Transportation (2018) 45:233–246
https://doi.org/10.1007/s11116-016-9734-2

A hybrid HMM model for travel path inference with sparse GPS samples

Erdem Ozdemir[1] · **Ahmet E. Topcu**[1] · **Mehmet Kemal Ozdemir**[2]

Published online: 29 September 2016

Abstract In this study, we propose a novel method for a travel path inference problem from sparse GPS trajectory data. This problem involves localization of GPS samples on a road network and reconstruction of the path that a driver might have been following from a low rate of sampled GPS observations. Particularly, we model travel path inference as an optimization problem in both the spatial and temporal domains and propose a novel hybrid hidden Markov model (HMM) that uses a uniform cost search (UCS)-like novel combinational algorithm. We provide the following improvements over the previous studies that use HMM-based methods: (1) for travel path inference between matched GPS positions, the proposed hybrid HMM algorithm evaluates all candidate paths to find the most likely path for both the temporal and spatial domains. In contrast, previous studies either create interpolated trajectories or connect matched GPS positions using the shortest path assumption, which might not be true, especially in urban road networks (Goh et al. 2012; Lou et al. 2009). (2) The proposed algorithm uses legal speed limits for the evaluation of discrepancy in the temporal domain as in Goh et al. (2012), and Lou et al. (2009) only if there is not sufficient historical average speed data; otherwise, we use historical average speed computed from data. Our experiments with real datasets show that our algorithm performs better than the state of the art VTrack algorithm (Thiagarajan et al. 2009), especially for cases where GPS data is sampled infrequently.

Keywords Traffic models · Map matching · Path inference · Route inference · Hidden Markov models

✉ Erdem Ozdemir
e.ozdemir@ybu.edu.tr

[1] Department of Computer Engineering, Yildirim Beyazit University, Ankara, Turkey

[2] Department of Electrical and Electronics Engineering, Istanbul Sehir University, Istanbul, Turkey

Introduction

Over the past years, GPS embedded handheld devices and on-car GPS systems have become popular. This increase has led to a rich collection of GPS samples, which has now opened up an opportunity for real-time delay estimation (Thiagarajan et al. 2009), route planning (Gonzalez et al. 2007), congestion point detection (Li et al. 2009), transportation mode detection (Schuessler and Axhausen 2009), and hot roads prediction (Thiagarajan et al. 2009). For all of these studies to perform well, the initial step is map matching, which is the correct alignment of GPS positions onto a road network and inference of the travel path that a vehicle is following using the knowledge of road networks and GPS samples (Lou et al. 2009; Yang et al. 2013).

The success of map matching and travel path inference algorithms requires high accuracy in GPS positions and sufficient numbers of GPS samples to be taken. Although GPS positions are considered to be accurate up to 5 m (Thiagarajan et al. 2009), there are several challenges in having GPS samples in high rates. First of all, sampling GPS data drains handheld devices' batteries quickly (Thiagarajan et al. 2010). Moreover, GPS does not work properly in urban canyon environments (Cui and Ge 2003). Shortage in certain roads and low number of GPS samples for different parts of roads are challenges for algorithms that try to infer travel paths that drivers follow. The issue stands out more especially in urban areas, where roads can be relatively short and vehicles can travel on many different road segments in short time intervals (Hunter et al. 2014).

In this study, we propose an effective algorithm that aims to reconstruct vehicles' travel routes even in GPS samples taken every 2 min, a duration considered to be a low sample rate in many studies (Chen and Bierlaire 2015; Hunter et al. 2014; Lou et al. 2009; Miwa et al. 2012; Thiagarajan et al. 2009). For instance, if a vehicle travels at a speed of 50 km/h, then the vehicle can travel about 1666 m between each GPS sample. In order to infer paths successfully at these low GPS sample rates, we consider the following points:

Global rather than local

Global optimization algorithms (Horst and Tuy 2013) process all GPS samples together to infer the most likely travel path given all input GPS samples (Hunter et al. 2014; Thiagarajan et al. 2009), local optimization algorithms generally construct vehicle path step by step with best choice at each step and the constructed vehicle path might not be the most likely one (Greenfeld 2002; Yu et al. 2006). Local optimization algorithms are in general computationally fast but their success for path inference shows a steep drop against a sparse number of GPS observations, especially in the case of incorrect alignments at the beginning of the local optimization algorithms. Global optimization algorithms tend to be computationally expensive, but errors in measurements are alleviated with the consideration of other GPS samples and previous decisions for travel path inference (Wenk et al. 2006). In this study, we adopt a global optimization algorithm that processes all the GPS samples in the query to infer the travel path.

