Map Matching with Inverse Reinforcement Learning

Takayuki Osogami and **Rudy Raymond**
IBM Research - Tokyo
5-6-52 Toyosu, Koto-ku, Tokyo, Japan
{osogami,raymond}@jp.ibm.com

Abstract

We study map-matching, the problem of estimating the route that is traveled by a vehicle, where the points observed with the Global Positioning System are available. A state-of-the-art approach for this problem is a Hidden Markov Model (HMM). We propose a particular transition probability between latent road segments by the use of the number of turns in addition to the travel distance between the latent road segments. We use inverse reinforcement learning to estimate the importance of the number of turns relative to the travel distance. This estimated importance is incorporated in the transition probability of the HMM. We show, through numerical experiments, that the error of map-matching can be reduced substantially with the proposed transition probability.

1 Introduction

It is getting increasingly popular for cars to provide their trajectory data to receive services of high quality, including car navigation based on the latest traffic conditions. Telematics service providers try to extract as much value as possible from such trajectory data using techniques of artificial intelligence. Because the trajectory data is usually obtained with the Global Positioning System (GPS), it is sampled at discrete epochs and prone to measurement errors. The first step of getting knowledge from such trajectory data is *map-matching* that finds the most likely route given a sequence of GPS data points. This map-matching has turned out to be surprisingly difficult even with moderate sampling rate and moderate error [Krumm *et al.*, 2007; Newson and Krumm, 2009].

A difficulty of map-matching stems from the tradeoff between the route that visits the places close to the GPS points and the route that are likely to be traveled. Naively connecting the points on the roads that are respectively closest to given GPS data points often results in a winding route that is unlikely to be traveled. A state-of-the-art technique is based on a Hidden Markov Model (HMM), where a latent state corresponds to a point on a road segment, and an observed variable corresponds to a GPS data point. [Newson and Krumm, 2009] shows that the *error* in map-matching as defined in Section 3 can be made below 0.11 %. However, they show that the error can increase dramatically, when the noise or the sampling rate of GPS observation is only slightly increased. Unfortunately, the sampling interval in real GPS data is often not short enough for reliable map-matching with existing approaches. For example, the GPS data points in T-drive [Yuan *et al.*, 2010; 2011], a dataset from taxis in Beijing, are often sampled with intervals over two minutes.

A key design element of an HMM is the transition probability between latent states. We propose a way to define and estimate the transition probability so that the results of the map-matching are improved particularly when the GPS data points are sparse. In [Newson and Krumm, 2009], the transition probability from a latent state to another depends on the travel distance between the corresponding two latent points along road segments. Their transition probability decreases exponentially as the travel distance increases. We find that, when the GPS data points are sparse, map-matching with this transition probability can lead to a short but winding route that is unlikely to be traveled by real drivers. The distance-based map-matching is also found in [Eisner *et al.*, 2011].

We propose to define the transition probability by the use of the path constructed based on a convex combination of multiple metrics such as the travel distance and the number of turns. The optimal path that minimizes the convex combination of these metrics can be found with the standard Dijkstra algorithm. The cost of this optimal path is used to determine the transition probability. Notice that it is not novel to simply use multiple metrics in map-matching [Brakatsoulas *et al.*, 2005; Goh *et al.*, 2012]. What has not been fully understood with the prior work is how to use the multiple metrics such that one can find the optimal paths with respect to the cost defined with those metrics. For example, [Goh *et al.*, 2012] evaluate paths with multiple metrics but consider only those paths that are shortest between two road segments. Related work is further discussed in Section 4.

Figure 1 illustrates a typical case where the proposed approach is effective. There are four GPS data points, $p_1, \ldots, p_4$. The task of map-matching is to infer the route that is taken by the vehicle from which those GPS data points are observed. A limitation of the existing approach of defining the transition probability based solely on the shortest path stems from the fact that the shortest path is not necessarily the path that is most likely to be traveled. In Figure 1, there are

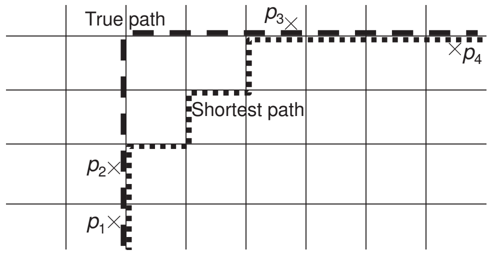


Figure 1: The necessity of taking into account the number of turns in map-matching. The dashed line shows the true (and shortest) path, the dotted line show another shortest (but winding) path, and four crosses show GPS data points.

many shortest paths from the latent road segment corresponding to p_2 to that corresponding to p_3. Although all of these paths have the shortest distance, some of the paths are more "natural" than the others. For example, one of the shortest paths shown in Figure 1 makes five turns, while there is an equally short path that makes only one turn.

