

Robust real-time route inference from sparse vehicle position data

Jagadeesh, George Rosario; Srikanthan, Thambipillai

2014

Jagadeesh, G. R., & Srikanthan, T. (2014). 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). Robust real-time route inference from sparse vehicle position data, 296-301.

<https://hdl.handle.net/10356/79437>

<https://doi.org/10.1109/ITSC.2014.6957707>

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/ITSC.2014.6957707>].

Downloaded on 30 Nov 2023 23:21:54 SGT

Robust Real-Time Route Inference from Sparse Vehicle Position Data

G. R. Jagadeesh and T. Srikanthan

Abstract— The ability to correctly infer the route traveled by vehicles in real time from infrequent, noisy observations of their position is useful for several traffic management applications. This task, known as map matching, is efficiently performed through probabilistic inference on a Hidden Markov Model that represents the candidate vehicle states and the transitions between them. In this paper, we present new methods for improving the accuracy and timeliness of existing solutions. We propose assigning the transition probability between a pair of candidate vehicle states by considering the alternative paths present in the context. A discrete route choice model is used to estimate the probability that a driver would choose the path under consideration over the best alternative available. In order to facilitate real-time operation, we present a simple yet effective heuristic to reduce the output latency of the route-inference algorithm with negligible loss of accuracy. Tests conducted with ground truth GPS data from a dense urban region in Singapore show that the proposed techniques outperform the conventional baseline approach.

I. INTRODUCTION

Accurate and timely data on traffic conditions can help to manage and mitigate the problem of traffic congestion. However, in many urban road networks, good quality traffic data is scarce due to the substantial cost and difficulty involved in correctly installing and maintaining roadside traffic sensing equipment. In this context, the proliferation of GPS devices in recent years has offered a valuable alternative for estimating the traffic conditions in the network. On-road vehicles equipped with GPS devices act as probes and periodically transmit their timestamped positions to a central system, which infers their traveled routes and estimates the traffic conditions along the routes.

The task of inferring the traveled route from the reported GPS positions of a vehicle is known as map matching. This is often a difficult task due to two reasons. First, the GPS positions have an associated error. While tests indicate that the error is less than 30 m in about 95% of the samples [1], it can be much higher in urban canyon environments due to satellite occlusions and multipath reflections. This error makes it non-trivial to match a given GPS position to a location on the road network, especially near intersections where a number of road segments lie in close proximity. The second reason is that, due to bandwidth considerations, the probes transmit GPS data at a low frequency (for instance, once in several minutes) [2]. This makes it hard to interpolate the path travelled by the vehicle between two reported positions, especially in dense areas of the network.

The problem of map matching high-frequency GPS data (typically, once every second) has received considerable attention and a number of methods have been proposed in the late 1990s and the early 2000s. A comprehensive review of the various methods can be found in [3]. The focus of this paper, however, is on the more difficult problem of map matching GPS data sampled at low frequencies.

Several algorithms have been proposed, especially in recent years, for inferring routes from sparsely-sampled probe vehicle data. They can be broadly grouped into two categories: (i) deterministic methods relying on geometric and topological criteria and (ii) probabilistic inference methods. An early deterministic method that map matched GPS data sampled at 30s intervals was proposed by Brakatsoulas et al. [4]. They used the Fréchet distance measure to determine the geometric similarity of the GPS trajectory and the candidate routes. Zheng and Quddus [5] compared the length and heading of the straight line between two consecutive GPS positions with those of the corresponding candidate routes in order to estimate a map-matching score. The idea of forming a set of reasonable candidate routes for a given sequence of GPS positions and selecting the best among them based on a route choice model has also been proposed [6] [1].

Due to the high degree of ambiguity caused by the GPS errors and the low sampling rates, map-matching methods based on probabilistic inference are generally favored over deterministic methods. The unknown true locations of the vehicle that correspond to the GPS observations and the transition paths between them can be represented using a Hidden Markov Model (HMM). Hence, a number of HMM-based and HMM-inspired probabilistic methods have been proposed for route inference from sparse position data [7] [8] [9] [10] [2]. An alternative, but closely related probabilistic map-matching approach based on a Conditional Random Fields model has been proposed in [11].

