Online Map-Matching of Noisy and Sparse Location Data with Hidden Markov and Route Choice Models

George R. Jagadeesh and Thambipillai Srikanthan

Abstract— **With the growing use of crowdsourced location data from smartphones for transportation applications, the task of map-matching raw location sequence data to travel paths in the road network becomes more important. High-frequency sampling of smartphone locations using accurate but power-hungry positioning technologies is not practically feasible as it consumes an undue amount of the smartphone's bandwidth and battery power. Hence, there exists a need to develop robust algorithms for map-matching inaccurate and sparse location data in an accurate and timely manner. This paper addresses the above need by presenting a novel map-matching solution that combines the widely-used approach based on a Hidden Markov Model (HMM) with the concept of drivers' route choice. Our algorithm uses a HMM tailored for noisy and sparse data to generate partial map-matched paths in an online manner. We use a route choice model, estimated from real drive data, to reassess each HMM-generated partial path along with a set of feasible alternative paths. We evaluated the proposed algorithm with real-world as well as synthetic location data under varying levels of measurement noise and temporal sparsity. The results show that the map-matching accuracy of our algorithm is significantly higher than that of the state of the art, especially at high levels of noise.**

Index Terms—**Map-matching, Location, Hidden Markov model, Route choice model.**

I. Introduction

LOCATION sequence data sourced from smartphones of travelers can be of great use for many Intelligent Transportation Systems (ITS) applications. For instance, such data can be used to estimate link travel times and to study travel behavior. This entails an important first step known as map-matching, which matches a sequence of raw location measurements to a sequence of road segments that make up the path taken by a vehicle. Map-matching algorithms combine location sequence data generated by positioning technologies with the road network data to provide locations on the road network. They mostly rely on densely-sampled Global Positioning System (GPS) data as input and use a variety of approaches including geometric analysis, topological analysis, and probabilistic methods [1].

The authors are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: {asgeorge, astsrikan} @ntu.edu.sg).

When the location data are of sufficiently good quality in terms of measurement accuracy and sampling frequency (e.g. GPS data sampled every few seconds), existing map-matching algorithms are capable of achieving excellent results. However, in practice, much of the location sequence data collected from smartphones are spatially inaccurate and temporally sparse. The spatial inaccuracy arises mainly because many applications prefer to use energy-efficient, but inaccurate, alternatives to GPS such as Wi-Fi positioning and cellular network positioning. These positioning technologies vary significantly in terms of their energy consumption and accuracy. For instance, while cellular network positioning is at least 10 times more energy-efficient compared to GPS [2], the typical location measurement error associated with the former is nearly 2 orders of magnitude higher than the latter [3]. The temporal sparsity of the location data results from the need to limit the smartphone's bandwidth usage by sampling the locations at sparse intervals. Noisy and sparse location data are adequate for many non-ITS applications. However, for such data to be usable for ITS applications, the state-of-the art map-matching algorithms need to be greatly improved.

A pioneering work on map-matching noisy and sparse location data was published by Newson and Krumm [4] in 2009. They used a Hidden Markov Model (HMM) to find the most likely path corresponding to a timestamped sequence of coordinates. Another HMM-based map-matching algorithm was proposed by Thiagarajan et al. [5], who tested it on Wi-Fi positioning data as well as on GPS data degraded with Gaussian noise. While only a few map-matching algorithms such the abovementioned ones have been evaluated with high levels of noise, most other works have dealt with relatively accurate, but sparsely sampled, GPS data. Goh et al. [6] proposed a HMM-based algorithm for online map-matching, in which partial sequences of road segments are output on the fly without waiting for all the data points to be received. Apart from HMM, another probabilistic graphical model, Conditional Random Fields (CRF), has also been used for map-matching [7][8].

A less common approach towards map-matching sparsely-sampled location data is to utilize information about drivers' route choice to select the best path from a set of admissible candidate paths. The simplest way of doing this is to use a deterministic route choice criterion. For instance, Rahmani and Koutsopoulos [9] generated a number of candidate paths in a search graph and selected the path with the minimum length from among a subset of paths whose travel times are within a threshold. This solution was later improved by Sarlas [10], who replaced the deterministic route choice criteria with a probabilistic logit route choice model [11] learned from GPS log data. Miwa et al. [12] applied a logit route choice model in a piecewise manner for short portions of the location sequence to select the best among several candidate paths.

An unconventional map-matching approach proposed by Bierlaire et al. [13] has some relevance to the work that we present in this paper. Instead of generating an unique best fitting path, their solution produces a set of potential paths along with a likelihood that the location measurements are recorded from each path. As an application example, they showed that the potential paths and their associated likelihoods can be used to estimate a route choice model.

Results reported in the literature on map-matching noisy location data [4] [5] indicate that the existing algorithms are grossly inadequate for map-matching cellular network positioning data, whose average errors are in the order of several hundred meters. However, cellular network positioning has some important advantages over other more accurate alternatives, such as Wi-Fi positioning. Wi-Fi positioning is usually unavailable in non-built-up areas and consumes several times as much battery power as cellular network positioning [2]. Many smartphone users keep GPS and Wi-Fi disabled to conserve battery life, leaving cellular network positioning as the only means of determining their location. There have been some works aimed at map-matching location data from cellular network positioning. Thiagarajan et al. [2] presented a HMM-based system named CTrack, which map-matches a stream of densely-sampled cellular base station fingerprints augmented with data from smartphone-based sensors such as accelerometer and compass. The system achieved a median accuracy of about 75%, which was found to be lower than the accuracy needed for ITS applications such as traffic delay estimation. Schulze et al. [14] used an approach based on deterministic route choice to map-match locations of the connected cellular base stations, unrefined by any fingerprinting process. They found that, on average, only about 55% of each track was matched correctly.

From the above, it is clear that a need exists for developing new methods for more accurately matching highly noisy location data such as cellular network positioning data to paths in the road network. This paper aims to address the above need by proposing an innovative extension to the widely-used HMM-based approach, which remains the state of the art for map-matching noisy and sparse location data. When supplied with noisy location data, HMM-based algorithms are prone to generating incorrect paths that are implausible from the perspective of drivers' route choice. Intuitively, it appears natural to hypothesize that the map-matching accuracy could be improved by complementing HMM-based algorithms with a model of drivers' route choice. We explore and validate this idea in detail in this paper. We propose a framework for online map-matching of highly-noisy and temporally-sparse location data for real-time ITS applications. In the proposed framework, we use a route choice model estimated from real-world data to reassess the partial paths generated by a HMM-based online map-matching method. We thoroughly evaluate the above solution with real-world as well as synthetic datasets under varying levels of measurement noise and temporal sparsity and show that it yields consistently better results compared to the state of the art.

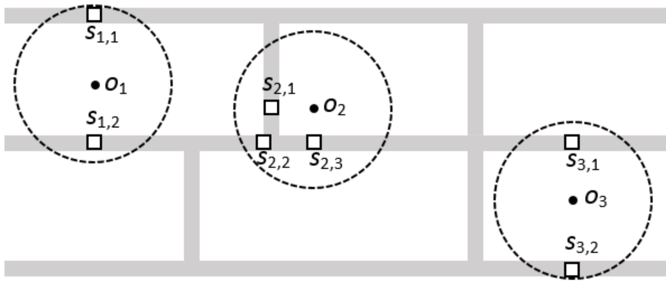
The remainder of the paper is organized as follows. In Section II, we build on our previous work [15] and describe a HMM-based method for online map-matching of streaming location data that have low temporal resolution and accuracy. In Section III, we present a methodology for modelling short-term route choice behavior from data on actual routes taken by real drivers. The overall map-matching algorithm, in which a multinomial logit route choice model is coupled with the HMM-based framework in order to improve the noise robustness, is presented in Section IV. In Section V, we describe the setup used for evaluating the proposed solution and discuss the results. Section VI summarizes our findings and concludes the paper.