The key question that needs to be addressed when we use multiple metrics is which metrics are more important than others and how much. For example, [Newson and Krumm, 2009] leave it future work to systematically estimate the parameters of their HMM. One potential approach is cross validation with grid search. When the traversed route is known for a limited sequence of GPS data, we can tune the weights on the multiple metrics in such a way that the accuracy of the map-matching is maximized for that limited sequence. It is also possible to recover the route for a limited sequence of GPS data reliably with existing approaches. Then this recovered route can be used as the true route for cross validation. The sampling interval of real GPS data can vary tremendously over time. For example, in T-drive [Yuan *et al.*, 2010; 2011], there are about the same GPS points with intervals less than one minute as those with more than five minutes. We could thus reliably recover the routes for the periods when the sampling intervals are sufficiently short, which can then be used to find the appropriate weights to recover the routes for the remaining periods.

However, cross validation has two limitations. First, finding good weights are not easy particularly when many metrics are considered. Second, we need a pair of GPS data and the corresponding true route. There are cases where traversed routes are available without the corresponding GPS data, for example, via interview with the driver of a vehicle, driving simulators or games, and travel plan. Without the corresponding GPS data, this information about the traversed routes is useless for cross validation.

To overcome the limitations of cross validation, we propose to estimate the weights of the multiple metrics by the use of inverse reinforcement learning (IRL), in particular Maximum Entropy IRL proposed in [Ziebart *et al.*, 2008]. The appropriate weights can then be found reliably by solving a convex optimization problem via a gradient method. Also, we only need traversed routes to find the appropriate weights.

The first contribution of this paper is the novel definition of the transition probability in the HMM for map-matching as well as the novel use of IRL to estimate the weights in this transition probability (See Section 2). Our second contribution is an empirical evidence that the quality of map-matching based on an HMM can be improved substantially by the use of the new transition probability (see Section 3). Specifically, using the benchmark provided by Newson and Krumm, we show that the state-of-the-art map-matching of [Newson and Krumm, 2009] can be made substantially more accurate by taking into account the number of turns in addition to the distance. The error can be reduced by more than 40 %.

2 Map-matching Algorithm

We start by describing the approach of map-matching with an HMM with a focus on the proposed transition probability (see Section 2.1). Our HMM is equivalent to that of [Newson and Krumm, 2009] except for the transition probability. The omitted details can thus be found in [Newson and Krumm, 2009]. In Section 2.2, we show how the weights in the transition probability is estimated by the use of IRL. Here, we follow the approach of Maximum Entropy IRL, whose details can be found in [Ziebart *et al.*, 2008].

2.1 Map-matching with a Hidden Markov Model

The key design elements of an HMM are the latent state, the observable variable, the emission probability, and the transition probability. A latent state of our HMM is a pair of a (latent) road segment and a (latent) point on the road segment, where a road segment is defined to be the directed road that connects an intersection to a neighboring intersection. An observable variable of our HMM is a GPS data point, specifically a longitude and a latitude.

The emission probability of a GPS data point from a latent point follows from a model of measurement error with the GPS. Specifically, let $\mathbf{z}$ be the location of the GPS data point, $\mathbf{x}$ be the location of the latent point, and σ be the standard deviation of the measurement error. Then the emission probability follows from the normal probability density function[1]:

$$\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{||\mathbf{z}-\mathbf{x}||_2}{2\sigma^2}\right). \tag{1}$$

The transition probability from a latent state, s, to another, s', is given by the value of the probability density function of an exponential distribution. Let $c(p)$ be the cost of traveling along a path, p, from the latent point corresponding to s to that corresponding to s', where the precise definition of the cost will be given in the following. For the purpose of map-matching with HMM, it suffices to consider only the path, $p^\star$, that minimizes $c(p)$, because $p^\star$ maximizes the transition probability. Let $c^\star$ be the minimum value of $c(p)$. Then the transition probability is given, with a parameter β, by

$$\frac{1}{\beta}\exp\left(-\frac{c^\star}{\beta}\right). \tag{2}$$

In [Newson and Krumm, 2009], $c^\star$ is defined to be the distance between the two latent points with the following normalization. Let $d^\star$ be the travel distance along $p^\star$, and d_0 be

[1]Throughout, we assume that the earth is flat for simplicity.

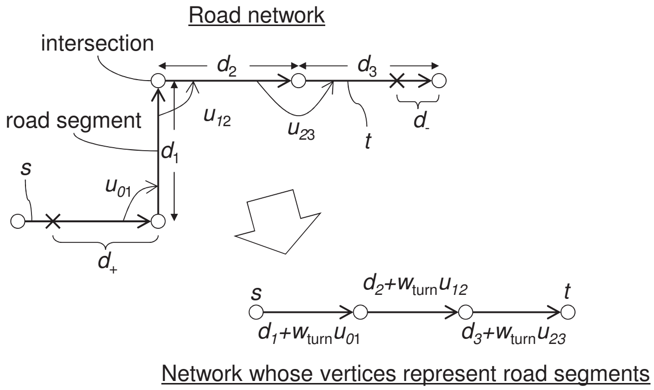


Figure 2: Constructing a network whose vertex represents a road segment.

the Euclidean distance between the two GPS points that correspond to s and s'. Then $c^\star = |d^\star - d_0|$. The motivation of this normalization stems from the empirical observation from the ground-truth-route: the probability that the path p is the correct route follows an exponential distribution with respect to $|d - d_0|$, where d is the travel distance along p [Newson and Krumm, 2009]. Namely, the intuition is that *the correct route is likely to be the route whose travel distance is close to the Euclidean distance between the corresponding GPS points*.

We extend this intuition to also consider turns: *the correct route is likely to be the route that makes turns in a manner similar to the routes that have been observed elsewhere*. We define the cost, $c^\star$, as the minimum value, $v^\star$, of the weighted sum of the travel distance and the cost of turns from the latent point of s to that of s' with the normalization as in [Newson and Krumm, 2009]. Specifically, our cost is $c^\star = |v^\star - d_0|$.

To calculate $v^\star$, we construct a network whose vertex represents a road segment (see Figure 2). There is an edge from a vertex, v, to another, v', if the road segment corresponding to v ends at the intersection where that corresponding to v' starts. The cost of an edge from v to v' is the weighted sum of the length, $d_{v'}$, of the latter road segment and the quantity, $u_{v\,v'}$, that represents the cost of the turn from the former road segment to the latter. For example, we can define

$$u_{v\,v'} = \begin{cases} 0 & \text{if } |\theta_{v\,v'}| < \pi/4, \\ 1 & \text{if } \pi/4 \le |\theta_{v\,v'}| \le 3\pi/4, \\ 2 & \text{if } 3\pi/4 < |\theta_{v\,v'}| < \pi, \\ 10 & \text{if } |\theta_{v\,v'}| = \pi. \end{cases} \tag{3}$$

where $\theta_{v\,v'}$ is the angle between the two road segments. Let $d_{v'}$ be the length of the latter road segment. Then the cost from v' to v is

$$d_{v'} + w_{\mathrm{turn}}\, u_{v\,v'}, \tag{4}$$

where w_{turn} is the weight of turn relative to travel distance.

Given an origin vertex, s, and a destination vertex, t, we can find the optimal path, from s to t, that minimizes the sum of the cost, using the Dijkstra algorithm. The total cost along the optimal path is the minimum value of the weighted sum for the following two quantities:

- the travel distance of a path from the end point of the road segment corresponding to s to that of t;
- the sum of the cost of turns over all pairs of neighboring road segments along that path.

We can then obtain the cost from a latent point to another by adjusting the cost from the corresponding latent road segment to another, taking into account the distance between a latent point and an end point of a latent road segment. For example, in Figure 2, two latent points are denoted with crosses. The cost from one latent point to the other is

$$v^\star := d_1 + d_2 + d_3 + d_+ - d_- + w_{\mathrm{turn}}\,(u_1 + u_2 + u_3).$$

With these design elements of an HMM, one can find the most likely sequence of latent road segments given the sequence of GPS data points, using the Viterbi algorithm [Bishop, 2007][2].

Notice that the cost (4) can include other metrics such as travel time and toll as in [Goh *et al.*, 2012; Krumm *et al.*, 2007]. The cost of turn as shown in (3) can itself be interpreted as the weighted sum of four types of turns: "straight" ($|\theta_{v\,v'}| < \pi/4$), "normal turn" ($\pi/4 \le |\theta_{v\,v'}| \le 3\pi/4$), "hard turn" ($3\pi/4 < |\theta_{v\,v'}| < \pi$), and "U-turn" ($|\theta_{v\,v'}| = \pi$). If the cost of turn is linear in the angle, $\theta_{v\,v'}$, one can simply use the angle as a metric of cost. What is essential here is that the cost must be a convex combination of multiple metrics so that we can use the Dijkstra algorithm to find the path that minimizes the cost. The convex combination also allows us to use IRL for inferring the weights for the multiple metrics, as we will see in Section 2.2.

2.2 Weight Estimation

We propose to use Maximum Entropy IRL of [Ziebart *et al.*, 2008] to infer the appropriate values of the weights in the cost (4). Here, it is assumed that we are given a set of routes that have actually been traveled. This set can consist of either a single route or multiple routes that have been traveled by either a single driver or multiple drivers. Ideally, those routes share common patterns with the routes traveled by the vehicles whose GPS dataset is to be used for map-matching.

Maximum Entropy IRL infers the values of the weights such that the likelihood of traveling given routes is maximized. The likelihood is defined based on the assumption that the probability of traveling the route with total cost, C, is proportional to $\exp(-C)$. In our case, the log-likelihood, $L(w)$, is convex with respect to the weights, w, and the optimal weights can be found via gradient methods (see [Ziebart *et al.*, 2008]). That is, w is iteratively updated as follows:

$$w := w + \alpha\,\nabla L(w),$$

where α determines the step size of the update.

In our context, $\nabla L(w)$ is given by the difference between two costs: $\nabla L(w) = f(w) - \tilde{f}(w)$. Here, $\tilde{f}(w)$ denotes the total cost of the given routes when the cost is defined with the weights, w. The value of $\tilde{f}(w)$ can thus be calculated in a straightforward manner. On the other hand, $f(w)$ denotes the sum of the expected total cost from the origin to the destination for each of the given routes, where the route from the

[2] Strictly speaking, the standard Viterbi algorithm does not allow the transition probability to depend on the GPS data points (specifically, d_0), which is discussed in [Raymond *et al.*, 2012].

origin to the destination is selected under the assumption that the route with total cost, C, is selected with the probability proportional to $\exp(-C)$.

The value of $f(w)$ can be calculated by the use of dynamic programming. Specifically, the expected total cost from an origin, $s_{\rm O}$, to a destination, $s_{\rm D}$, is calculated as follows. First, we recursively calculate the following equation

$$Z_{n+1}(s) = \begin{cases} \displaystyle\sum_{s' \in N(s)} e^{-c_{s,s'}} Z_n(s') & \text{if } s \in \mathcal{S} \setminus \{s_{\rm D}\} \\ 1 & \text{if } s = s_{\rm D}, \end{cases} \tag{5}$$

starting from $Z_0(s_{\rm D}) = 1$ and $Z_0(s) := 0$ for $s \in \mathcal{S} \setminus \{s_{\rm D}\}$, where $c_{s,s'}$ denotes the cost from s to s', $\mathcal{S}$ denotes the set of the road segments under consideration, and $N(s)$ denotes the set of road segments reachable from s. After a sufficiently large number, M, of recursion, we obtain the probability, $p(s, s')$, of traveling from a road segment, s, to another, s', by $p(s, s') = Z_{M-1}(s)/Z_M(s)$ for each pair of $s, s' \in \mathcal{S}$. We then recursively calculate the following equation

$$D_{n+1}(s) = \sum_{s' \in \{\hat{s} | s \in N(\hat{s})\}} D_n(s')\, p(s', s) \tag{6}$$

for $n = 1, \ldots, M$, starting from $D_0(s_{\rm O}) = 1$ and $D_0(s) = 0$ for $s \in \mathcal{S} \setminus \{s_{\rm O}\}$. The total expected cost from $s_{\rm O}$ to $s_{\rm D}$ is then obtained by

$$\sum_{s \in \mathcal{S}} \sum_{n=0}^{M} D_M(s)\, p(s, s')\, c_{s,s'}.$$

This total expected cost is calculated for each of the given routes and summed over all of the given routes to obtain $f(w)$. In our numerical experiments, we set M twice as large as the number of road segments that constitute the corresponding given route.

3 Experimental Results

We now study the effectiveness of the proposed transition probability. We use the benchmark set that is used in [Newson and Krumm, 2009] and made publicly available, where the GPS data points are collected for a route of approximately 80 km in Seattle, for which the true route is known.

We evaluate the quality of the results of map-match with *the route mismatched fraction* (error) as proposed in [Newson and Krumm, 2009]. Roughly speaking, the error of a route, R, found by map-matching is given by the fraction $(L_- + L_+)/L$, where L_- is the total length of the road segments that are in the true route but not in R, L_+ is the total length of the road segments that are not in the true route but in R, and L is the length of the true route. More precisely, the error takes into account the order of the traced road segments.

We also compute *precision* and *recall*. The precision of a route, R, found by map-matching refers to the ratio of the number of the road segments, along R, that are contained in the true routes against the total number of the road segments in R. The recall refers to that ratio but now against the total number of the road segments in the true route. These measures take into account neither the order of the road segments nor the length of the road segment, but their values give intuitive sense of the quality of map-matching.

Figure 3 evaluates the routes found with map-matching with respect to the three measures. Here, we use the cost defined by (3) and (4), and vary the value of $w_{\rm turn}$ in (4) as indicated along the horizontal axis. The purpose of this first set of experiments with varying $w_{\rm turn}$ is to isolate the impact of the use of multiple metrics in the transition probability of the HMM for map-matching from the quality of the estimated weights in the transition probability. The zigzag line shows the results with the proposed transition probability, and the red horizontal line shows the results with the transition probability in [Newson and Krumm, 2009]. Throughout we fix $\sigma = 10.0$ meters in (1) and $\beta = 0.1$ meters in (2), but qualitative findings hold for a range of values for these parameters.

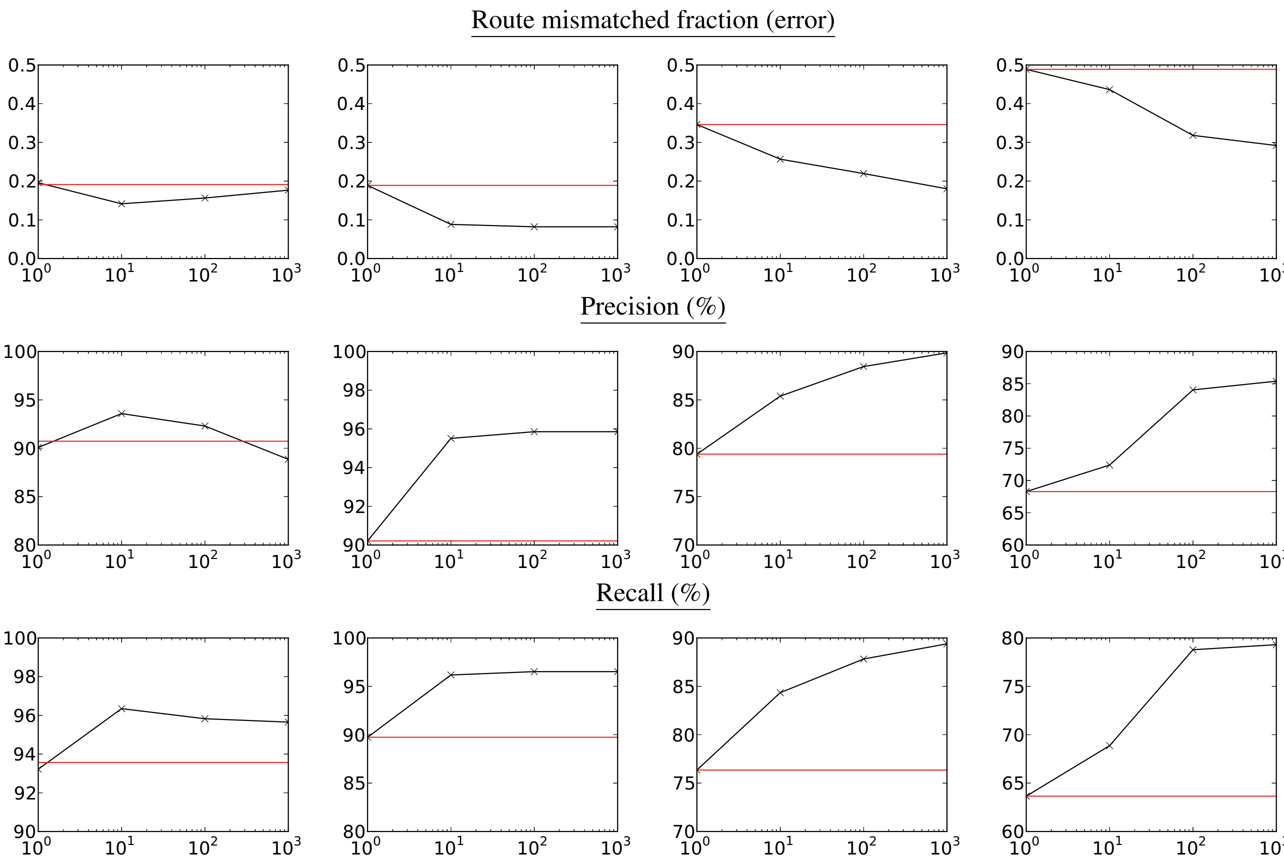
The top row shows the error (the smaller, the better) for four sampling rates of the GPS data points as indicated below the figures. For all sampling intervals from 30 seconds (Column (a)) to 240 seconds (Column (d)), the error can be reduced by the proposed transition probability by appropriately choosing $w_{\rm turn}$. The reduction is substantial when the sampling interval is long. When the sampling interval is 240 seconds, the error is reduced from 0.48 to 0.29, which is a 40 % reduction. There can however be a counter effect of the proposed transition probability. When the sampling interval is 30 seconds, the error is slightly increased (by 2.5 %) with $w_{\rm turn} = 1.0$. Also, we can observe that making $w_{\rm turn}$ too large can increase the error.

The middle row shows the precision and the bottom row shows the recall (the larger, the better). When the sampling interval is 30 seconds or 60 seconds, the precision and the recall are around 90 % with the transition probability in [Newson and Krumm, 2009], but these can be increased as high as 95 % by the use of the proposed transition probability. For longer sampling intervals, the effectiveness of the proposed transition probability is more substantial. The precision and the recall can be increased from 65-80 % to 80-90 %.

The remaining question is whether we can set the value of $w_{\rm turn}$ appropriately. We now evaluate the quality of our map-matching when $w_{\rm turn}$ is inferred by the use of Maximum Entropy IRL. In our second set of experiments, we divide the true route and the benchmark GPS data points into twelve datasets. Each dataset consists of the true route of 48 consecutive road segments and corresponding GPS data points. We use one of the twelve datasets to infer the value of $w_{\rm turn}$ by the use of Maximum Entropy IRL and apply our map-matching algorithm with the inferred $w_{\rm turn}$ to the remaining eleven datasets. This experiment is repeated twelve times, so that every dataset is once used as the training set for inferring the value of $w_{\rm turn}$. In these experiments, we use the following definition of $u_{v\,v'}$ instead of (3):

$$u_{v\,v'} = \begin{cases} 0 & \text{if } |\theta_{v\,v'}| < \pi/4, \\ 1 & \text{if } \pi/4 \le |\theta_{v\,v'}| \le 3\pi/4, \\ 2 & \text{if } 3\pi/4 < |\theta_{v\,v'}|. \end{cases}$$

Ideally, the relative weights such as 0, 1, and 2 should also be inferred with Maximum Entropy IRL, but the volume of our available datasets is not sufficiently large for reliably inferring these relative weights. We hence infer only $w_{\rm turn}$ and



(a) 30 seconds (b) 60 seconds (c) 120 seconds (d) 240 seconds

Figure 3: Error, precision, and recall are plotted against w_{turn} as indicated along the horizontal axis. The sampling intervals of the GPS data range from 30 seconds (a) to 240 seconds (d). The zigzag line represents the results with the proposed transition probability; the red horizontal line represents the results with the transition probability in [Newson and Krumm, 2009].

evaluate the effectiveness of the approach of the use of Maximum Entropy IRL in map-matching.

Figure 4 shows the results of this second set of experiments. Again, we use the route mismatched fraction (error), precision, and recall to evaluate the quality of the results of map-matching. However, we now show the average values of these metrics over the twelve experiments. More precisely, the values of the error are weighted by the length of the corresponding true route when we take their average values. Figure 4 plots the error (Column (a)), the precision (Column (b)), and the recall (Column (c)) against the sampling interval that range from 30 seconds to 600 seconds. The solid line shows the results with the proposed transition probability, where the values of w_{turn} are inferred by the use of Maximum Entropy IRL. The red dashed line show the corresponding results with the transition probability in [Newson and Krumm, 2009].

Observe that the error with the proposed approach is substantially lower than that with the approach of [Newson and Krumm, 2009] particularly when the sampling interval is long. When the sampling interval is 420 seconds, the error is reduced from 0.870 to 0.372, which is over 57 % reduction. The corresponding precision is increased from 52.9 % to 76.3 % (over 44 % up), and the recall is increased from 38.4 % to 69.8 % (over 81 % up).

The inferred values of w_{turn} ranges from 117.1 to 2,257 in our experiments. This means that the cost of a turn is equivalent to the length of 117.1 meters to 2,257 meters. In one of the twelve datasets, however, the inferred value of w_{turn} did not converge. Hence, for this dataset, we use the transition probability of [Newson and Krumm, 2009] instead of ours in the results shown in Figure 4.

In our experiments, the computation for HMM takes 0.65 seconds on average, and IRL takes 0.25 seconds per iteration on average on a single core of a Windows workstation with 2.6 GHz Xeon CPU. HMM is implemented with Java, IRL is implemented with Python. The number of iterations for IRL to converge depends on various conditions, including the initial value and the step size of steepest decent. Our particular settings require from a few iterations to a few thousand iterations. Notice that we only need to run IRL once to obtain the weights for a driver. Those weights can then be used for subsequent map-matching with HMM.

4 Related Work

An in-depth introduction of map-matching can be found in [Quddus *et al.*, 2007]. When there is a sufficient volume

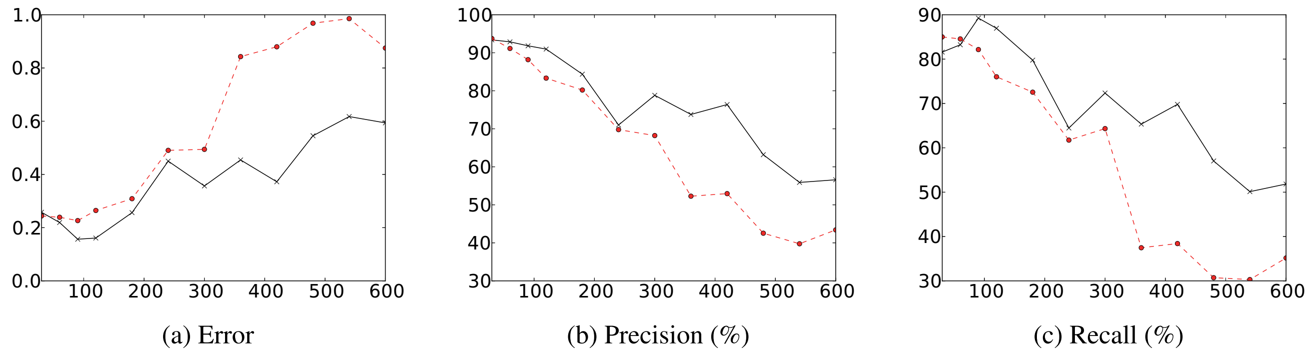


Figure 4: Error, precision, and recall are plotted against the sampling interval (seconds) as indicated along the horizontal axis. The solid lines represent the results with the proposed transition probability, where the values of w_{turn} are inferred with Maximum Entropy IRL; the red dashed lines represent the results with the transition probability in [Newson and Krumm, 2009].

of measured data, simple approaches that match GPS points to nearest roads are often satisfactory [Greenfeld, 2002; White *et al.*, 2000]. Otherwise, one can use either geometric approaches or probabilistic models, the HMM in particular.

The geometric approaches rely on topological similarities for finding the road sequences that best match the given sequence of GPS points. The topological similarities are, for example, measured with the Frèchet distance [Brakatsoulas *et al.*, 2005; Chen *et al.*, 2011] or path shape measures [Funke and Storandt, 2011]. [Brakatsoulas *et al.*, 2005] also uses the measure that takes into account the angle between a directed road segment and the direction from a GPS point to another. This measure is however different from the turn, or the angle between two consecutive road segments, that we consider in this paper. Geometric approaches are generally not suitable for the GPS points that are sampled with a long interval, because it is hard to find the right similarly measure between a straight line connecting two distant points with a sequence of many road segments connecting them. Because there are no GPS points in between those two points, there will be many ambiguous sequences that are similar under the measure. In addition, geometric approaches are more computationally expensive than the one with probabilistic models.

The HMM was first used for map-matching applications in [Lamb and Thiebaux, 1999] by combining it with the Kalman filter. The idea of leveraging transition probabilities of the HMM to eliminate moving between unconnected roads was introduced in [Hummel, 2006]. The naive definition of the transition probability in [Hummel, 2006] has turned out to be quite sensitive to the noise of GPS data particularly when the sampling interval is long. To achieve more robustness, [Krumm *et al.*, 2007] defines the transition probability by taking into account the travel time of the path connecting consecutive GPS points, and [Newson and Krumm, 2009] takes into account the corresponding travel distance, as we have discussed in Section 2. Despite its simplicity, the resulting HMM of [Newson and Krumm, 2009] is surprisingly more robust against noise and sparseness than the prior approaches.

Recently, [Goh *et al.*, 2012] has extended the HMM for online map-matching, where they consider a metric that penalizes the changes in driving directions that are not necessarily made at intersections. However, [Goh *et al.*, 2012] computes this metric for the path that minimizes the *distance* from a latent state to another. The approach of [Goh *et al.*, 2012] thus cannot find the true path unless the true path is the shortest. This is inevitable, because the transition probability is modeled with a support vector machine (SVM), so that it is intractable to find the optimal path with respect to the cost considered in [Goh *et al.*, 2012]. They also use 3,000 instances of transitions to train the SVM, while we only use 48 road segments to infer our weights with IRL.

Map-matching is still an active area of research despite its long history (see, e.g., the most recent line of work in [Karagiorgou and Pfoser, 2012; Buchin *et al.*, 2012; Haunert and Budig, 2012; Torre *et al.*, 2012; Liu *et al.*, 2012; Wei *et al.*, 2012; Javanmard *et al.*, 2012]). We have shown that IRL [Ziebart *et al.*, 2008] that has been developed in artificial intelligence can be used to significantly improve the quality of map-matching. We believe that more results from the artificial intelligence community, particularly those of learning behaviors from sensors [Pasula *et al.*, 1999; Chua *et al.*, 2011; Verwer *et al.*, 2011], can be applied to map-matching.

5 Conclusion

We have proposed that the transition probability of an HMM for map matching should be defined with a convex combination of multiple metrics, including the travel distance and the number of turns. The convex combination allows us to use Maximum Entropy IRL to infer the weights for the multiple metrics and the Dijkstra algorithm to find the path that minimizes the cost.

Our numerical experiments suggest that the quality of map-matching can be substantially improved by the use of the proposed transition probability. Interesting future directions include consideration of other metrics such as travel time and and toll. The definition of the cost of turn can also be refined.

Acknowledgments

This research was supported by JST, CREST.

References

[Bishop, 2007] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2nd edition, 2007.

[Brakatsoulas *et al.*, 2005] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proceedings of VLDB 2005*, pages 853–864, 2005.

[Buchin *et al.*, 2012] K. Buchin, T. J. Arseneau, S. Sijben, and E. P. Willems. Detecting movement patterns using brownian bridges. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 119–128, 2012.

[Chen *et al.*, 2011] D. Chen, A. Driemel, L. J. Guibas, A. Nguyen, and C. Wenk. Approximate map matching with respect to the fréchet distance. In *Proceedings of ALENEX11*, pages 75–83, 2011.

[Chua *et al.*, 2011] S.-L. Chua, S. Marsland, and H. W. Guesgen. Behaviour recognition in smart homes. In *Proceedings of IJCAI-11*, pages 2788–2789, 2011.

[Eisner *et al.*, 2011] J. Eisner, S. Funke, A. Herbst, A. Spillner, and S. Storandt. Algorithms for matching and predicting trajectories. In *Proceedings of ALENEX 2011*, pages 84–95, 2011.

[Funke and Storandt, 2011] S. Funke and S. Storandt. Path shapes: an alternative method for map matching and fully autonomous self-localization. In *Proceedings of ACM SIGSPATIAL GIS 2011*, pages 319–328, 2011.

[Goh *et al.*, 2012] 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 *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 776–781, 2012.

[Greenfeld, 2002] J. S. Greenfeld. Matching GPS observations to locations on a digital map. In *Proceedings of the 81st Transportation Reseach Board*, 2002.

[Haunert and Budig, 2012] J.-H. Haunert and B. Budig. An algorithm for map matching given incomplete road data. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 510–513, 2012.

[Hummel, 2006] B. Hummel. Map matching for vehicle guidance. In *Dynamic and Mobile GIS: Investigating Space and Time, J. Drummond and R. Billen, Editors*. CRC Press: Florida, 2006.

[Javanmard *et al.*, 2012] A. Javanmard, M. Haridasan, and L. Zhang. Multi-track map matching. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 394–397, 2012.

[Karagiorgou and Pfoser, 2012] S. Karagiorgou and D. Pfoser. On vehicle tracking data-based road network generation. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 89–98, 2012.

[Krumm *et al.*, 2007] J. Krumm, J. Letchner, and E. Horvitz. Map matching with travel time constraints. In *Proceedings of the Society of Automotive Engineers 2007 World Congress*, 2007.

[Lamb and Thiebaux, 1999] P. Lamb and S. Thiebaux. Avoiding explicit map-matching in vehicle location. In *Proceedings of the 6th World Conference on Intelligent Transportation Systems*, 1999.

[Liu *et al.*, 2012] K. Liu, Y. Li, F. He, J. Xu, and Z. Ding. Effective map-matching on the most simplified road network. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 609–612, 2012.

[Newson and Krumm, 2009] P. Newson and J. Krumm. Hidden Markov map matching through noise and sparseness. In *Proceedings of ACM SIGSPATIAL GIS 2009*, pages 336–343, 2009.

[Pasula *et al.*, 1999] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *Proceedings of IJCAI-99*, pages 1160–1167, 1999.

[Quddus *et al.*, 2007] 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*, 15:312–328, 2007.

[Raymond *et al.*, 2012] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue. Map matching with hidden Markov model on sampled road network. In *Proceedings of ICPR 2012*, 2012.

[Torre *et al.*, 2012] F. Torre, D. Pitchford, P. Brown, and L. Terveen. Matching GPS traces to (possibly) incomplete map data: bridging map building and map matching. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 546–549, 2012.

[Verwer *et al.*, 2011] S. Verwer, M. de Weerdt, and C. Witteveen. Learning driving behavior by timed syntactic pattern recognition. In *Proceedings of IJCAI-11*, pages 1529–1534, 2011.

[Wei *et al.*, 2012] H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan. Fast viterbi map matching with tunable weight functions. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 613–616, 2012.

[White *et al.*, 2000] C. E. White, D. Bernstein, and A. L. Kornhauser. Some map matching algorithms for personal navigation assistants. In *Transport. Res. Part C: Emerging Tech.*, volume 8, 1.6, pages 91–108, 2000.

[Yuan *et al.*, 2010] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of ACM SIGSPATIAL GIS 2010*, pages 99–108, 2010.

[Yuan *et al.*, 2011] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *Proceedings of ACM KDD '11*, pages 316–324, 2011.

[Ziebart *et al.*, 2008] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of AAAI-08*, pages 1433–1438, 2008.