Shorter reasonable paths

While inferring the travel path between two mapped GPS samples on a road network, one can consider the shortest path between the mapped GPS samples as the correct path, but this assumption is not likely to hold especially when the temporal distance between two

GPS samples is high or when the roads are in urban parts of a city. To overcome this issue, the vehicle's average speed for the shortest path between two GPS samples and the historical average speed, which is the average speed of other vehicles computed from data for the same shortest path, can be compared. If the discrepancy between historical average speed and the vehicle's average speed is high, one might consider other possible paths rather than the shortest path between the two GPS samples. To the best of our knowledge, other algorithms that combine spatial analysis only consider the shortest path (Chen and Bierlaire 2015; Dalumpines and Scott 2011; Lou et al. 2009; Oshyaniv et al. 2014; Schuessler and Axhausen 2009) for travel path inference problems.

In this study, we align GPS samples onto roads and infer paths between GPS samples together with temporal and spatial constraints. In a nutshell, we prefer paths according to their distance and fitness to historical speed data. Figure 1 and Table 1 illustrate the case. In Fig. 1 there are two GPS samples that are ordered with their timestamps from the earliest to the latest. There are two alternative paths connecting these two samples, as illustrated with solid and dashed lines. Without available time data, one would have preferred the dashed path. When historical average speed for the dashed line in Table 1 is

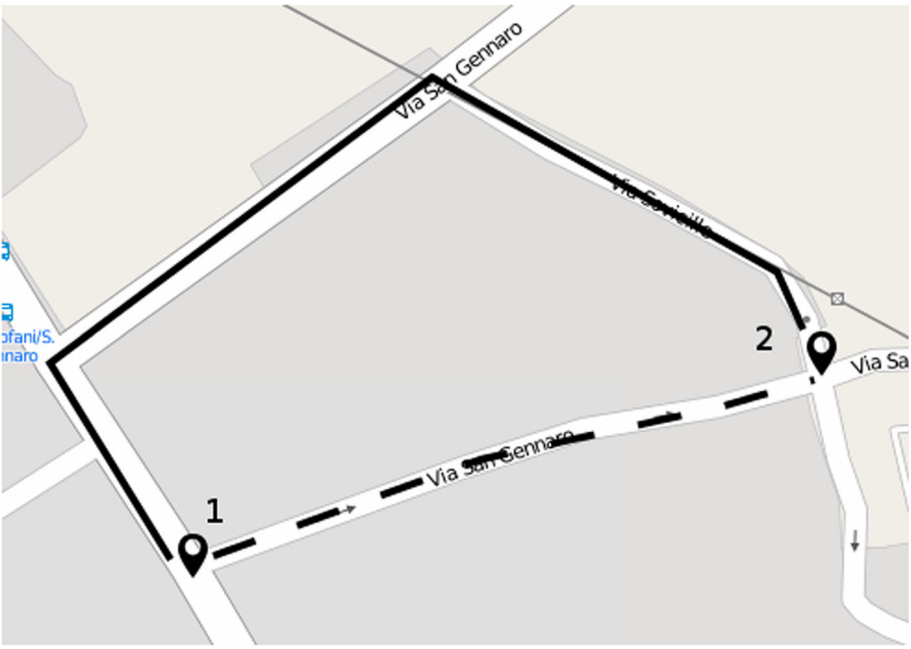


Fig. 1 Shorter paths rather than longer paths

Table 1 Vehicle's average speed and historical average speeds for solid and dashed lines

Dashed route		Solid route	
Vehicle's average speed (km/h)	Historical average speed (km/h)	Vehicle's average speed (km/h)	Historical average speed (km/h)
60	20	80	85

given, one would notice that if the dashed route were the correct one, the vehicle's average speed would show a significant discrepancy from historical average speed. Therefore, the solid line would be preferred over the dashed one.

Historical data for average speed on roads

Although historical average speed computed from data is commonly used in different traffic-related problems (Gonzalez et al. 2007; Li and McDonald 2002; Work et al. 2008), to the best of our knowledge, most of the path inference algorithms assume constant legal speed limits according to road types and transportation modes (Chen and Bierlaire 2015; Hunter et al. 2014; Lou et al. 2009; Yuan et al. 2010). In this study, we use average speed computed from historical speed data of a given road. For roads where we do not have sufficient historical data, we use legal speed limits of roads.

In this work, we propose a hybrid hidden Markov model (HMM)-based algorithm. Our algorithm searches for an optimal path according to a score function, which gives higher scores for paths that are spatially close to GPS samples and whose temporal discrepancies between the vehicle's average speed and historical average speed are low. We compared our algorithm with VTrack (Thiagarajan et al. 2009), which is a representative work among studies that use HMM (Goh et al. 2012; Lou et al. 2009; Thiagarajan et al. 2009). VTrack uses only positions of GPS samples and road network for travel path inference problem while our algorithm uses temporal domain as well. Our results indicate that our algorithm shows better performance in travel path inference problems than VTrack when sparse GPS samples are available.

