse roads, many candidate paths of GPS observations have similar spatial context information, but their temporal features are quite different, as with, say, expressways and bypasses; thus we see that temporal correlation can heighten the divergence among the solutions to some extent.

When the sampling interval is too large (e.g., exceeding 240 seconds), the effect of temporal context diminishes sharply; so we remove the temporal feature from transition features in order to make the algorithm execute faster; meanwhile, we introduce route preference. These correct matched trajectories, resulting from consideration of route preference, are derived from the trips that mainly happen in the morning or evening peak hours. During these peak hours, probably, the shortest path is not the fastest path, due to traffic congestion, and drivers may choose a roundabout but habitual route. This may weaken the effect of the spatial feature and generate more mismatches under a low-sampling-rate. The addition of route preference maintains the accuracy of the algorithm at a relatively stable and high level. Further, it tends to correctly match or mismatch an entire trajectory. Once mismatching of an observation occurs, a considerable proportion of observations would be matched to incorrect road segments.

Note that the trajectories we used are derived from taxi trips, which are irregular, due to randomness of picking up passengers. If this technology was applied to private cars, it is likely that higher performance would be achieved.

6. Conclusions and Future Work

This paper proposes a CRF-based map matching algorithm, which has the advantage of integrating context information into features flexibly. The spatial and temporal correlations between neighborhood sampling points are chosen as features in order to improve distinguishability of the trajectory. To further improve the algorithm's performance, in the case of a low-sampling-rate, we discover and utilize the personal route preference information to supply the lack of effective features. Experimental results illustrate that this algorithm can accurately find the actual path on multiple sampling frequency datasets. Compared with other map matching methods, the performance is significantly improved. In the future, we will further study the impact of unexpected social events on the effectiveness of our method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research reported in this paper was conducted as part of the project "The research and application of the urban air environment regulation and control technology based on the Internet of Things," which is under the State High-Tech Development Plan (the 863 program), and funded by National Science Foundation of China (Project no. 2012AA063303) and National Science Foundation of China (Project no. 51408237) "Research on Modeling and Simulation of Bicycle Microscopic Behaviors at Nonsignalized Intersections Based on Information Fusion." In addition, the authors thank Dr. Ling Huang for her help in English improvement of their paper.

References

[1] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger?" in *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*, pp. 109–118, ACM, September 2011.

[2] W. He, D. Li, T. Zhang, L. An, M. Guo, and G. Chen, "Mining regular routes from GPS data for ridesharing recommendations," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing (UrbComp '12)*, pp. 79–86, ACM, 2012.

[3] S. Ma, Y. Zheng, and O. Wolfson, "T-share: a large-scale dynamic taxi ridesharing service," in *Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE '13)*, pp. 410–421, IEEE, Brisbane, Australia, April 2013.

[4] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 186–194, ACM, August 2012.

[5] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, pp. 89–98, ACM, September 2011.

[6] J. Zhang, "Smarter outlier detection and deeper understanding of large-scale taxi trip records: a case study of NYC," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing (UrbComp '12)*, pp. 157–162, ACM, Beijing, China, August 2012.

[7] C. Chen, D. Zhang, P. S. Castro et al., "iBOAT: isolation-based online anomalous trajectory detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 806–818, 2013.

[8] S. Chawla, Y. Zheng, and J. Hu, "Inferring the root cause in road traffic anomalies," in *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM '12)*, pp. 141–150, December 2012.

[9] D. Tong, W.-H. Lin, and A. Stein, "Integrating the directional effect of traffic into geostatistical approaches for travel time estimation," *International Journal of Intelligent Transportation Systems Research*, vol. 11, no. 3, pp. 101–112, 2013.

[10] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD International Conference*, pp. 25–34, ACM, August 2014.

[11] E. Jenelius and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," *Transportation Research Part B. Methodological*, vol. 53, pp. 64–81, 2013.

[12] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 316–324, ACM, August 2011.

[13] J. Yuan, Y. Zheng, C. Zhang et al., "T-drive: driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 99–108, ACM, November 2010.

[14] N. Nguyen and Y. Guo, "Comparisons of sequence labeling algorithms and extensions," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 681–688, ACM, June 2007.

[15] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 282–289, 2001.

[16] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09)*, pp. 336–343, ACM, November 2009.

[17] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pp. 591–598, 2000.

[18] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 282–289, 2001.

[19] D. R. Brillinger, “Modeling spatial trajectories,” in *Handbook of Spatial Statistics*, A. E. Gelfand, P. Diggle, P. Guttorp, and M. Fuentes, Eds., pp. 463–475, CRC Press, Boca Raton, Fla, USA, 2010.

[20] J. Gong, J. Tang, and A. C. M. Fong, “ACTPred: activity prediction in mobile social networks,” *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 265–274, 2014.

[21] X. Su, H. Tong, and P. Ji, “Activity recognition with smartphone sensors,” *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 235–249, 2014.

[22] J. S. Greenfeld, “Matching GPS observations to locations on a digital map,” in *Proceedings of the Transportation Research Board 81st Annual Meeting*, 2002.

[23] S. S. Chawathe, “Segment-based map matching,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’07)*, pp. 1190–1197, IEEE, June 2007.

[24] C. Wenk, R. Salas, and D. Pfoser, “Addressing the need for map-matching speed: localizing global curve-matching algorithms,” in *Proceedings of the IEEE 18th International Conference on Scientific and Statistical Database Management*, pp. 379–388, 2006.

[25] H. Alt, A. Efrat, G. Rote, and C. Wenk, “Matching planar maps,” *Journal of Algorithms. Cognition, Informatics and Logic*, vol. 49, no. 2, pp. 262–283, 2003.