II. HMM-Based Online Map-Matching

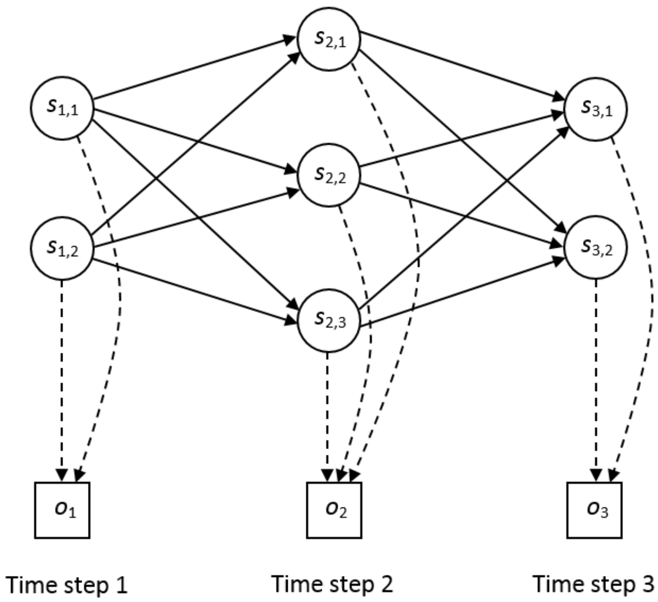
A *location observation* (or simply *observation*) o_t consists of the measured latitude, longitude and timestamp corresponding to a mobile user's location at time step t. A *road network* is a directed graph $G = (V, E)$, where V is a set of nodes corresponding to intersections or endpoints of the road segments and E is a set of edges representing road segments. Each road segment has a number of attributes including road class, length and free-flow travel time. A *path* p between nodes u and v is a sequence of connected road segments (edges) $e_1, \dots, e_n$ such that u is the start node of e_1 and v is the end node of e_n. In the work presented in this paper, a path does not necessarily start and end at nodes. It could start or end at any point that lies along the centerline of any road segment. Given a sequence of N observations $o_{1:N} = \{o_1, \dots, o_N\}$ and a road network G, the map-matching problem is to find the path p in G corresponding to $o_{1:N}$.

The location observations are subject to measurement noise and therefore the true on-road locations corresponding to them are unknown. These true on-road locations are considered as the hidden states in the HMM used for map-matching. In theory, the state space of the HMM would consist of all possible on-road locations in the road network. However, in practice, the state space for a given location observation is limited to the road segments that lie within a fixed range around the measured location. (In our implementation, this range equals 4 times the standard deviation of the location measurement noise.) Furthermore, for each such road segment, only the point along its centerline that is closest to the measured location is considered as a state. In this paper, we denote the k^{th} state at time step t as $s_{t,k}$. The hidden true state at time step t is denoted as s_t^*.

Two properties of HMMs need to be noted: First, the observation o_t at time step t depends only on the hidden state



(a) An example showing raw location measurements and the states corresponding to them.



(b) HMM representation of the above example with arrows indicating the conditional dependencies.

Fig. 1. Illustration of the HMM framework for map-matching.

s_t^* at that time. The second property, known as the Markov property, states that the hidden state s_t^* at time step t depends only on the hidden state s_{t-1}^* at time step $t-1$ and is not influenced by the history of hidden states before that.

The above two dependencies are respectively associated with two conditional probability distributions: For an observation o_t, each state $s_{t,k}$ is assigned an emission probability $P(o_t|s_{t,k})$, which is the conditional probability of the observation o_t being generated if $s_{t,k}$ is the true state. The probability of the vehicle moving from a state $s_{t-1,j}$ at time step $t-1$ to another state $s_{t,k}$ at time step t is given by a transition probability $P(s_{t,k}|s_{t-1,j})$.

The above concepts are illustrated in Figure 1 using a simple example, where the points o_1, o_2 and o_3 denote a sequence of location observations. The dashed circles in Figure 1(a) represent the error range of each location observation. The closest points on each of the road segments that lie either partially or fully within the error range are considered as states. Figure 1(b) illustrates a HMM corresponding to the given example. The arrowed lines indicate the conditional dependencies in the HMM. The dashed arrowed lines indicate the conditional dependency of the observations on the states and have an associated emission probability. The solid arrowed lines denote the transitions between states and are associated with a transition probability.

In a HMM, there could be multiple sequences of states that are consistent with a given sequence of observations. The most likely state sequence in a HMM can be efficiently calculated using the Viterbi algorithm [16]. Details of the emission and transition probabilities used in our map-matching algorithm and the determination of the most likely state sequence are presented below.

A. *Emission Probability*

For each state representing a road segment within the error range of a location measurement, the emission probability depends on the distance between itself and the observed location. It is intuitive to think that states that are closer to the observed location would have a higher emission probability compared to those that are farther from it. In the case of the true state, the distance between itself and the observed location is the location measurement error, which is generally assumed to follow a Gaussian distribution with zero mean. For a given state $s_{t,k}$ and a location observation o_t, the emission probability is given as

$$P(o_t|s_{t,k}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{g(o_t,s_{t,k})^2}{2\sigma^2}} \quad (1)$$

where σ is the empirically-estimated standard deviation of the measurement error and $g(o_t, s_{t,k})$ is the great-circle distance, which is the shortest distance along the surface of the earth, between o_t and $s_{t,k}$. Points at low latitudes that are separated by short distances can be treated as being on a two-dimensional plane and in such cases, the Euclidean distance can be used as an approximation of the great-circle distance. We use this approximation as it reduces the amount of computation required.

It is worth noting that, in practice, the location measurement error may not strictly conform to the above model, especially in dense urban networks. Irrespective of the positioning technology used, the error is known to exhibit non-Gaussian characteristics and geographical variations. However, the model based on Gaussian distribution, while simple, has been shown to be effective in several previous works on map matching [2][4][6].

B. *Transition Probability*

The transition probability between a state $s_{t-1,j}$ at time step $t-1$ and another state $s_{t,k}$ at time step t depends on the features of the optimal path between them. In our work, we consider the path with the minimum free-flow travel time, found using the well-known Dijkstra's algorithm [17], as the optimal path.

For computing the transition probability, Newson and Krumm [4] relied on the idea that transitions through circuitous paths are less likely compared to direct paths. They used the difference between the driving distance along a path and the great-circle distance between the current and previous location observations as a measure of the circuitousness of the path. This measure of circuitousness has a couple of drawbacks. First, as pointed out in [18], the transition probability's dependence on the current and previous observations violates the properties of an ideal HMM. In the case of highly noisy location measurements, the great-circle distance between the observed locations may introduce significant errors in estimating the circuitousness of a path. The second drawback relates to the fact that the transition probabilities computed based on the above measure of circuitousness vary greatly for equally plausible transition paths depending on the sampling interval [19].

Taking the above factors into consideration, we propose a new measure of circuitousness for the optimal path between states $s_{t-1,j}$ and $s_{t,k}$ as

$$y(s_{t-1,j}, s_{t,k}) = \frac{d(s_{t-1,j}, s_{t,k}) - g(s_{t-1,j}, s_{t,k})}{\Delta T} \tag{2}$$

where the functions d and g give the driving distance and the great-circle distance in meters, respectively, between the states and ΔT is the time interval between time steps $t-1$ and t in seconds. In our implementation, the driving distance between two points on the road network is the length of the minimum-travel-time path between them found by Dijkstra's algorithm.

We also consider a measure of temporal implausibility in order to assign low transition probabilities to paths that cannot be traversed within the time interval ΔT unless the vehicle travels at an unreasonably high speed. We define the measure of temporal implausibility for states $s_{t-1,j}$ and $s_{t,k}$ as

$$z(s_{t-1,j}, s_{t,k}) = \frac{max\left(\left(f(s_{t-1,j}, s_{t,k}) - \Delta T\right), 0\right)}{\Delta T} \tag{3}$$

where $f(s_{t-1,j}, s_{t,k})$ is the free-flow travel time, in seconds, of the optimal path between states $s_{t-1,j}$ and $s_{t,k}$.

Based on analysis of true transition paths from real drive data, the measure of circuitousness defined in (2) and the measure of temporal implausibility defined in (3) are assumed to follow exponential distributions. We define the transition probability of moving from state $s_{t-1,j}$ to state $s_{t,k}$ as

$$P(s_{t,k}|s_{t-1,j}) = \lambda_y e^{-\lambda_y y(s_{t-1,j}, s_{t,k})}\, \lambda_z e^{-\lambda_z z(s_{t-1,j}, s_{t,k})} \tag{4}$$

where λ_y and λ_z are parameters of the exponential distributions to be empirically determined.

C. Online Viterbi Inference

The Viterbi algorithm computes the most likely sequence in the HMM using the following recurrence relations.

$$V_{1,k} = P(o_1|s_{1,k}) \tag{5}$$

$$V_{t,k} = P(o_t|s_{t,k})\, max_j \left(V_{t-1,j}\, P(s_{t,k}|s_{t-1,j})\right) \tag{6}$$

In the above equations, $V_{t,k}$ denotes the joint probability of the most likely state sequence ending at state $s_{t,k}$ based on the observations $o_1,\ldots,o_t$. The index j in (6) that maximizes $V_{t,k}$ is stored as the back pointer for state $s_{t,k}$. It points to the predecessor state $s_{t-1,j}$ of the state $s_{t,k}$ in the most likely sequence ending at the latter.

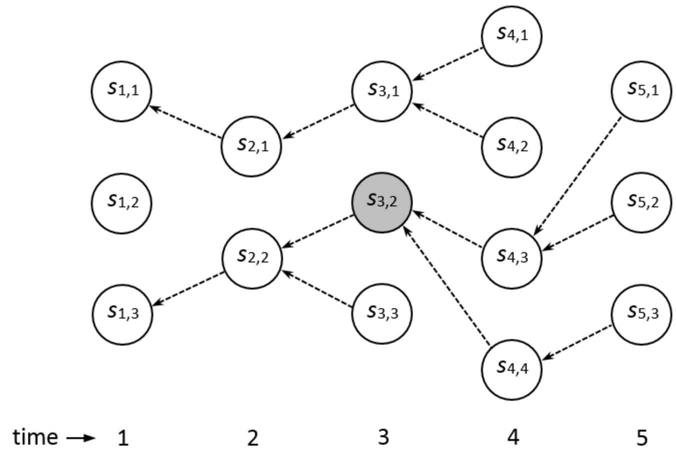


Fig. 2. An example of the convergence condition in the Viterbi algorithm.

When map-matching is performed in offline mode, the Viterbi algorithm outputs the most likely state sequence from time step 1 to time step N after all the N observations are received and processed. In such a case, the state $s_{N,x}$ at time step N that has the highest joint probability is selected as the final state in the most likely sequence. From that state, the complete sequence is traced backwards using the back pointers. The path p corresponding to the given observation sequence $o_{1:N}$ is obtained by concatenating the optimal paths between successive states in the most likely state sequence.

For many real-time applications that receive live streams of input data, the Viterbi algorithm needs to be performed in an online manner such that portions of the most likely state sequence are generated from time to time without waiting for all the observations to be received. In some cases, partial, optimal state sequences can be generated using the concept of transitive closure, which defines the reachability relations in a directed graph [20]. This method, which has been used for online map-matching in [6], checks for a convergence condition where there remains only one state at a time step that is reachable through a chain of back pointers from every state of a subsequent time step. This idea is illustrated in Figure 2, where tracing backward from any of the states at time step 5 leads to the shaded state $s_{3,2}$ at time step 3. In such a case, we say that a convergence condition has occurred with $s_{3,2}$ being the convergence state. In the given example, the partial sequence up to the convergence state (i.e. $s_{1,3}$, $s_{2,2}$, $s_{3,2}$) can be released as output as it cannot be altered by future observations received after time step 5. Thus, in online mode, the Viterbi algorithm can output a part of the most likely state sequence each time a convergence state is found.

When online map-matching is performed as described above, the emission, transition and joint probabilities associated with the states at the current time step are computed when a location observation is received. The determination of the transition probabilities, which involve a number of shortest path computations between states at the previous and current time steps, account for most of the computation time. The number of shortest path computations can be large if the number of states per time step, which depends on the level of location measurement noise, is high [19]. A heuristic

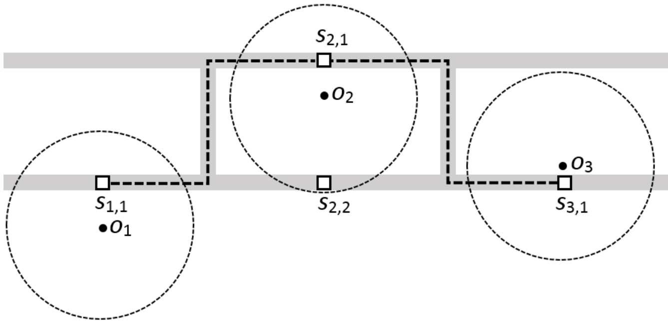


Fig. 3. The HMM-based approach may produce unreasonable paths for highly noisy location data.

technique that limits the number of states per time step can be used to significantly reduce the computation time with negligible loss of accuracy [21][15]. This heuristic technique is applied during the operation of the Viterbi algorithm by retaining only a fixed number, say k, of states with the highest joint probabilities for further computation. An useful rule of thumb is to set this number k equal to the standard deviation of the location measurement noise in meters.

D. A Limitation of HMM-based Map Matching

A major weakness of HMM-based map-matching stems from the Markov independence assumption, according to which the probability distribution of the next state depends only on the current state and not on the past or future states. This results in the loss of contextual information when calculating the probability distribution of the next state. It has been argued by Srivatsa et al. [22] that mobility in a road network is non-Markovian, especially when a vehicle intents to reach a specific destination, typically through the shortest or fastest path. In other words, the paths taken by vehicles are influenced by the drivers' route choices, which are not well accounted for in the HMM framework. This limitation becomes more apparent when the location observations have a high level of measurement noise. Given a noisy observation sequence, some parts of the most likely path found by the HMM-based approach may not make sense from a route choice perspective. This is illustrated using a simplified example in Figure 3 where the HMM-based approach may find the state sequence $s_{1,1}$, $s_{2,1}$, $s_{3,1}$ as the most likely one, although the alternative sequence $s_{1,1}$, $s_{2,2}$, $s_{3,1}$ appears intuitively more probable from the standpoint of human route choice. As we show later in this paper, this weakness of HMM-based map-matching can be largely overcome through the appropriate use of a route choice model that gives the choice probabilities of each alternative path in a given context.

Unlike HMMs, probabilistic frameworks such as CRFs are not limited by the Markov independence assumption and therefore, in theory, can model higher-order dependencies among more than two states. However, the exact inference of parameters in higher-order CRFs is known to be intractable. Existing CRF-based map matching methods [7][8] consider only first-order dependencies between adjacent states and hence suffer from the same inability as HMMs to utilize contextual information.

III. Short-Term Route Choice Modelling

Given a number of alternative paths between two points on a road network, a route choice model can be used to estimate the probability of each of the alternative paths being chosen by a driver. For this purpose, we use a multinomial logit model with a few simplifying assumptions.

A. Multinomial Logit Model

In most route choice models, the driver's preference for an alternative path is quantified by a value called utility and the driver is assumed to choose the alternative path with the highest utility. In some application contexts, the characteristics of the individual drivers are considered in the calculation of utility. In this work, we consider all drivers to be identical.

Given a set of alternative paths, called the choice set, C, the utility associated with alternative path $p_i \in C$ is given by

$$U_i = V_i + \varepsilon_i \qquad (7)$$

where V_i is a deterministic term and ε_i is a random term for capturing the uncertainty involved. The deterministic term V_i is modelled as a linear-in-parameters function of the attributes of alternative path p_i. We use the following attributes to determine the utility of a path as these factors are generally known to influence drivers' route preferences. (We also considered the length of the path, but tests showed that it does not appear to be significantly correlated with route choice.)

1. Free-flow travel time (FTT; in seconds)
2. Number of traffic signals (NTS)
3. Average road class (ARC; road classes are numbered from 1, starting with the highest)
4. Number of class changes (NCC)

The deterministic term of the utility for alternative path p_i is given by

$$V_i = \beta_{FTT} FTT_i + \beta_{NTS} NTS_i + \beta_{ARC} ARC_i + \beta_{NCC} NCC_i \qquad (8)$$

where FTT_i is the FTT attribute of alternative path p_i, β_{FTT} is a coefficient parameter corresponding to the FTT attribute, and so on. For convenience, we rewrite (8) as

$$V_i = \boldsymbol{\beta}' \boldsymbol{x}_i \qquad (9)$$

where $\boldsymbol{\beta}'$ is a vector of coefficient parameters $(\beta_{FTT}, \beta_{NTS}, \beta_{ARC}, \beta_{NCC})$ and $\boldsymbol{x}_i$ is a vector of attributes $(FTT_i, NTS_i, ARC_i, NCC_i)$ corresponding to the alternative path p_i.

In a multinomial logit model, the random term ε_i in (7) is assumed to be independent and identically Gumbel distributed. The scale factor of the random term is irrelevant to the choice of the alternative with the highest utility [23] and is therefore typically normalized to one. Without loss of generality, the mean of the random term can be assumed to be zero if an alternative-specific constant is included in the calculation of the deterministic term of the utility. The alternative-specific constant, which is an unknown parameter to be estimated, captures the bias towards an alternative due to the unobserved factors. However, the notion of alternative-specific constants is not intuitively meaningful in the context of route choice, where alternative paths in different situations may not have anything in common with each other. For instance, it does not make sense to suppose that the k^{th} alternative path in one choice set has the same level of bias as that of the k^{th} alternative path in another choice set. We, therefore, do not include alternative-specific constants in the model. In effect, we assume that the unobserved factors have the same impact on all the alternative paths. Overall, the route choice model used in our work can be considered as a reasonable approximation of the multinomial logit model. This simplified model gives the probability of a driver choosing alternative path p_i within the choice set C as

$$P(p_i|C) = \frac{e^{\beta' x_i}}{\sum_{p_j \in C} e^{\beta' x_j}} \tag{10}$$

B. Choice Set Generation

In order to estimate or apply a route choice model, it is necessary to have a process for generating the alternative paths that exist between two points on a road network. For any origin-destination pair in a road network, the number of all possible alternative paths between them could be very large. However, this may include many highly circuitous and unreasonable paths that are irrelevant in the context of route choice. We, therefore, aim to generate a choice set containing a limited number of reasonable and distinct alternative paths.

For finding multiple paths between a given origin and destination, a k-shortest-paths algorithm [24] is generally considered to be the straightforward approach. However, it has been empirically found that k-shortest-paths algorithms tend to produce paths that are highly similar [25]. A commonly observed example involves a path along a freeway and an alternative path that repeatedly exits and re-enters the freeway through slip roads at interchanges. Another approach, known as the link-penalty approach, performs repeated shortest-path computations and minimizes overlapping among paths by imposing a penalty on links in paths that are already found. Our proposed method for choice set generation, explained below, is broadly based on the link-penalty approach.

It needs to be noted that we, in this work, are concerned with short paths whose travel times are typically in the order of a few minutes. These short paths may not correspond to the complete trips made by drivers (e.g. from home to work). Rather, the short paths that are of interest to us connect pairs of points that lie along the overall paths taken by drivers. It is worth recalling that we intend to use a route choice model for improving the partial paths generated by the HMM-based online map-matching method described earlier. The time points at which the vehicle was at the start and end points of the partial paths and therefore the elapsed time between them are assumed to be available.

Our analysis of data from real drives shows that between points separated by a few minutes of travel, there are rarely more than 3 or 4 reasonable alternative paths. Therefore, we limit the choice set to a maximum of 5 alternative paths. The first alternative path in the choice set is pre-identified and made available. During the estimation of the route choice model, the true path chosen by the driver is included as the

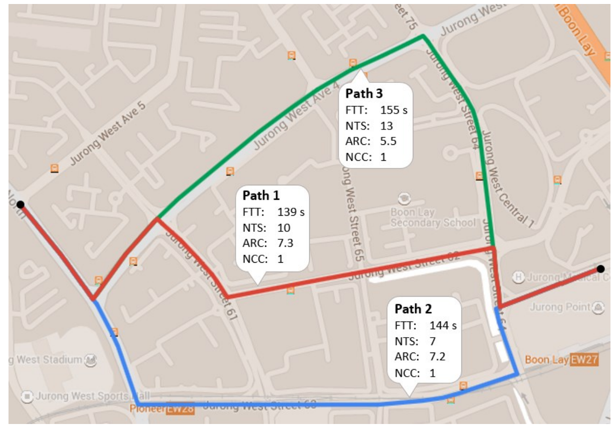


Fig. 4. An example of multiple alternative paths produced by the choice set generation method (Base map image: Google Maps).

first alternative in the choice set. During the application of the model, the HMM-generated partial path is considered as the first alternative path in the choice set. As paths found based on the minimum-travel-time criterion are known to account for most of the paths chosen by real drivers [26], the minimum-travel-time path is always included in the choice set.

After the pre-identified first alternative path and the minimum-travel-time path (if different from the former) are added to the choice set, we apply a penalty to the links that are part of the above paths. This is done by inflating the travel times of the relevant links by a factor in order to reduce the chances of them being part of subsequent alternative paths found based on the minimum-travel-time criterion. However, as pointed out in [25], applying a penalty to links that are close to the start point or end point of the paths may cause unreasonable paths to be found subsequently. Therefore, while inflating the travel time of a link that lies along a path in the choice set, we take the distances along the path from the start point to the link and from the link to the end point into consideration. The inflated travel time of link e, with start node u and end node v, lying along path p_i between start point q and end point r is given by

$$\tau'(e) = \tau(e) + \tau(e)\,\omega\,\frac{min\big(d_i(q,u), d_i(v,r)\big)}{d_i(q,r)} \tag{11}$$

where $\tau(e)$ is the original travel time of link e, ω is the inflation factor and the function d_i gives the driving distance between two points along path p_i. Our trials show that an inflation factor value of 5 produces the best results in practice.

Subsequently, we compute the minimum-travel-time path thrice. After each path computation, the travel times of the links in the path are inflated as explained above. The computed path is included in the choice set only if it does not overlap any of the alternative paths already present in the choice set by more than a proportion α, which we empirically set to 50%. Also, the computed path should be temporally possible in order to be added to the choice set. Allowing for a high degree of speeding, a path is considered to be temporally possible if its free-flow travel time is not more than thrice the actual elapsed time between its start and end points. In this manner, the choice set is populated with a maximum of 5 alternative paths. Figure 4 shows an example case of three alternative paths generated by the above procedure along with their attributes. The overall flow of the choice set generation method for a given origin and destination is as follows.

1. Add pre-identified path p_p to choice set C.
2. Compute minimum-travel-time path p_m.
3. Add p_m to C if different from p_p.
4. Inflate the travel times of all links in p_p and p_m.
5. Compute the current minimum-travel-time path p_m.
6. Add p_m to C if it does not overlap any path in C by more than 50% and is temporally possible.
7. Inflate the travel times of all links in p_m.
8. Repeat steps 5-7 twice.
9. Reset the travel times of all links to original values.

C. *Model Estimation*

The β parameters in the multinomial logit model (see equations (8) – (10)) can be estimated from a sample of choice observations. In the context of route choice, the choice observations correspond to path choices made by real drivers. A choice observation n consists of a choice set C_n with several alternative paths and information about which alternative path was actually chosen by a driver. Given a path chosen by a driver between two points during a real drive, it is included as the first alternative in a choice set. Subsequently, other feasible alternatives paths are generated using the choice set generation method. This process is repeated for a number of chosen paths obtained from real drive data. Maximum likelihood estimation is commonly used for estimating the parameters of route choice models. It estimates the value of the parameters for which the sample of choice observations is most likely to have occurred [27].

IV. The Overall Map-Matching Algorithm

Our complete solution for online map-matching of highly noisy and sparse location data utilizes the HMM-based online map-matching module presented in Section II as well as the route choice model described in Section III. The proposed overall solution is as described below.

The HMM module receives and processes location observations from a vehicle-based user on the fly. Each time a convergence condition is detected by the Viterbi algorithm in online mode as described in Section II-C, the partial path up to the convergence state is released for further processing. We refer to this path as the HMM-generated partial path. The convergence states correspond to points on the road network where the vehicle's location is matched with a relatively high degree of certainty. Therefore, it can be said that the start and end points of the HMM-generated partial paths are generally accurate, while the paths themselves may have a degree of error associated with them. Multiple alternative paths are generally plausible between the start and end points of the HMM-generated partial path.

Let q and r be the start and end points, respectively, of a HMM-generated partial path. The choice set generation procedure described in Section III-B is applied to generate a

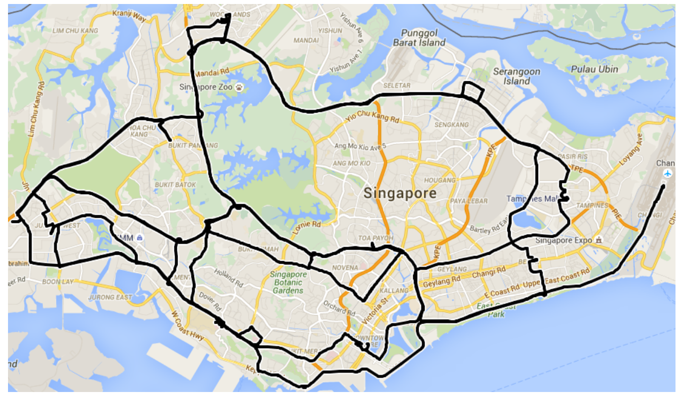


Fig. 5. The roads covered during the taxi-based data collection in Singapore are shown in black (Base map image: Google Maps).

number of reasonable and largely distinct alternative paths between q and r. As noted earlier, the choice set unconditionally includes the HMM-generated partial path as one of the alternatives.

We now face the task of identifying the most-likely path from among the several alternative paths in the choice set. We consider the alternative path that has a high probability of being chosen by a driver and is highly consistent with the sequence of location measurements as the most-likely path. Towards this end, for each alternative path, we calculate two probability measures that we refer to as the route choice probability and the observation generation probability, respectively.

The route choice probability $P(p_i|C)$, which is the probability of a driver choosing the alternative path p_i within the choice set C, is calculated as given in (10) based on the multinomial logit model. The observation generation probability of an alternative path is the probability that the given location observations were generated while traveling along that path. Let $o_{1:K}$ be a sequence of K location observations corresponding to the alternative paths in the choice set. Assuming Gaussian-distributed location measurement errors, we define the observation generation probability for a given alternative path p_i as

$$P(o_{1:K}|p_i) = \prod_{k=1}^{K} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{g_c(o_k,p_i)^2}{2\sigma^2}} \tag{12}$$

where σ is the standard deviation of the location measurement error to be empirically estimated and $g_c(o_k, p_i)$ is the great-circle distance between o_k and the point closest to it on p_i.

Among all the alternative paths in the choice set, the one for which the product of the route choice probability and the observation generation probability is the highest is identified as the most likely path taken by the vehicle between the start and end points. This forms the partial output of the overall map-matching algorithm.

The steps in the proposed overall solution can be summarized as follows:

1. If a convergence condition is detected in the Viterbi algorithm, produce the HMM-generated partial path.
2. Generate a choice set of multiple alternative paths

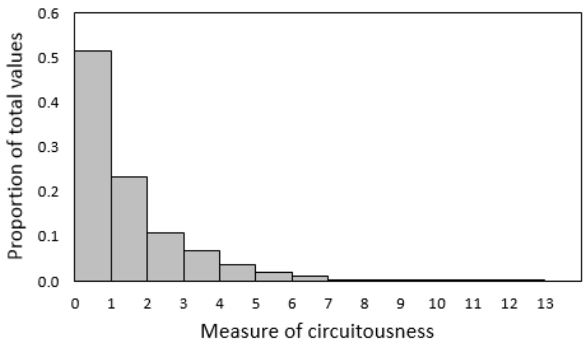


Fig. 6. Histogram of the measure of circuitousness.

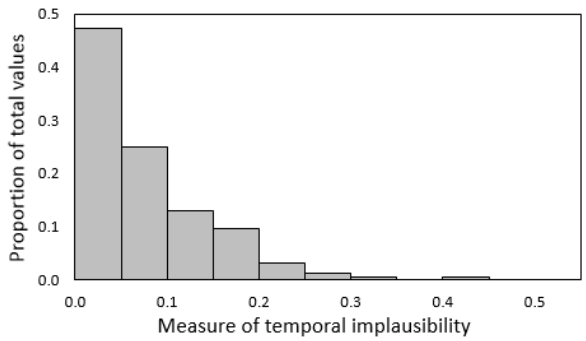


Fig. 7. Histogram of the measure of temporal implausibility (with only the non-zero values considered).

including the HMM-generated partial path.

3. For each alternative path, compute the route choice and observation generation probabilities.
4. Output the alternative path that maximizes the product of the route choice and observation generation probabilities and go to step 1.

V. Evaluation

We evaluate the proposed map-matching solution using real as well as synthetic location data. In the following, we first describe the processes of collecting data, identifying the ground truth and estimating the model parameters of the HMM and the route choice model. Subsequently, we describe the evaluation setup and present the results.

A. Data Collection and Ground Truth Determination

We use noisy location data based on cellular network positioning for our evaluation. The location data were collected during taxi trips made in Singapore using an Android smartphone with the GPS and Wi-Fi functions disabled. For this purpose, we used an application that sends the cellular base station fingerprints to the Google location server and obtains the location estimates from the server through the Android location application program interface. Our test dataset consists of 21807 location points recorded mostly at the rate of one point per second during 20 taxi trips, which covered a total distance of 421 km. The roads covered during the taxi trips are shown in Figure 5.

The ground truth locations and paths corresponding to the above trips were determined using GPS data recorded at the rate of one sample per second using another smartphone. We first applied the HMM-based map-matching method described in Section II to the GPS locations, whose measurement errors are typically in the order of a few meters. The map-matched locations and paths thus obtained were subjected to thorough manual inspection and correction with the aid of the knowledge about the actual paths taken during the taxi trips. While the ground truth paths obtained in the above manner are 100% accurate, the ground truth locations corresponding to the measured locations could be off by up to a few meters due to the unavoidable GPS error. However, this is a reasonable approximation considering the fact that the typical measurement error of the cellular network positioning data used in this work is in the order of several hundred meters.

TABLE I
Estimated Parameters of the Route Choice Model

Parameter	Corresponding attribute	Estimated value	t–Test value
β_{FTT}	Free-flow travel time (s)	-0.019	-6.97
β_{NTS}	Number of traffic signals	-0.100	-5.48
β_{ARC}	Average road class	-0.244	-2.25
β_{NCC}	Number of class changes	-0.272	-4.68

An alternative ground truth dataset created using the GPS tracks from a separate set of 1000 drives totaling 13139 km was used for estimating the path-related parameters explained in Sections II-C and III-A. These GPS tracks, provided by a Singapore-based ITS company, were collected and anonymized from the users of a mobile application.

B. Parameter Estimation

The first parameter to be estimated is the standard deviation of the location measurement error σ used in (1) and (12). For each location data point obtained through cellular network positioning during the taxi trips, the measurement error is nothing but the great-circle distance between itself and the corresponding ground truth location. For the test dataset, the mean, median and worst-case measurement errors were 300 m, 258 m and 4652 m, respectively. In line with the practice followed by several map-matching algorithms [4][6], we calculate the standard deviation of the measurement error using the Median Absolute Deviation (MAD), which is a robust estimator resilient to outliers. If $\{g_i\}$ is the set of all measurement errors in the test set and if the errors are assumed to be Gaussian distributed with zero mean, the standard deviation is given as

$$\sigma = 1.4826\, median_i(g_i) \tag{13}$$

For the test dataset, σ is 382 m.

The transition probability model given by (4) requires the parameters λ_y and λ_z, corresponding to the measures of path circuitousness and temporal implausibility, respectively, to be estimated. It needs to be recalled that the transition probability depends on the features of the short paths that connect states corresponding to two successive time steps. For estimating λ_y, we use the ground truth paths in the alternative ground truth dataset. Each ground truth path is randomly segmented into multiple short paths, whose travel durations range from 1 minute to 5 minutes. Overall, there are a total of 4828 paths with an average length of 2.6 km.

As evident from the histogram shown in Figure 6, the values of the measure of circuitousness for all the paths follow an exponential distribution. The rate parameter λ_y of the exponential distribution can be estimated as the inverse of the mean of all such values. For the alternative ground truth data set, λ_y is 0.69. Contrary to our expectation, the values of the measure of temporal implausibility for all the paths do not fit well to an exponential distribution. This is because for about 95% of the paths, the measure of temporal implausibility is zero. However, we are mainly interested in the paths with a non-zero measure of temporal implausibility. When only such paths are considered, the values of the measure of temporal implausibility follow an exponential distribution as shown in Figure 7. The rate parameter λ_z of this distribution is estimated to be 13.35.

For estimating the β parameters in the route choice model explained in Section III-A, we use as input a number of true paths that are obtained by randomly segmenting the paths in the alternative ground truth dataset. A total of 2016 such paths are used. These paths, which correspond to portions of real drives, have an average travel duration of approximately 6 minutes and an average length of 4.2 km. Each of these true paths is included as the first alternative path in a choice set. The choice set generation procedure is used to generate up to 4 alternatives to the true path. Thus, each choice observation consists of a choice set with up to 5 alternatives, with the first alternative being the chosen one. The β parameters, estimated by maximum likelihood using the Biogeme [28] software package, are presented in Table I. All the estimated parameters are statistically significant ($p < 0.05$).

C. *Evaluation Setup and Metrics*

We use the following two sets of location sequence data as inputs for evaluating the proposed map-matching algorithm:

1. A *real* dataset consisting of the cellular network positioning data collected during the taxi trips, but made temporally sparse by subsampling at intervals ranging from 1 to 5 minutes.
2. A *synthetic* dataset created by adding random Gaussian noise to the ground truth locations corresponding to the points in the real dataset. The standard deviation of the added Gaussian noise ranges from 200 m to 1000 m. For each noise level, the input locations are sampled at intervals ranging from 1 to 5 minutes.

The F-score, which is the harmonic mean of precision and recall, is used to evaluate the accuracy of the algorithm. For a path output by the map-matching algorithm, the F-score, precision and recall are defined as

$$F\_score = 2 \times \frac{precision \times recall}{precision + recall} \tag{14}$$

$$precision = \frac{L_{correct}}{L_{matched}} \tag{15}$$

$$recall = \frac{L_{correct}}{L_{truth}} \tag{16}$$

where $L_{matched}$ is the length of the output path, L_{truth} is the length of the corresponding ground truth path and $L_{correct}$ is the length of the portions of the output path that overlap with the ground truth path.

We also evaluate the timeliness of the algorithm. Timeliness is quantified by the input-to-output latency, which in this context is the time elapsed between the time a location observation was received and the time when the map-matched path corresponding to it is output.

We compare the accuracies and the mean latencies of the following three map-matching solutions under varying levels of spatial inaccuracy and temporal sparsity.

1. The HMM-plus-route-choice-model (*HMM+RCM*) algorithm: This is the proposed overall solution that uses the HMM and the route choice model.
2. The *HMM-only* algorithm: This solution includes only the HMM-based online map-matching method presented in Section II.
3. The *Newson-Krumm* algorithm: This is the HMM-based algorithm proposed in [4] that is widely regarded as the state of the art for map-matching noisy and sparse location data.

D. *Results*

Figure 8 shows the accuracies (F-scores) of the compared solutions when applied to the real dataset with sampling intervals ranging from 1 to 5 minutes. For all sampling intervals, the accuracy of the proposed HMM+RCM algorithm is significantly higher than the Newson-Krumm algorithm. The HMM+RCM algorithm also outperforms the HMM-only algorithm, especially at longer sampling intervals. The accuracy of the proposed methods degrade gracefully as the sampling interval increases. Averaging over all sampling intervals, the accuracies of the HMM+RCM algorithm, the HMM-only algorithm and the Newson-Krumm algorithm are 91.3%, 89.6% and 81.2%, respectively.

The mean latencies observed when the algorithms are applied to the real dataset are presented in Figure 9. Quite predictably, the mean latency increases with increasing sample interval for all the algorithms. (The HMM-only and the HMM+RCM algorithms use the same HMM module and hence have the same latency.) The Newson-Krumm algorithm achieves relatively lower mean latencies at low sampling intervals but not at higher sampling intervals.

We use the synthetic dataset to study the effect of measurement noise on the map-matching performance. Figure 10 presents the average accuracy results obtained under varying levels of noise with all the sampling intervals (1 to 5 minutes) considered. While the Newson-Krumm algorithm degrades drastically with increasing levels of noise, the HMM+RCM and HMM-only algorithms display a high degree of noise robustness. The HMM+RCM algorithm manages to achieve an accuracy of over 80% even when the standard deviation of measurement noise is increased to 1000 m.

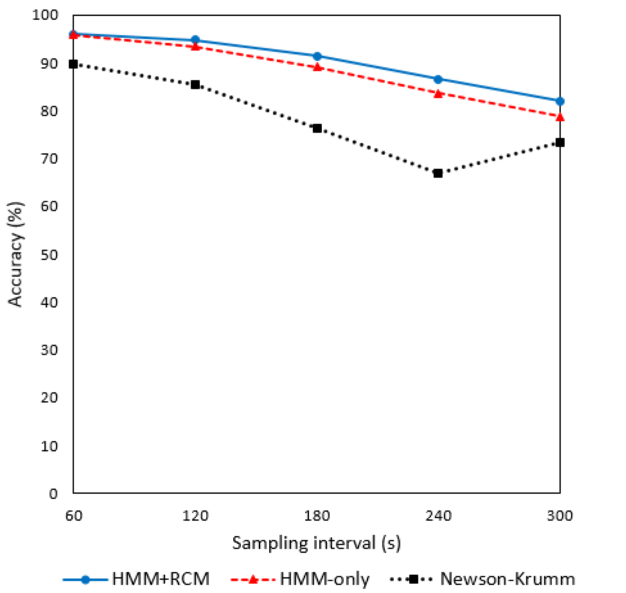


Fig. 8. Comparison of map-matching accuracy on the real dataset for various sampling intervals.

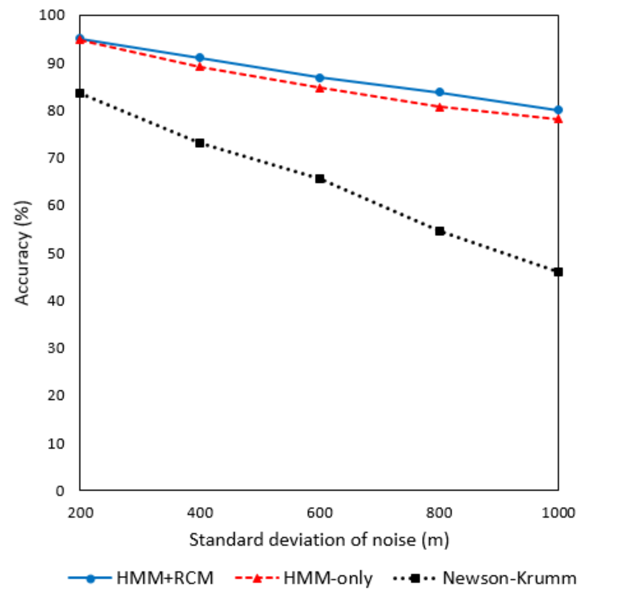


Fig. 10. Comparison of map-matching accuracy on the synthetic dataset for various levels of noise (averaged over all sampling intervals).

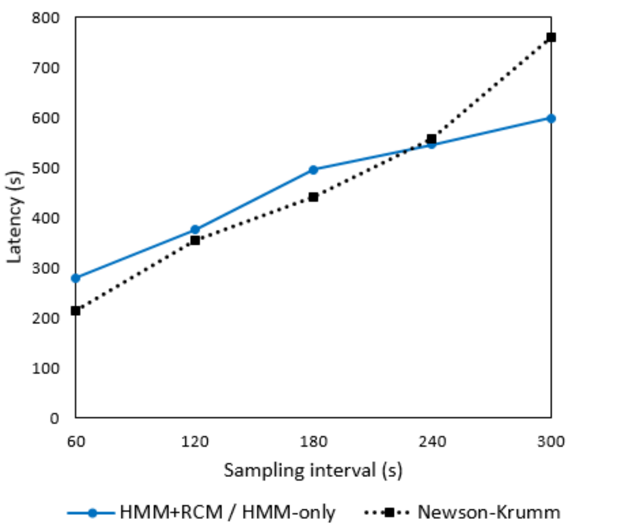


Fig. 9. Comparison of mean latency on the real dataset for various sampling intervals.

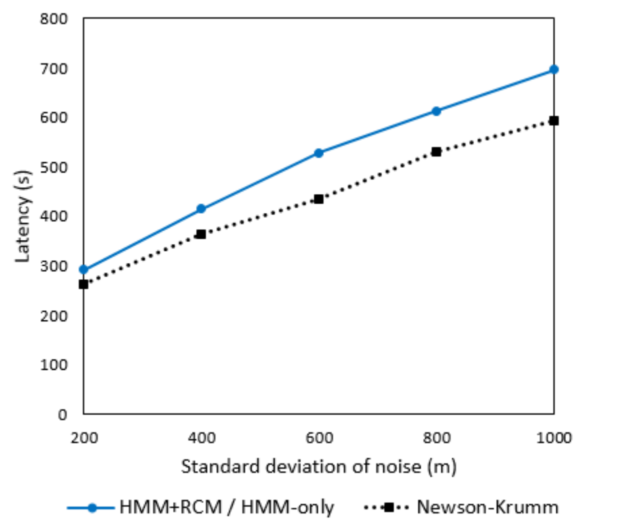


Fig. 11. Comparison of mean latency on the synthetic dataset for various levels of noise (averaged over all sampling intervals).

Figure 11 shows the relationship between the mean latency and the noise level. The mean latency of the HMM+RCM and HMM-only algorithms increases steadily with increasing levels of noise and is moderately higher than that of the Newson-Krumm algorithm. It may be worth noting that when all the sampling intervals are aggregated, the location sequences with low sampling intervals contribute more points and thus have more influence on the overall mean latency. As seen earlier in Figure 9, the mean latency of the Newson-Krumm algorithm is lower at low sampling intervals.

The accuracy comparison between the HMM-only algorithm and the Newson-Krumm algorithm is particularly interesting because both the algorithms differ only in the transition probability model used. We attribute the significant difference in the accuracies of the above two algorithms to the following differences between their respective transition probability models. First, while the Newson-Krumm model assumes that transitions between states occur through the shortest-distance path, we use the minimum-travel-time path, which better represents reality. Our analysis of GPS tracks from real drives show that in most cases, vehicles move from one point to another through the minimum-travel-time path along major roads even when a shorter-distance path involving minor roads exist between those points. Second, as explained in Section II B, there are some drawbacks in the distance difference measure used by Newson and Krumm to quantify the circuitousness of a transition path. It relies on the distance between the current and previous location observations and therefore is adversely affected by the location measurement errors, especially in the case of highly noisy data. Also, Newson and Krumm appear to ignore the fact that the distribution of their distance difference measure varies significantly for different sampling intervals. In contrast, our measure of circuitousness takes the sampling interval into consideration and is independent of the location observations.

Finally, unlike the Newson-Krumm model that does not consider the travel times of the transition paths, our model includes a measure of temporal implausibility that helps to assign low transition probabilities for paths whose free-flow travel times are higher than the sampling interval.

We believe that the accuracy results presented above are superior to those reported by other works in the literature that dealt with comparable levels of measurement noise (e.g. [2] and [14]). The results appear to suggest that the impact of the route choice model becomes more apparent at higher levels of temporal sparsity and noise. This is quite intuitive, as noisy and sparse location data gives rise to a higher number of plausible paths between the convergence states in the HMM, thereby justifying the use of a route choice model to identify the most likely path.

VI. Conclusions

As a considerable portion of location data crowdsourced from road users is highly inaccurate and sparse, it is important to develop robust algorithms capable of map-matching such data in an accurate and timely manner. In this paper, we have presented an innovative solution for the above problem. It extends the HMM-based state-of-the-art approach for online map-matching by combining it with a route choice model estimated from real drive data. We propose and use a choice set generation procedure for generating a number of reasonable alternatives for each partial path found using the HMM. The alternative paths are assessed based on their route choice probabilities given by the route choice model as well as their consistency with the observed locations in order to identify the most likely path.

We have evaluated the proposed algorithm using real-world location data obtained through cellular network positioning as well as synthetic data with high levels of added measurement noise. The results show that the algorithm achieves substantially higher accuracy compared to the state of the art besides being robust to high levels of noise and sparsity. The main implication of the results is that it appears feasible to effectively utilize smartphone location sequence data sourced from users at low sampling frequencies using energy-efficient positioning technologies. We believe that users are more likely to allow their locations to be sampled if the sampling is performed in the above manner without placing undue demands on the smartphone's bandwidth and battery power.

There are a few avenues available for extending this work. In particular, there is a need to further improve the timeliness of the proposed map-matching algorithm. A possible way of doing this is to explore probabilistic approaches [20] for heuristically limiting the latency of the algorithm without any significant loss of accuracy. Another area of future work is to validate the proposed solution in a practical deployment scenario where location observations with varying degrees of accuracy are received from a large number of users with heterogeneous travel behavior. It is also desirable to develop a more advanced route choice model that takes the attributes of the driver and other factors such as the time of travel (e.g. peak/off-peak hour) and prevailing traffic conditions into consideration. A more open-ended future task is to explore the possibility of overcoming the limitation arising out of the Markov independence assumption in the HMM by developing a higher-order probabilistic graphical model, where the transition probability depends on several past and future states. It would be interesting to compare the map-matching solution based on such a model with the proposed HMM+RCM algorithm.

References

[1] M.A. Quddus, W.Y. Ochieng, and R.B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15(5), pp. 312–328, 2007.

[2] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *8th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, 2011.

[3] P. A. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning," *Transactions in GIS*, vol. 13(s1), pp. 5-25, 2009.

[4] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 336-343, 2009.

[5] A. Thiagarajan, L. S. Ravindranath, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan, "VTrack: Accurate, energy-aware traffic delay estimation using mobile phones," In *7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 85-98, 2009.

[6] C.Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on Hidden Markov model for real-time traffic sensing applications," in *15th IEEE Intelligent Transportation Systems Conference*, pp. 776-781, 2012.

[7] T. Hunter, P. Abbeel, and A. M. Bayen, "The path inference filter: model-based low-latency map matching of probe vehicle data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15(2), pp. 507-529, April 2014.

[8] J. Yang, and L. Meng, "Feature selection in conditional random fields for map matching of GPS trajectories", in *Progress in Location-Based Services 2014*, Springer International Publishing, pp. 121-135, 2015.

[9] M. Rahmani, and H. N. Koutsopoulos, "Path inference from sparse floating car data for urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 30, pp. 41-54, 2013.

[10] G. Sarlas, "Processing low-frequency floating car data for transportation applications," Master's thesis, Division of Traffic and Logistics, KTH Royal Institute of Technology, Stockholm, Sweden, 2013.

[11] M. Ben-Akiva and M. Bierlaire, "Discrete choice methods and their applications to short term travel decisions," in *Handbook of Transportation Science*, Springer, pp. 5-33, 1999.

[12] T. Miwa, D. Kiuchi, T. Yamamoto, and T. Morikawa, "Development of map matching algorithm for low frequency probe data," *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 132–145, 2012.

[13] M. Bierlaire, J. Chen, and J. Newman, "A probabilistic map matching method for smartphone GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 78-98, 2013.

[14] G. Schulze, C. Horn, and R. Kern, "Map-matching cell phone trajectories of low spatial and temporal accuracy," in *18th IEEE Intelligent Transportation Systems Conference*, Las Palmas, Spain, pp. 2707-2714, 2015.

[15] G. R. Jagadeesh and T. Srikanthan, "Probabilistic map matching of sparse and noisy smartphone location data," in *18th IEEE Intelligent Transportation Systems Conference*, Las Palmas, Spain, pp. 812-817, 2015.

[16] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Transactions on Information Theory*, vol. 13(2), pp. 260-269, 1967.

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[18] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue, "Map matching with hidden Markov model on sampled road network," in *21st International Conference on Pattern Recognition*, Tsukuba, Japan, pp. 2242-2245, 2012.

[19] H. Koller, P. Widhalm, M. Dragaschnig, and A. Graser, "Fast hidden Markov model map-matching for sparse and noisy trajectories," in *18th IEEE Intelligent Transportation Systems Conference*, Las Palmas, Spain, pp. 2557-2561, 2015.

[20] M. Narasimhan, P. Viola, and M. Shilman, "Online decoding of Markov models under latency constraints," in *23rd international conference on Machine learning*, ACM, pp. 657-664, 2006.

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

[22] M. Srivatsa, R. Ganti, J. Wang, and V. Kolar, "Map matching: facts and myths," in *21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, pp. 484-487, 2013.

[23] K. E. Train, "Properties," in *Discrete Choice Methods with Simulation*. Cambridge, UK: Cambridge University Press, 2003.

[24] D. Eppstein, "Finding the *k* shortest paths," *SIAM Journal on Computing*, vol. 28(2), pp. 652-673, 1998.

[25] D. Park and L. Rilett, "Identifying multiple and reasonable paths in transportation networks: A heuristic approach," *Transportation Research Record: Journal of the Transportation Research Board*, 1607, pp. 31-37, 1997.

[26] M. Ben-Akiva, M. J. Bergman, A. J. Daly, and R. Ramaswamy, "Modeling inter urban route choice behavior," in *9th International Symposium on Transportation and Traffic Theory, Delft, Netherlands*, pp. 299-330, 1984.

[27] M. Ben-Akiva and S. R. Lerman, "Multinomial Choice," in *Discrete choice analysis: theory and application to travel demand*. Cambridge, MA: MIT Press, 1985.

[28] M. Bierlaire, "BIOGEME: a free package for the estimation of discrete choice models," in *3rd Swiss Transport Research Conference*, Ascona, Switzerland, 2003.