Related work

Map matching problems date back to the 1990s (Quddus et al. 2007). The first algorithms matched GPS samples into shape points or edges in the road network using geometric information of GPS samples and shapes of edges (White et al. 2000). These algorithms are in the group of point-to-point matching and there are sets of data structures for fast searches for the closest edge given a GPS sample (Quddus et al. 2007). However, the closest edge or shape point is not necessarily the correct one, especially in urban road networks, which can be quite dense in small regions.

Later, topological map matching algorithms that use geometry, connectivity, and contiguity of links emerged (Greenfeld 2002; White et al. 2000; Yu et al. 2006). These algorithms improve the performance of the geometrical algorithms by adding additional topological information, but still they are mostly greedy algorithms and their performance drops in the event of measurement errors or high variance in spatial accuracy of GPS samples.

Recent studies on map matching focus not only on localization of GPS samples on a map but also on inference of the travel path (Bierlaire et al. 2013; Chen and Bierlaire 2015; Hunter et al. 2014; Lou et al. 2009; Miwa et al. 2012; Schuessler and Axhausen 2009; Thiagarajan et al. 2009). Some of these studies focus on travel routes with sparse GPS samples (Chen and Bierlaire 2015; Hunter et al. 2014; Lou et al. 2009; Miwa et al. 2012; Thiagarajan et al. 2009). In Aly and Youssef (2015), Goh et al. (2012), Lou et al. (2009) and Thiagarajan et al. (2009), HMM-based algorithms are used to model roads as hidden states and GPS samples as observations from hidden states. For cases when there

can be alternative paths between observations, VTrack (Thiagarajan et al. 2009) interpolates new virtual GPS samples in equal intervals over the line that connects real GPS samples. However, linear interpolation of virtual samples is not always a correct assumption and can lead to problems when the number of virtual GPS samples outweighs the number of real GPS samples. Another approach for extracting routes between observations is to compute shortest paths between observations (Bierlaire et al. 2013; Chen and Bierlaire 2015; Goh et al. 2012; Lou et al. 2009). Shortest paths may not always be the right choice, especially in urban networks (Hunter et al. 2014), as explained in Fig. 1 and Table 1.

Recently, Aly and Youssef (2015), Bierlaire et al. (2013) and Chen and Bierlaire (2015) use data from other sensors of mobile phones such as bluetooth, acceleration, etc. together with GPS data to reduce uncertainty in map matching. In Yuan et al. (2010), the authors use both spatial and temporal information of GPS samples and the road network and derive a voting-based algorithm where samples influence each other based on their distances. There are also path inference algorithms that derive admissible paths and filter paths based on probabilities assigned to those roads based on training data (Bierlaire et al. 2013; Chen and Bierlaire 2015; Hunter et al. 2014). Generation of admissible paths can grow exponentially. HMM-based algorithms use the Viterbi algorithm to quickly find the most likely paths by utilizing the best subsequences for each GPS sample in the search problem.

In contrast to previous studies, we don't have interpolated virtual samples and shortest path assumptions to connect two matched GPS samples, and we consider historical average speed rather than legal speed limits for temporal domain analysis.

Methodology

Problem statement

A GPS trajectory is an ordered set of observations (samples) $O = \{O_1 \rightarrow O_2 \rightarrow \cdots \rightarrow O_N\}$ where each $O_i \in O$ is a $\{lat, lon, time\}$ tuple with latitude, longitude, and timestamp of the sample. The samples in O are ordered by their time such that O_i precedes O_j if and only if $O_i.time <= O_j.time$. A road network can be modeled with a graph structure $G = (E, V)$ where E is the set of edges in graph G and the edges model road segments in the road network. Each edge E_k has a start vertex $E_k.start$ and end vertex $E_k.end$. V is the set of vertices that are used to model intersections of road segments in the road network. Each vertex V_l has (x, y) coordinates as attributes.

A path (route) R in graph G is an ordered set of edges $(e_1, e_2, \ldots, e_{|R|}) \in E \times E \times \cdots \times E$ where each edge is connected to its preceding edge such as $e_k.end = e_{(k+1)}.start$ and $|R|$ is the number of edges in R.

We consider the travel path inference as an optimization problem to find the optimal path R according to a score function F given a trajectory O and a road network G as given in the following expression (1):

$$max_R F(R, O, G) \tag{1}$$

F function

The score function F should ideally give higher scores for paths that are likely to be the driving route. In this study, we model the problem as a derivation of the HMM (Blunsom 2004) in which roads are modeled as hidden states while GPS samples are observations over the roads. We use emission probability (measurement equation) to score how likely an observation is to be sampled from a road, while transition probability can score how likely the transition between two edges is. In typical applications of the HMM to map matching and travel path inference, states are assumed to be edges and the transition probability between two edges is greater than zero if they are directly connected (Thiagarajan et al. 2009). In this study, we also assume states are edges, but when two edges are not directly connected, we generate possible paths between those edges to compute transition probability.

Emission probability P(O|E)

Emission probability gives how likely a sample is to be generated from an edge. A common choice is normal distribution $\sim\mathcal{N}(\mu, \sigma^2)$ for GPS observation O's distance to its projection on edge E (Lou et al. 2009; Thiagarajan et al. 2009).

Transition probability $P(E_i \rightarrow E_j)$

After the assignment of two observations into edges E_i and E_j, there can be different possible paths that connect E_i to E_j. To compute transition probability from $E_i \rightarrow E_j$, we search for a path that maximizes an exponential function $\exp^{(-K)}$ where K is the product of paths' length and discrepancy between the vehicle's average speed and historical average speed for that path. To formulate K's computation in a general problem setting, we assume a driver's path R that is composed of subpaths $R_{(ij)}$, which is just a path that connects road segments E_i to E_j within R. Note that E_i and E_j are edges to which observations O_i and O_j are assigned. We further define the $HAS(e)$ function for edges $e \in E$, which gives historical average speed for an edge e, and $A_s(R_{(ij)})$ is the average speed of the vehicle within $R_{(ij)}$ defined with in (2).

$$A_s(R_{ij}) = \frac{\sum_{e \in R_{(ij)}}}{O_j.time - O_i.time} \tag{2}$$

Spatial distance $SD(R_{(ij)})$ for subpath $R_{(ij)}$ is simply given in (3) as:

$$SD(R_{ij}) = \sum_{e \in R_{(ij)}} l(e) \tag{3}$$

l is a function that returns the length of an edge in meters.

To compute temporal distance $TD(R_{ij})$, we create average speed vector $[A_s(R_{(ij)}), \ldots, A_s(R_{(ij)})]$ and historical average speed vector $[HAS(e_1), \ldots, HAS(e_n)]$ whose elements are historical average speeds for each edge within R_{ij}, similar to Lou et al. (2009). The temporal distance between these two vectors can be calculated using cosine distance. Note that the cosine distance will be less than one, and hence multiplication of temporal and spatial distance would be less than just spatial distance. This situation is undesirable when searching for the best path in combinational-based algorithms. Therefore, we add one to the cosine distance, which is one minus cosine similarity, and our temporal distance turns out to be (4):

$$TD(R_{ij}) = 2 - \cos\left(\frac{\sum_{e\in R_{(ij)}} A_s(R_{(ij)}) \cdot HAS(e)}{\sqrt{\sum_{e\in R_{(ij)}} A_s(R_{(ij)})^2} \cdot \sqrt{\sum_{e\in R_{(ij)}} HAS(e)^2}}\right) \tag{4}$$

Here, K is just a multiplication of spatial distance and temporal distance given in (5) as:

$$K = \left(\sum_{e\in R_{(ij)}} l(e)\right) \cdot \left(2 - \cos\left(\frac{\sum_{e\in R_{(ij)}} A_s(R_{(ij)}) \cdot HAS(e)}{\sqrt{\sum_{e\in P_{(ij)}} A_s(R_{(ij)})^2} \cdot \sqrt{\sum_{e\in R_{(ij)}} HAS(e)^2}}\right)\right) \tag{5}$$

To write the final F function, we also define another assignment function $\alpha(o) : o \in O \rightarrow e \in R$ that assigns observations to edges. This function assigns each GPS observation O_i to some edge in route (path) R. There are some restrictions for the assignment function, such as that the first observation should be assigned to the first edge in R and the last observation should be assigned to the last edge in R. Additionally, observations and assigned edges should have the same order, i.e. $\alpha(O_i)$ precedes edge $\alpha(O_j)$ in route R if and only if O_i precedes O_j in trajectory O.

By multiplying emission and transition scores and using the α function, we can write the F function as follows:

$$F(R, O, G) = max_{\alpha} P(e_1|O_1) \cdot \prod_{i=2}^{|T|} P(O_i|\alpha(O_i)) \cdot P(\alpha(O_{(i-1)}) \rightarrow \alpha(O_i)) \tag{6}$$

Finding R that maximizes F

Finding R that maximizes F can be done by the Viterbi algorithm, which uses the dynamic programming technique. However, in real map datasets, there would be thousands of edges or hidden states, and the Viterbi algorithm would need to fill a table of size $|E| \times |O|$ (Blunsom 2004). Since GPS samples are generally considered to be accurate up to some distances, consideration of all edges in a map would be very impractical. In this study, similar to data relevance concept in Bierlaire et al. (2013), we assume that an observation would be sampled from one of the N nearest edges. Having a constraint to consider the N nearest edges for each observation, the Viterbi algorithm would need to fill a table of size $N \times |O|$, where N is significantly less than $|E|$ for real datasets.

The Viterbi algorithm starts by assigning the first observation into candidate edges and then computes their scores. For a new observation, it extends previous scores by multiplying the transition score of a new observation's candidate edges from previous edge and emission scores for each candidate edge. This process iterates until the last observation. A backtracking algorithm is then used to find the most likely edge sequence.

To compute transition scores, we assume that two successive observations are assigned to two edges E_i and E_j. There can be many possible paths between edges E_i and E_j and the goal is to find the path that gives the maximum transition score among all possible paths as transition scores. To solve this problem, we propose an algorithm similar to uniform cost search (UCS) that starts the search from the source edge E_i and gradually expands highest scored paths until it reaches the target edge E_j (Verwer et al. 1989). The pseudocode of the algorithm is given as follows:

The algorithm above uses a node structure that has a parent, an edge, and distance attributes. This node structure is used to represent paths. The algorithm starts with the

Require: E_i as source edge

Require: E_j as target edge

Require: O_i as observation assigned to E_i

Require: O_j as observation assigned to E_j

create set $ExpandedEdges$ for set of expanded edges

create priority queue $Queue$ for edges to be expanded

create $Node$

$Node.edge \leftarrow E_i$

$distance \leftarrow$ distance from projection of O_i on E_i to $E_i.end$

$Queue \leftarrow Node$

while Queue is not empty **do**

$Candidate \leftarrow Queue.remove$

if $Candidate.edge$ eq E_j **then**

return $Candidate$

else if $ExpandedEdges$ does not contain $Candidate.edge$ **then**

for $Neighbor : neighbors(Candidate.edge)$ **do**

create $NeighborNode$

$NeighborNode.edge \leftarrow neighbor$

$NeighborNode.distance \leftarrow Candidate.distance + distance(Neighbor)$

$NeighborNode.parent \leftarrow Candidate$

if $NeighborNode.edge$ eq E_j **then**

$timeFactor \leftarrow computeTimeFactor(averageSpeed, neighborWithDistance)$

$NeighborNode.distance = NeighborNode.distance * timeFactor$

end if

$Queue \leftarrow NeighborNode$

end for

$expandedNodes \leftarrow candidate.node$

end if

end while

return R_{ij} subpath between E_i and E_j and its distance

source edge as the path and adds it to a priority queue. It then iteratively pops the path with the lowest distance and adds new paths by expanding the last popped node with its neighbors. Before pushing a new path into the priority queue, the algorithm checks whether the target edge is the last edge on the path, and if it is, the algorithm computes the deviation of the path from historical speed data and multiplies the speed deviation by the distance of the path. This process continues until the algorithm pops a path that ends on the target edge. This algorithm finds subpath $R_{(ij)}$ that is optimal according to expression (5), which reflects both the spatial and temporal domains.

Experiments

Dataset

We have used mobility traces of taxi cabs in Rome, Italy (Bracciale et al. 2014). The dataset contains GPS positions collected from 320 taxi drivers that work in the center of Rome over the course of two months. Samples are collected approximately every 15 s. For the road network, we extracted a road network for the center of Rome from OpenStreetMap (OpenStreetMap 2015) and ignored nodes that are not related to the road network. In the pruned map, there are about 100K ways and 500K shape points. Finally, we manually labeled 25 routes for 50 GPS samples from 5 drivers and used these 25 routes as our ground truth. Average time for routes is about 12.5 min. For historical speed data, we run our algorithm using all GPS samples for all drivers except the ones used in our test routes and compute the average speed for each road in the road network. For roads without historical speed data, we use speed limits on the road if they exist.

Experiment setup

For evaluating our algorithm with low frequency GPS samples, we reduced the number of GPS samples by only keeping a single GPS sample in every M GPS samples. In this experiment, we have used $M = 1, 2, 4, 8$ values to reduce the number of GPS samples and inferred routes using reduced trajectories. We then evaluate the success of our algorithms and other comparison algorithms with Jaccard similarity for each route and take the average as the overall success of the algorithm. Besides the M parameter, there are also other parameters such as μ and σ in computation of emission probability, and N as the number of candidate edges for each observation. Here, we selected $\mu = 0$, $N = 10$ and tested different values of $\sigma \in 1, 5, 10$.

Computational performance

We also compute average time to infer paths for our queries to show our algorithm's practical feasibility. Our code is implemented in Java and tests are performed on an Intel Core i5-3337U CPU @ 1.80GHz 4, 8 GB RAM HP laptop. Average time to infer travel paths for the hybrid HMM algorithm is measured as 1.27 s.

Comparison algorithms

We have compared our algorithm with VTrack, which, like our proposed algorithm is also based on the HMM, but it assumes at least one single GPS sample (observation) to be generated from each edge (state) on the path. To have enough GPS samples, VTrack creates interpolated virtual GPS samples for its algorithm to work. It also has outlier removal as a pre-processing step (Thiagarajan et al. 2009), but in our experiments, we have not implemented outlier removal since, in our case, test routes are manually labeled by using all GPS samples with high frequency and these are chosen to have no outliers.

Results

We present the performance of our algorithm for different M and σ values given in Table 2. The results show that the best performance is achieved when $\sigma = 10$ and $M = 1$. The results also indicate that when all GPS samples are used, our algorithm infers close routes (over 0.9 similarity) to the ones that are manually labeled. For the case where we decrease the number of GPS samples, the performance also drops, as expected. The algorithm achieves the best performance in general at $\sigma = 10$, which indicates that the standard deviation of GPS samples' accuracy is best modeled with 10 m. The results indicate that even with samples taken every 2 min (M = 8), we can achieve 0.82 performance, while it is 0.9 for samples taken every 1 min (M = 4).

The results for the VTrack algorithm are presented in Table 3. The results indicate that our algorithm performs better than VTrack for all different values of M. VTrack drops under 0.7 at $M = 8$. The under-performance of VTrack can be explained by VTrack's assumption of linearly interpolated GPS samples. We elaborate on the situation with an example set of GPS samples for the VTrack algorithm in Table 4. In figures a and b in Table 4, we show a set of high frequency GPS samples and the manually labeled path from these samples. In figure c in Table 4, we illustrate sparse GPS samples when only one GPS sample is kept for every $M = 4$ samples. Figure d in Table 4 shows interpolated GPS samples between observed GPS samples for the VTrack algorithm. Figure e in Table 4

Table 2 Average Jaccard similarity of the inferred paths and manually labeled routes for the hybrid HMM algorithm

	$\sigma = 1$	$\sigma = 5$	$\sigma = 10$
M = 1	0.93	0.97	0.99
M = 2	0.89	0.91	0.91
M = 4	0.84	0.87	0.90
M = 8	0.79	0.81	0.82

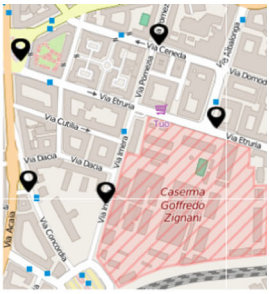
Table 3 Average Jaccard similarity of the inferred paths and manually labeled routes for the VTrack algorithm

M = 1	0.85
M = 2	0.81
M = 4	0.74
M = 8	0.69

Table 4 A table showing a set of figures for illustration of the VTrack and hybrid HMM algorithms

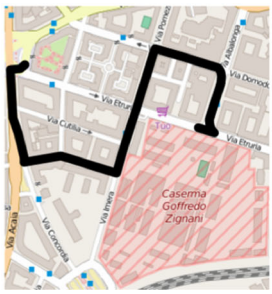
(a) High Frequency GPS Samples (M = 1)

(b) Labeled Path

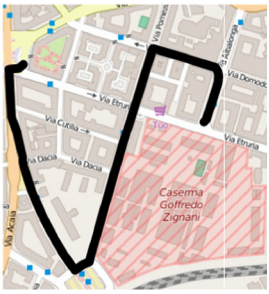


(c) Sparse GPS Samples (M = 4)

(d) Interpolated GPS Samples



(e) Path by VTrack



(f) Path by Hybrid HMM

shows the path inferred by VTrack considering both virtual and original GPS samples by just considering the spatial domain. Our algorithm, which does not assume the shortest path, is the correct one. It explores different paths and their scores with respect to both the spatial and the temporal domain, and it selects the correct path instead of the spatially closest path, as seen in figure f in Table 4.

We also performed another analysis that uses only speed limits instead of historical speed data with the hybrid HMM algorithm. The results are listed in Table 5. When we compare results of the hybrid HMM using historical average speeds in Table 2, we see that for most of the different σ and M, historical average speed shows better performance than

Table 5 Average Jaccard similarity of the inferred paths and manually labeled routes for the hybrid HMM algorithm using only speed limits

	$\sigma = 1$	$\sigma = 5$	$\sigma = 10$
M = 1	0.91	0.95	0.97
M = 2	0.86	0.91	0.91
M = 4	0.84	0.87	0.89
M = 8	0.77	0.78	0.79

just using speed limits. The results can be explained by the fact that speed limits do not reflect real average speeds on roads. Especially in dense urban parts of the city, the real average speed might be lower than speed limits, while the opposite might be true on highways. We also computed the percentage of roads for which we had historical average speed as 71% in our experiments; for the rest, we used speed limits. This indicates that using high frequency GPS samples to infer travel paths for other drivers and infer historical average speeds can improve the results when we encounter scarcity of GPS samples in travel path inference problems.

Conclusion

In this work, we propose a novel travel path inference method that maps GPS samples onto a road map and also infers routes with a low rate of sampled GPS data. Our results show that our algorithm can be used for path inference with high frequency GPS samples as well as for cases with low frequency GPS samples. Our algorithm is novel because it handles the assignment of GPS samples and inference of routes between assigned GPS samples in a new way as a single optimization problem, using both temporal and spatial data. The optimization problem can also be expanded with other factors such as weather, time of day, road work, etc. if relevant data is available.

For practical usage, collecting GPS data with handheld mobile devices is energy-consuming and algorithms that work on sparse GPS samples are more likely to be adopted for applications that target large sets of people. As a future direction, we are also aiming to show that our model could work with other datasets that are typically sparse such as WiFi traces or GSM/cell tower data and develop algorithms that will work on distributed systems to achieve scalability and high throughput in real time. With high throughput, we can work on tools that can learn massive numbers of people's driving patterns from inferred paths or can conduct further analysis for anomalies in traffic such as car accident, or for better route recommendation systems.

Additionally, road networks from real datasets are quite complicated and it might be possible to simplify road networks without significant loss in performance. Simplification of large road networks is still a challenging problem to be solved.

Acknowledgments Funding was provided by Türkiye Bilimsel ve Teknolojik Araştirma Kurumu (#113C037).

References

Aly, H., Youssef, M.: Semmatch: road semantics-based accurate map matching for challenging positioning data. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 5. ACM (2015)

Bierlaire, M., Chen, J., Newman, J.: A probabilistic map matching method for smartphone GPS data. Transp. Res. Part C Emerg. Technol. **26**, 78–98 (2013)

Blunsom, P.: Hidden Markov models. Lect. Notes **15**, 18–19 (2004)

Bracciale, L., Bonola, M., Loreti, P., Bianchi, G., Amici, R., Rabuffi, A.: CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from http://crawdad.org/roma/taxi/20140717/taxicabs, July 2014. traceset: taxicabs

Chen, J., Bierlaire, M.: Probabilistic multimodal map matching with rich smartphone data. J. Intell. Transp. Syst. **19**(2), 134–148 (2015)

Cui, Y., Ge, S.S.: Autonomous vehicle positioning with GPS in urban canyon environments. IEEE Trans. Robot. Autom. **19**(1), 15–25 (2003)
Dalumpines, R., Scott, D.M.: GIS-based map-matching: development and demonstration of a postprocessing map-matching algorithm for transportation research. In: Advancing geoinformation science for a changing world, pp. 101–120. Springer (2011)
Goh, C.Y., Dauwels, J., Mitrovic, N., Asif, M.T., Oran, A., Jaillet, P.: Online map-matching based on hidden Markov model for real-time traffic sensing applications. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 776–781. IEEE (2012)
Gonzalez, H., Han, J., Li, X., Myslinska, M., Sondag, J.P.: Adaptive fastest path computation on a road network: a traffic mining approach. In: Proceedings of the 33rd international conference on Very large data bases, pp. 794–805. VLDB Endowment (2007)
Greenfeld, J.S.: Matching GPS observations to locations on a digital map. In: Transportation Research Board 81st Annual Meeting (2002)
Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer Science and Business Media, Berlin (2013)
Hunter, T., Abbeel, P., Bayen, A.: The path inference filter: model-based low-latency map matching of probe vehicle data. IEEE Trans. Intell. Transp. Syst. **15**(2), 507–529 (2014)
Li, M., Zhang, Y., Wang, W.: Analysis of congestion points based on probe car data. In: 12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC'09, pp. 1–5. IEEE (2009)
Li, Y., McDonald, M.: Link travel time estimation using single GPS equipped probe vehicle. In: The IEEE 5th International Conference on Intelligent Transportation Systems, 2002. Proceedings, pp. 932–937. IEEE (2002)
Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., Huang, Y.: Map-matching for low-sampling-rate GPS trajectories. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 352–361. ACM (2009)
Miwa, T., Kiuchi, D., Yamamoto, T., Morikawa, T.: Development of map matching algorithm for low frequency probe data. Transp. Res. Part C Emerg. Technol. **22**, 132–145 (2012)
OpenStreetMap. OpenStreetMap openstreetmap. (2015). https://www.openstreetmap.org. Accessed 5 Sep 2015
Oshyaniv, M.F., Sundberg, M., Karlström, A.: Consistently estimating link speed using sparse GPS data with measured errors. Proc. Soc. Behav. Sci. **111**, 829–838 (2014)
Quddus, M.A., Ochieng, W.Y., Noland, R.B.: Current map-matching algorithms for transport applications: state-of-the art and future research directions. Transp. Res. Part C Emerg. Technol. **15**(5), 312–328 (2007)
Schuessler, N., Axhausen, K.: Processing raw data from global positioning systems without additional information. Transp. Res. Rec. J. Transp. Res. Board **1**(2105), 28–36 (2009)
Schuessler, N., Axhausen, K.W.: Map-matching of GPS traces on high-resolution navigation networks using the multiple hypothesis technique (MHT). Arbeitsberichte Verkehrsund Raumplanung **568**, 1–22 (2009)
Thiagarajan, A., Biagioni, J., Gerlich, T., Eriksson, J.: Cooperative transit tracking using smart-phones. In: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, pp. 85–98. ACM (2010)
Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., Eriksson, J.: Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pp. 85–98. ACM (2009)
Verwer, B.J.H., Verbeek, P.W., Dekker, S.T.: An efficient uniform cost algorithm applied to distance transforms. IEEE Trans. Pattern Anal. Mach. Intell. **11**(4), 425–429 (1989)
Wenk, C., Salas, R., Pfoser, D.: Addressing the need for map-matching speed: localizing global curve-matching algorithms. In: 18th International Conference on Scientific and Statistical Database Management, 2006, pp. 379–388. IEEE (2006)
White, C.E., Bernstein, D., Kornhauser, A.L.: Some map matching algorithms for personal navigation assistants. Transp. Res. Part C Emerg. Technol. **8**(1), 91–108 (2000)
Work, D.B., Tossavainen, O.-P., Blandin, S., Bayen, A.M., Iwuchukwu, T., Tracton, K.: An ensemble kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In: 47th IEEE Conference on Decision and Control, 2008. CDC 2008, pp. 5062–5068. IEEE (2008)
Yang, H., Cheng, S., Jiang, H., An, S.: An enhanced weight-based topological map matching algorithm for intricate urban road network. Proc. Soc. Behav. Sci. **96**, 1670–1678 (2013)
Yu, M.: Improved positioning of land vehicle in ITS using digital map and other accessory information. PhD thesis, The Hong Kong Polytechnic University (2006)

Yuan, J., Zheng, Y., Zhang, C., Xie, X., Sun, G.-Z.: An interactive-voting based map matching algorithm. In: Proceedings of the 2010 Eleventh International Conference on Mobile Data Management, pp. 43–52. IEEE Computer Society (2010)

Erdem Ozdemir received the B.S. and M.S. degrees in computer engineering from Bilkent University, Turkey, in 2008 and 2011, respectively. He is currently a Ph.D. student under the supervision of Dr. Ahmet Ercan Topcu in the Department of Computer Engineering at Yildirim Beyazit University, Ankara. His research interests include the use of probabilistic methods on complicated problems.

Ahmet E. Topcu received his B.S. degree Electrical and Electronics Engineering from the Middle East Technical University, Ankara, Turkey, in 1997. He completed M.S. degree from the Syracuse University, Syracuse, New York, US in 2001, and Ph.D. degree from the Indiana University, Bloomington, Indiana, US. In 2011, he joined the Department of Computer Engineering, Yıldırım Beyazıt University, Ankara, Turkey as an Assistant Professor. His current research interests include cloud computing, and big data analytics. He worked as a researcher at Louisiana State University, Baton Rouge, Louisiana, US for a year after completion his Ph.D. degree. He is also reviewer at SCI indexed international journals, committee member and referee for the grant projects.

Dr. Ozdemir completed his B.Sc and M.Sc in electrical engineering at METU, Ankara Turkey in 1996 and 1998 in, respectively. He received his Ph.D. from Syracuse University, Syracuse, USA in 2005 from electrical engineering. Between 1999–2012 he has worked in the area of broadband communication with the focus on CATV systems and 4G wireless systems. He is currently with Istanbul Sehir University, Istanbul, Turkey as an Asst. Prof at the EE department. His research interests are traffic density estimation via wireless devices, receiver algorithms for OFDM based systems, 5G receiver design, and FM band directional channel modeling.