An important consideration in map matching is the tradeoff between accuracy and timeliness. A class of algorithms, known as global algorithms, process the entire sequence of GPS positions in order to produce an optimal solution. However, global algorithms are unsuitable when real-time solutions are required. In such scenarios, online algorithms are used. Online algorithms generate possibly suboptimal partial solutions without waiting for future position samples.

The work presented in this paper is based on the HMM framework. We propose two improvements to the state of the art. First, we show that while assigning the transition probability for a candidate path between two candidate vehicle states, it is beneficial to examine the wider context for the existence of better alternative paths. We use a

The authors are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (phone: +65 67906643; e-mail: {asgeorge, astsrikan}@ntu.edu.sg).

discrete route choice model to estimate the likelihood that the candidate path would be chosen over the best alternative available. Second, we examine ways to improve the timeliness of the HMM-based solution for map matching without significantly compromising the accuracy. We propose a simple heuristic to reduce the output latency of the Viterbi algorithm used to determine the most likely state sequence in the HMM. The effectiveness of the proposed methods are evaluated against a conventional baseline using ground-truth annotated GPS data collected in Singapore.

The rest of the paper is organized as follows. Section 2 provides a formal description of the problem. The baseline HMM framework for solving the map-matching problem is described in Section 3. The new improved method for assigning the transition probabilities in the HMM using a discrete route choice model is presented in Section 4. Section 5 focuses on reducing the output latency of Viterbi inference in the HMM. The experimental setup and the results obtained are presented in Section 6. Finally, Section 7 summarizes and concludes the paper.

II. PROBLEM STATEMENT

A *road network* is represented as a directed graph $N(V,E)$, where V is a set of nodes representing intersections or end points of the road segments and E is a set of edges representing road segments. The geometry of each road segment is stored as a piecewise linear curve. Each road segment has a number of attributes such as length, road class, speed limit, etc. Let a *route* $R = (e_k \mid k = 1, \dots, K)$ be an ordered sequence of K road segments e_1, \dots, e_K .

Let g_t be a *GPS observation* reported by a probe vehicle at time step t . Each GPS observation consists of a longitude, latitude and a timestamp. Let $G = (g_t \mid t = 1, \dots, T)$ be a *GPS trajectory*, or a sequence of GPS observations g_1, \dots, g_T , from time step 1 to time step T .

Given a sequence of GPS observations g_1, \dots, g_T corresponding to T time steps, the problem is to find the route R^* , which is most likely to have generated the given GPS observation sequence.

III. THE HMM FRAMEWORK FOR MAP MATCHING

Due to the GPS errors, the true location or *state* of the probe vehicle that yielded a given GPS observation is generally unknown. Instead, each GPS observation is associated with a number of *candidate states* that could have potentially generated it. Each candidate state represents a location on a nearby road segment. For each road segment in the vicinity (e.g. within a 100m range), the point closest to the GPS observation is chosen as the candidate state. Let $s_{i,j}$ denote the j^{th} candidate state associated with the GPS observation g_i reported at time step i . As an example, Figure 1(a) shows three GPS observations g_1, g_2 and g_3 on a simplified road network. For clarity, each GPS observation is shown to have only two candidate states. The dotted and dashed lines indicate two of the many possible routes that are consistent with the given sequence of GPS observations. Each route passes through one of the candidate states of each time step. Figure 1(b) shows a graphical representation of all the candidate states and all possible transitions between them for the given example. The

route inference problem can be viewed as one of identifying the most likely path in the graph that passes through one of the candidate states of each time step.

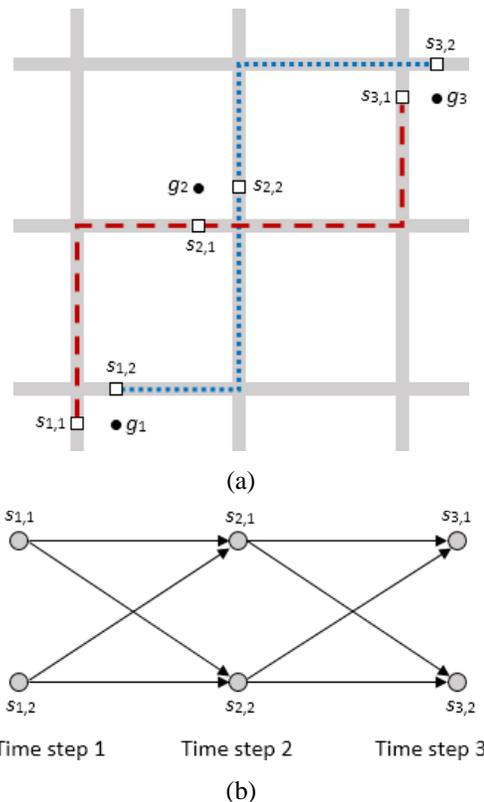


Figure 1. (a) An example showing GPS observations, candidate states and two of the possible routes (b) A graph of the candidate states and transitions.

A Hidden Markov Model, with some customizations, is well-suited for modeling the problem described above. The unknown true location of the vehicle at each time step can be considered as a hidden state that could be one of the candidate states for that time step. Every candidate state at time step t could emit the GPS observation g_t with a probability distribution known as the emission probability. Each candidate state at time step t can transition to each of the candidate states at time step $t+1$ with a transition probability. Given a sequence of observations, the corresponding state space, the emission probabilities and the transition probabilities, the most likely sequence of hidden states can be inferred using the Viterbi algorithm. In the following subsections, we describe the conventional approach towards assigning the emission and transition probabilities for HMM-based map matching and explain the Viterbi inference process.

A. Emission Probability

For a given GPS observation and a candidate state, the emission probability represents the likelihood of the former being observed if the probe vehicle is on the latter. Intuitively, a GPS observation closer to the candidate state should result in a higher emission probability compared to one that is far away. If the candidate state is indeed the true location of the vehicle, then the distance between the candidate state and the GPS position is the GPS error. All the probabilistic map matching methods known to us assume that the GPS error follows a zero-mean normal distribution. On that basis, the

emission probability of a candidate state $s_{i,j}$ for a GPS observation g_i is given as:

$$M(s_{i,j}, g_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} \quad (1)$$

where d is the distance between $s_{i,j}$ and g_i , and σ is the empirically estimated standard deviation of the GPS error.

The standard deviation is typically estimated from ground-truth annotated GPS data, in which the true road segment corresponding to each GPS position sample is known. The GPS error for a given GPS position is obtained by calculating the distance between it and its ground truth location. Several authors [7][9][12] have estimated the standard deviation using the median absolute deviation (MAD), which is a robust estimator. If the set of GPS error values corresponding to K GPS samples is given by $D = (d_k | k = 1, \dots, K)$, the standard deviation of the GPS error is estimated using MAD as:

$$\sigma = 1.4826 \text{ median}_k(d_k) \quad (2)$$

B. Transition Probability

In the HMM-based map-matching framework, the likelihood that the probe vehicle would move from a candidate state $s_{t,j}$ at time step t to another candidate state $s_{t+1,k}$ at time $t+1$ is represented by a transition probability. Since the candidate states represent locations on the road segments, transitions between them are dependent on the topology of the road network. Therefore, the transition probability for a pair of candidate states can be viewed as the likelihood of the best path between them being the true path taken by the probe vehicle. Two influential papers [7][10] relied on the idea that for true paths, the driving distance ("path length") tends to be close to the distance between the GPS positions ("GPS distance"). In other words, direct paths are considered more likely to be the true paths compared to circuitous paths. While different criteria could be used to compute the 'best' path using a standard routing algorithm such as the A* algorithm [13], we use the shortest-distance criteria.

Based on the above insight, Lou et al. [10] defined the transition probability simply as the ratio of the GPS distance to the path length. The transition probability from candidate state $s_{t,j}$ to candidate state $s_{t+1,k}$ is given by:

$$X(s_{t,j}, s_{t+1,k}) = \frac{D(g_t, g_{t+1})}{L(s_{t,j}, s_{t+1,k})} \quad (3)$$

where $D(g_t, g_{t+1})$ is the distance between the GPS observations g_t and g_{t+1} at time steps t and $t+1$ respectively, and $L(s_{t,j}, s_{t+1,k})$ is the length of the shortest path from candidate state $s_{t,j}$ to candidate state $s_{t+1,k}$.

In some methods [14][9], temporal factors such as the estimates of travel time and speed were used for calculating the transition probability. However, travel time and speed estimates are often unreliable due to variations in traffic conditions. We therefore did not rely on temporal factors. The transition probabilities in our baseline HMM framework are assigned as given in (3). In HMM implementations, the usual practice is to normalize the transition probabilities such that the probabilities of all the outgoing transitions from any state sum to 1. However, in the map-matching context, this produces an undesirable bias in favor of states with a low out-

degree [11][8]. Our implementation uses unnormalized transition probabilities.

C. Inferring the Most Likely State Sequence

The most likely sequence of hidden states in a HMM is commonly found using a dynamic programming method known as the Viterbi algorithm. The Viterbi algorithm finds the state sequence with the highest joint probability, which is the product of the emission and transition probabilities along the sequence. Given a GPS observation sequence g_1, \dots, g_T , for T time steps, the algorithm recursively calculates the maximum joint probability for each state at each time step as follows:

$$V_{1,k} = M(s_{1,k}, g_1) \quad (4)$$

$$V_{t,k} = M(s_{t,k}, g_t) \max_j (V_{t-1,j} X(s_{t-1,j}, s_{t,k})) \quad (5)$$

In the above equations, $V_{t,k}$ represents the joint probability of the most likely state sequence ending at state $s_{t,k}$, based on the GPS observations g_1, \dots, g_t . The functions M and X give the emission probability and transition probability respectively as defined in (1) and (3). The index j used to calculate $V_{t,k}$ in (5) points to the predecessor state $s_{t-1,j}$ of the state $s_{t,k}$ in the most likely sequence ending at the latter based on GPS observations up to time step t . After all the T time steps are processed as described above, the index x of the state $s_{T,x}$ that has the highest joint probability at the final time step T is determined as:

$$x = \arg \max_i (V_{T,i}) \quad (6)$$

Starting from the state $s_{T,x}$, the predecessor states are recursively retrieved up to the first time step in order to obtain the most likely state sequence in the reverse order. The most likely route R^* of the probe vehicle is constructed by linking successive pairs of states in the most likely state sequence with the respective shortest paths between them. If there are T GPS observations and the maximum number of candidate states of one GPS observation is Q , the Viterbi algorithm has a runtime complexity of $O(TQ^2)$.

IV. ROUTE CHOICE BASED TRANSITION PROBABILITY

Given a pair of candidate states $s_{t,j}$ and $s_{t+1,k}$, the transition probability from the former to the latter represents the probability of the best path between them being the true path taken by the vehicle. It can also be viewed as a measure of a driver's preference for that path. For instance, in Section 3.2, the directness of a path is considered as the quality that determines the preference for it. More complex data-driven approaches that learn driver preferences from GPS traces have also been recently proposed [11]. In all the probabilistic map-matching methods known to us, the transition probability is assigned by evaluating the attributes of the candidate path in isolation without considering the wider context in which it is situated. However, a driver's preference for a route is known to depend on the alternative routes available for the given origin-destination pair and the problem can be framed as one of discrete route choice [15]. Considering the available alternatives to a given candidate path can help to better estimate its transition probability as shown in the following example.

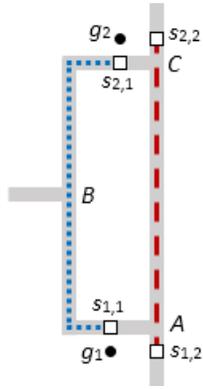


Figure 2. An example where a route choice based approach is beneficial

Figure 2 shows a part of a simplified road network with two GPS observations g_1 and g_2 . For clarity, it is assumed that the road segments AB , BC and AC have unidirectional traffic flowing in the bottom-to-top direction. The GPS observation g_1 is associated with two candidate states $s_{1,1}$ and $s_{1,2}$. Similarly, the GPS observation g_2 has two candidate states $s_{2,1}$ and $s_{2,2}$. If the attributes of the road segments are similar, the transition probability of the dotted-line path between $s_{1,1}$ and $s_{2,1}$ is likely to be close to that of the dashed-line path between $s_{1,2}$ and $s_{2,2}$. However, it is helpful to examine the wider context in which the states $s_{1,1}$ and $s_{2,1}$ are situated. If $s_{1,1}$ is the true location of the vehicle corresponding to g_1 , then it is reasonable to assume that the vehicle has come from the *preceding junction* A . Similarly, if $s_{2,1}$ is the true location of the vehicle corresponding to g_2 , then we can assume that the vehicle is headed towards the *succeeding junction* C . For a driver wanting to go from the preceding junction A to the succeeding junction C , the straight path along the road segment AC is the obvious preferred choice. While computing the transition probability for the dotted-line path between states $s_{1,1}$ and $s_{2,1}$, we should consider the probability that a driver would choose the path ABC over the best alternative path AC present in the context.

As it can be seen from the above example, two transitions that appear almost equally probable based on the conventional approach, may be associated with significantly different transition probabilities when viewed from a route-choice perspective. It is worth noting that the proposed approach obeys the Markov assumption that the transition probability between two states depend only on those states and not on the history before the current time step.

In the proposed method, each road segment in the road network is associated with a preceding junction and a succeeding junction. Here, 'junction' denotes any node in the road network, except those nodes with only one incoming and one outgoing road segments. For any road segment, the preceding (succeeding) junction can be obtained by tracing backward (forward) until a junction is reached. This process should be performed offline in order to avoid computation overhead during runtime.

The transition probability from state $s_{t,j}$ to state $s_{t+1,k}$ is assigned as follows:

- Let u be the preceding junction of $s_{t,j}$. Let v be the succeeding junction of $s_{t+1,k}$. Construct the path p from u to v that passes through $s_{t,j}$ and $s_{t+1,k}$. This is done by

concatenating the road segments linking u to $s_{t,j}$, the shortest path from $s_{t,j}$ to $s_{t+1,k}$, and the road segments linking $s_{t+1,k}$ to v .

- Determine the shortest path p^* between u and v .
- The probability that a driver would choose path p over the best alternative p^* is computed as follows:

Let $A = \{p, p^*\}$ be the set of alternatives or *choice set* available to the driver. Let $W(p)$ be the *utility* or benefit the driver obtains by choosing path p . In this work, we consider the inverse of a path's normalized length as its utility. i.e. The driver obtains more utility by travelling along shorter paths compared to longer ones. The proposed approach would work equally well if any other driver preference model is used.

We use the Binary Logit Model [15], according to which the probability of the driver choosing path p among the two options in the choice set A is given as:

$$P(p|A) = \frac{e^{\beta W(p)}}{e^{\beta W(p)} + e^{\beta W(p^*)}} \quad (7)$$

where β is a positive scale factor, which we empirically set as 2. Similarly, probability of the driver choosing path p^* is given as:

$$P(p^*|A) = \frac{e^{\beta W(p^*)}}{e^{\beta W(p)} + e^{\beta W(p^*)}} \quad (8)$$

We express the ratio of the probabilities given in (7) and (8) as the route choice probability for the pair of candidate states $s_{t,j}$ and $s_{t+1,k}$. It is given as:

$$C(s_{t,j}, s_{t+1,k}) = \frac{e^{\beta W(p)}}{e^{\beta W(p)} + e^{\beta W(p^*)}} \quad (9)$$

If the path being evaluated is the same as the best path in the context (i.e. $p = p^*$), then the route choice probability $C(s_{t,j}, s_{t+1,k})$ takes a value of 1.

- We define the new transition probability from state $s_{t,j}$ to state $s_{t+1,k}$ as:

$$X_0(s_{t,j}, s_{t+1,k}) = C(s_{t,j}, s_{t+1,k}) X(s_{t,j}, s_{t+1,k}) \quad (10)$$

where the function X denotes the conventionally assigned transition probability as given in (3).

A disadvantage of the proposed method is that it introduces additional shortest-path computations and increases the computational load. However, these shortest-path computations typically involve only short distances and can be efficiently solved by heuristically limiting the search to a small portion of the road network [16].

V. REAL-TIME VITERBI INFERENCE

The Viterbi algorithm outputs the most likely sequence of hidden states in a HMM only after all the observations are received and processed. However in many applications involving real-time streaming data, it is desirable to have an optimal or near-optimal partial solution before the entire observation sequence is received. This is especially so in the case of traffic-related applications that rely on real-time route inference from sparsely-sampled GPS data.

A modified form of the Viterbi algorithm, known as the online Viterbi algorithm [17][18], generates partial, optimal solutions without waiting for all the observations. This method, which has been applied for map matching in [9], can be briefly explained as follows. As seen in Section 3.3, for every state at time step t , the Viterbi algorithm stores a back pointer to a predecessor state at time $t-1$ which lies along the most likely sequence passing through the former. These back pointers help to construct the most likely state sequence. However, some of these back pointers can be considered as superfluous if they are not reachable (i.e. 'back-traceable') from any of states of a future time step. Figure 3 illustrates an example where the arrows denote back pointers, the shaded circles represent states that are back-traceable from at least one of the states of the latest time step t , and the unshaded circles denote states that cannot be back-traced. The online Viterbi algorithm keeps track of the number of back-traceable nodes of every time step. When it detects a time step with only one back-traceable state, it outputs the partial state sequence up to that time step. For instance, in Figure 3, the partial state sequence up to time step $t-2$ can be released because it is not possible for future GPS observations to alter the subsequence up to that point.

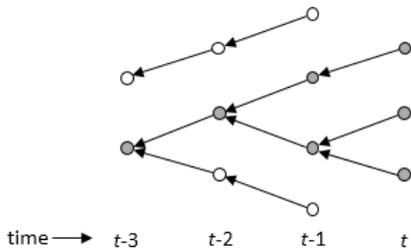


Figure 3. An example of states and back pointers in Viterbi inference

As the online Viterbi algorithm can produce a partial output only when it detects a time step with a single back-traceable state, it has a variable output latency. For instance, in the above example, releasing the subsequence up to time step $t-2$ at time step t corresponds to an output latency of 2 time steps. It is desirable to explore alternative methods that aim to reduce the output latency at the cost of optimality.

A naïve greedy way to reduce the output latency is to output the state with the highest joint probability at every time step. However, the state sequence obtained in this manner may have a high degree of error. We refer to this method, which has zero latency, as the 0-lag method. A common alternative approach is to wait for a fixed number of future GPS observations before releasing the partial output. In cases where two road segments look equally probable (e.g. a freeway and an exit ramp), waiting for the next GPS sample can help to clarify which of the two road segments the vehicle could have been on. We refer to this method with a fixed latency of k time steps as the k -lag method.

While future GPS observations can help to remove the ambiguity in some situations, it is unnecessary to wait for them in many other situations. Often, one state in the current time step would have a significantly higher joint probability compared to the others indicating that the vehicle is much more likely to be on the former. In such a case, intuitively it makes sense to output the solution up to the current time step without introducing any lag. Based on the above insight, we

propose a modification to the k -lag method, which we call as the heuristic k -lag method. We define a heuristic function $h(t)$ that gives the ratio of the highest joint probability among all the states of time step t to the second-highest joint probability:

$$h(t) = \frac{V_{t,best}}{V_{t,second}} \quad (11)$$

where *best* and *second* are the indices of the states with the highest and second-highest joint probabilities respectively. If $h(t)$ is greater than a threshold at the current time step t , the heuristic k -lag method outputs the partial state sequence up to t . Else, it outputs the partial state sequence up to the time step $t-k$. A threshold value in the range of 5 to 10 yields good results. In this work, the threshold is empirically set as 8.

VI. EXPERIMENTS AND RESULTS

A. The Experimental Setup

For testing the methods proposed in this paper, we used GPS data collected from roads in the central region of Singapore encompassing the Central Business District, a densely built-up area with urban canyon characteristics. The test area extended over a 11 km x 13 km grid. The GPS data was collected as part of a traffic survey conducted by a company providing traveler information services (see Acknowledgement). The data, totaling about 230 thousand GPS samples, was logged by taxis equipped with u-blox GPS receivers at the rate of 1 sample per second. It was organized into 137 GPS trajectories, with each trajectory having a duration of at least 15 minutes. The standard deviation of the GPS error was estimated to be 5.78 m using the MAD estimator as given in (2).

When high-frequency GPS data is available at the rate of 1 sample per second, the ground truth route taken by the vehicle can be determined very accurately using global map-matching algorithms. We implemented a map-matching method similar to [6] in order to programmatically determine the ground truth routes. They were subsequently verified through manual visual inspection. Low-frequency GPS trajectories, that serve as inputs for the methods described in this paper, were obtained by downsampling the original GPS trajectories at intervals ranging from 30 seconds to 3 minutes.

The HMM framework described in Section 3 serves as the baseline for evaluating the effectiveness of the improvement proposed in Section 4. The accuracy of a route inferred by a given map-matching method is estimated as:

$$\text{Accuracy} = \frac{\text{Total length of correctly identified road segments}}{\text{Total length of the ground truth route}}$$

B. Results

The accuracies of the baseline method for various sample intervals are shown in Figure 4. The accuracy gracefully degrades as the sampling interval is increased. The baseline results are on par with, if not better than, the best results reported in [6], where a comparison of several map-matching methods have been presented. When the transition probability definition is replaced with the new approach proposed in Section 4, a consistent increase in the accuracy is observed for all sampling intervals as shown in Figure 4.

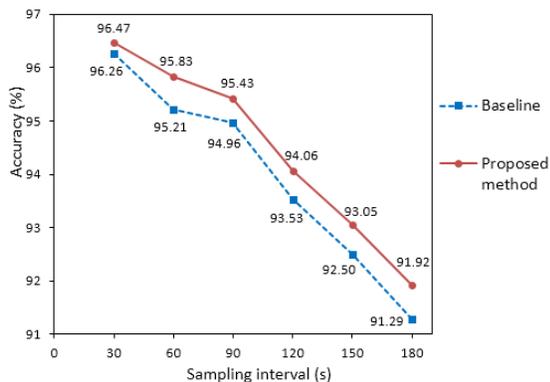


Figure 4. Accuracy of the baseline method and the proposed method

A comparison of the overall accuracy and mean output latency of the online Viterbi algorithm with the other methods described in Section 5 is shown in Figure 5. The greedy 0-lag method has substantially lower accuracy compared to the online Viterbi algorithm. In contrast, the accuracies of the 1-lag and the Heuristic 1-lag methods are only slightly lower than the online Viterbi algorithm. k -lag methods with $k > 1$ did not yield any noticeable improvement over the 1-lag method despite having higher output latencies. It can be seen from Figure 5 that the Heuristic 1-lag method has a significantly lower mean output latency while achieving almost the same level of accuracy as the 1-lag method.

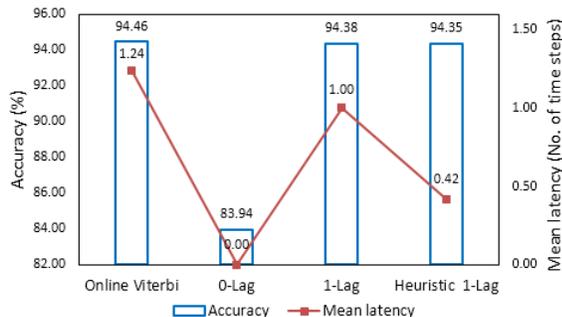


Figure 5. Comparison of accuracy and mean latency

VII. CONCLUSIONS

In this paper, we have presented a HMM-based framework for inferring routes travelled by probe vehicles by map matching sparse GPS data reported from them. We have proposed a new approach towards assigning transition probabilities in the HMM based on the idea that a better estimate of a driver's preference for a candidate transition path can be obtained by considering the alternative paths available in the context. We use a logit model for estimating the route choice probability of a transition path. Tests conducted with ground-truth annotated GPS data shows that the proposed approach provides a consistent improvement in the accuracy. We have also examined techniques commonly used for reducing the output latency of HMM-based map-matching methods and proposed a simple heuristic improvement. The proposed heuristic significantly reduces the mean output latency with negligible loss of accuracy.

There are a few areas that can be further investigated. It would be informative to examine the performance of the

proposed methods in the presence of large GPS errors. Another item for future work is to refine the route choice model. While we have used a route choice model that assumes a driver preference for least-distance paths, it would be interesting to follow a data-driven approach and estimate the route choice model from ground truth routes.

ACKNOWLEDGMENT

We thank Quantum Inventions Pvt. Ltd., Singapore for providing the GPS dataset used in this work.

REFERENCES

- [1] 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.
- [2] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.Z. Sun, "An interactive-voting based map matching algorithm", in *11th IEEE International Conference on Mobile Data Management (MDM)*, pp. 43–52, 2010.
- [3] 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.
- [4] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data", in *31st International Conference on Very Large Data Bases*, pp. 853–864, 2005.
- [5] Y. Zheng and M.A. Quddus, "Weight-based shortest-path aided map-matching algorithm for low-frequency positioning data", in *Transportation Research Board 90th Annual Meeting*, 2011.
- [6] 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.
- [7] P. Newson & 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*, pp. 336–343, 2009.
- [8] 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)*, 2009.
- [9] 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.
- [10] 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.
- [11] T. Hunter, P. Abbeel, and A. M. Bayen. "The path inference filter: model-based low-latency map matching of probe vehicle data", In *Algorithmic Foundations of Robotics X*, Springer Berlin Heidelberg, pp. 591–607, 2013.
- [12] Y. Wang, Y. Zhu, Z. He, Y. Yue, and Q. Li, "Challenges and opportunities in exploiting large-scale GPS probe data", HP Laboratories, Technical Report HPL-2011-109, 2011.
- [13] P.E. Hart, N.J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", *IEEE transactions on Systems Science and Cybernetics*, vol. 4(2), pp. 100–107, 1968.
- [14] J. Krumm, J. Letchner, and E. Horvitz, "Map Matching with Travel Time Constraints", in *Society of Automotive Engineers (SAE) World Congress*, 2007
- [15] M. Ben-Akiva, and M. Bierlaire, "Discrete choice methods and their applications to short-term travel decisions", *Handbook of Transportation Science*, Chapter 2, Kluwer Academic Publishers, 1999.
- [16] L. Fu, D. Sun, and L. R. Rilett, "Heuristic shortest path algorithms for transportation applications: state of the art" *Computers & Operations Research*, vol. 33(11), pp. 3324–3343, 2006.
- [17] R. Šrámek, "The on-line Viterbi algorithm", Master's thesis, Comenius University, Bratislava, 2007.
- [18] J. Bobbin, "An incremental Viterbi algorithm", in *6th IEEE International Conference on Machine Learning and Applications*, pp. 104–111, 2007.