[26] S. Brakatsoulas, D. Pfoser, and R. Salas, “On map-matching vehicle tracking data,” in *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB ’05)*, pp. 853–864, VLDB Endowment, 2005.

[27] H. Yin and O. Wolfson, “A weight-based map matching method in moving objects databases,” in *Proceedings of the16th International Conference on Scientific and Statistical Databse Management (SSDBM ’04)*, pp. 437–438, IEEE, June 2004.

[28] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, “Map-matching for low-sampling-rate GPS trajectories,” in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 352–361, ACM, November 2009.

[29] L. Liao, D. Fox, and H. Kautz, “Extracting places and activities from GPS traces using hierarchical conditional random fields,” *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 119–134, 2007.

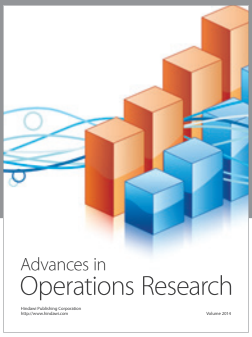
[30] F. Sha and F. Pereira, “Shallow parsing with conditional random fields,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. 134–141, Association for Computational Linguistics, May 2003.

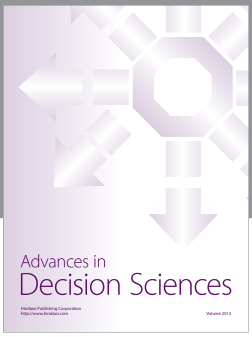
[31] Y. Cui and S. S. Ge, “Autonomous vehicle positioning with GPS in urban canyon environments,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 15–25, 2003.

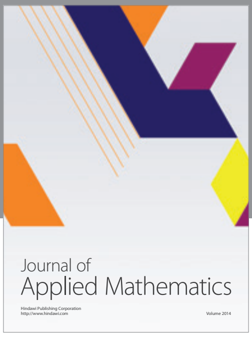
[32] T. Hunter, P. Abbeel, and A. Bayen, “The path inference filter: model-based low-latency map matching of probe vehicle data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 507–529, 2014.

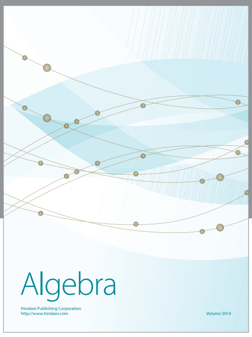
[33] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 1999.

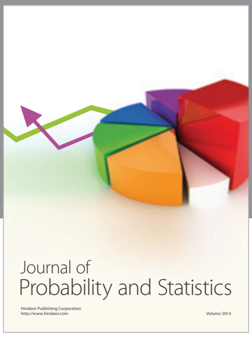
[34] J. Froehlich and J. Krumm, “Route prediction from trip observations,” SAE Technical Paper, SAE International, 2008.


Advances in
Operations Research

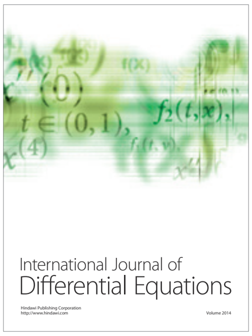

Advances in
Decision Sciences


Journal of
Applied Mathematics

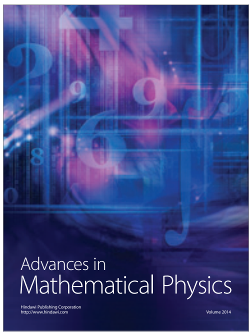

Algebra

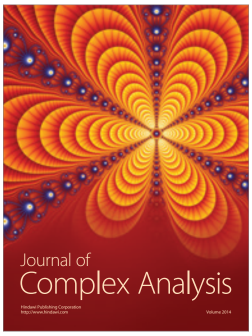

Journal of
Probability and Statistics

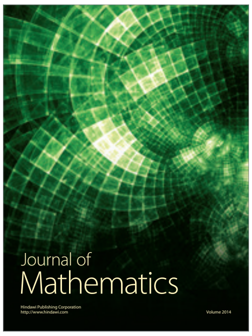
The Scientific
World Journal

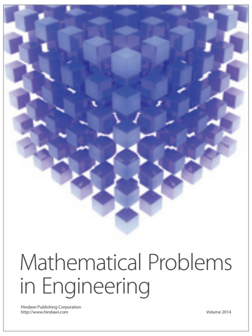

International Journal of
Differential Equations

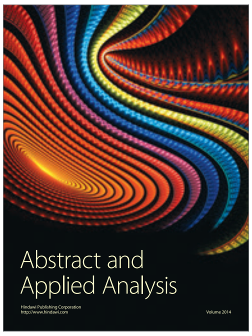
International Journal of
Combinatorics

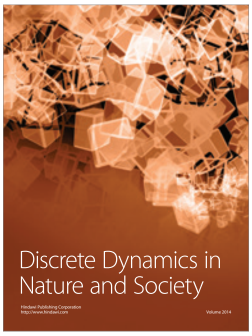

Advances in
Mathematical Physics

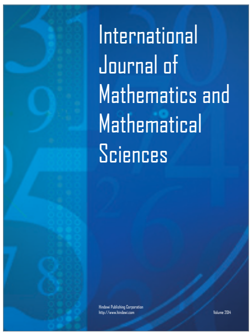

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

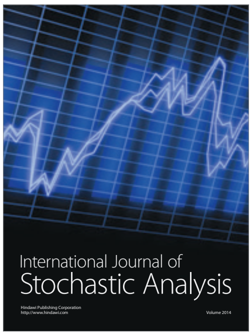

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization