PROBABILISTIC MAP-MATCHING FOR LOW-FREQUENCY GPS TRAJECTORY

Abstract

The ability to infer routes taken by vehicles from sparse and noisy GPS data is of crucial importance in many traffic applications. The task, known as map-matching, can be accurately approached by a popular technique known as ST-Matching. The algorithm is computationally efficient and has been shown to outperform more traditional map-matching approaches, especially on low-frequency GPS data. The major drawback of the algorithm is a lack of confidence scores associated with its outputs, which are particularly useful when GPS data quality is low. In this paper, we propose a probabilistic adaptation of ST-Matching that equips it with the ability to express map-matching certainty using probabilities. The adaptation, called probabilistic ST-Matching (PST-Matching) is inspired by similarities between ST-Matching and probabilistic approaches to map-matching based on a Hidden Markov Model. We validate the proposed algorithm on GPS trajectories of varied quality and show that it is similar to ST-Matching in terms of accuracy and computational efficiency, yet with the added benefit of having a measure of confidence associated with its outputs.

Keywords: map-matching, gps data, hidden markov model, dynamic programming

INTRODUCTION

Over the last years we have witnessed a rapid increase in the availability of GPS-receiving devices, such as smart phones or car navigation systems. The devices generate vast amounts of temporal positioning data that have been proven invaluable in various applications, from traffic management (Kühne, Schäfer, Mikat, & Lorkowski, 2003) and route planning (Gonzalez, Han, Li, Myslinska, & Sondag, 2007; Kowalska, Shawe-Taylor, & Longley, 2015; Li, Zeng, Zhang, Li, & Wu, 2011) to inferring personal movement signatures (Liao, Patterson, Fox, & Kautz, 2006).

Critical to the utility of GPS data is their accuracy. The data suffer from measurement errors caused by technical limitations of GPS receivers and sampling errors caused by their receiving rates. When digital maps are available, it is common practice to improve the accuracy of the data by aligning GPS points with the road network. The process is known as map-matching.

Most map-matching algorithms align GPS trajectories with the road network by considering *positions* of each GPS point, either in isolation or in relation to other GPS points in the same trajectory. The techniques, although often computationally efficient, are not very accurate in cases when the sampling rate is low or the street network complexity is high.

More advanced map-matching techniques utilise both *timestamps* and *positions* of GPS points in order to achieve a higher degree of accuracy. A highly popular example of a spatio-temporal algorithm is ST-Matching (Lou et al., 2009). It uses spatial information to find candidate roads for each GPS point and then seeks a sequence of candidate roads that best matches the temporal profile of the GPS trajectory. The algorithm is easy to implement, computationally efficient and has been shown to outperform purely spatial map-matching approaches, especially when the sampling rate is low. The major limitation of technique, and the before-mentioned spatial approaches, is its deterministic nature. It would always snap a GPS trajectory to a road network, regardless if it even came from the road network in the first place. The lack of confidence scores associated with its outputs might lead to very misleading results, especially when the data quality is low.

In isolation from the deterministic developments, *probabilistic* approaches to map-matching have been designed that address the issue of confidence using probabilities. They also belong to the class of spatio-temporal techniques as they use both spatial and temporal information when calculating probabilities of specific map-matching outputs. They typically represent the map-matching problem using a hidden Markov Model (HMM) where hidden states are true positions that are learnt from noisy GPS trajectories (Goh et al., 2012; Jagadeesh & Srikanthan, 2014; Newson & Krumm, 2009). The most likely map-matching output can then be efficiently learnt by applying a dynamic programming algorithm, such as the Viterbi algorithm, to the HMM lattice. Probabilistic approaches calculate the most likely or a few most likely road paths and output them together with their likelihoods. They are methodologically powerful, but largely isolated from the rest of the

map-matching community, often limiting their uptake by researchers and practitioners from other fields.

In this paper, we present a map-matching algorithm that bridges the gap between the deterministic and probabilistic classes of spatio-temporal algorithms. It is an adaptation of the well-established ST-Matching that turns it from being deterministic to fully probabilistic. The adaptation brings the best of the deterministic and probabilistic worlds into a highly accurate and computationally efficient map-matching algorithm that is capable of expressing levels of map-matching confidence. The proposal is inspired by apparent similarities between ST-Matching and HMM-based approaches to map-matching.

The paper is outlined as follows. It begins by introducing ST-Matching and a general HMM-based framework as its probabilistic counterpart. It analyses similarities and differences between the two approaches in order to propose a probabilistic adaptation of ST-Matching, called probabilistic ST-Matching or PST-Matching in short. It evaluates the robustness of PST-Matching on a range of GPS trajectories of varied frequencies and levels of noise. Similarly to ST-Matching, the proposed algorithm shows high accuracy on datasets with low GPS frequency, yet with the added benefit of confidence scores associated with its outputs.

PROBLEM STATEMENT

In this section, we define the problem of probabilistic map-matching.

Definition 1 (*GPS trajectory*): A sequence of GPS points, where each GPS point contains latitude, longitude and timestamp.

Definition 2 (*Road network*): A directed graph with vertices representing road intersections and edges representing road segments. Bidirectional road segments are represented by two edges, each corresponding to a single direction of flow. Roads and intersections can be uniquely identified using their IDs.

Definition 3 (*Path*): A connected sequence of street segments in the road network.

Given a road network and a GPS trajectory, the goal of probabilistic map-matching is 1) to find the most likely path that the GPS trajectory was generated from and 2) to quantify the confidence that the path is indeed the true path taken.

Fig. 1. Exemplary road network with a GPS trajectory to be map-matched.

METHODOLOGY

In this section, we describe our probabilistic ST-Matching algorithm in detail. We begin by introducing its components: the ST-Matching algorithm (deterministic) and a general HMM-based approach (probabilistic). We then outline modifications required to make the ST-Matching algorithm fit the general HMM-based framework, thus turning it into a probabilistic technique.

ST-Matching Algorithm

ST-Matching is a deterministic map-matching approach that combines spatial and temporal information to effectively align low-sampling-rate GPS trajectories with the road network. It is easy to implement and has

been shown to outperform more traditional map-matching approaches in terms of accuracy and running time. Its architecture consists of two basic steps: candidate graph preparation and best path computation.

1) Candidate graph:

Candidate graph stores all possible true paths given a GPS trajectory. Nodes of the graph are candidate position for each GPS observation, edges are shortest road paths between neighbouring candidate positions. The preparation of the graph involves the following steps.

Firstly, candidate positions are computed by retrieving road segments within radius r of each GPS observation and then finding a position on each segment at the shortest distance to the relevant observation. The procedure is exemplified in Figure 2. The obtained candidate positions are represented as nodes in the candidate graph. The number of candidate positions can differ among GPS observations, depending on the number of street segments within the search radius. In the paper, we use $c_{i,j}$ to denote the j^{th} candidate position of GPS observation p_i.

(a) candidate positions **(b)** candidate graph

Fig. 2. Exemplary road network with a GPS trajectory to be map-matched.

Secondly, shortest paths between pairs of candidate positions at adjacent time steps are evaluated based the road topology. They are represented as edges in the candidate graph, as shown in Figure 2b.

Finally, the nodes and edges of the candidate graph are weighted based on the spatio-temporal profile of the GPS trajectory and the topology of the underlying road network. Their weights reflect their *observation* and *transmission probabilities*, respectively.

Definition 4 (*Observation probability*): Given a GPS observation p_i at time step i and a corresponding candidate position $c_{i,j}$, it defines the probability that the GPS observation p_i is emitted from the candidate position $c_{i,j}$.

The observation probability is specified as a Gaussian distribution of the distance between p_i and $c_{i,j}$:

$$\mathrm{N}(\mathrm{c}_{i,j}) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(\mathrm{x}_{i,j}-\mu)^2}{2\sigma^2}} \quad (1)$$

where $x_{i,j}$ is the distance between p_i and $c_{i,j}$. The mean μ of the distribution is set to zero, the standard deviation σ is empirically estimated.

Definition 5 (*Transmission probability*): Given two candidate positions $c_{i-1,j}$ and $c_{i,k}$ for two neighbouring GPS observations p_{i-1} and p_i, it defines the probability that the "true" path from p_{i-1} to p_i follows the shortest path from $c_{i-1,j}$ to $c_{i,k}$.

The transmission probability is defined as follows:

$$\mathrm{V}(\mathrm{c}_{\mathrm{i-1,j}} \rightarrow \mathrm{c}_{\mathrm{i,k}}) = \frac{d_{i-1 \rightarrow i}}{s_{(i-1,j) \rightarrow (i,k)}} \quad (2)$$

where $d_{i-1 \rightarrow i}$ is the Euclidean distance from p_{i-1} to p_i and $s_{(i-1,j) \rightarrow (i,k)}$ is the length of the shortest path from $c_{i-1,j}$ to $c_{i,k}$.

The above definition only considers spatial information when calculating the likelihood of transmission. A spatio-temporal version of the transmission probability is also considered in the original paper (Lou et al., 2009) and could be easily incorporated into the methodology presented in this paper in the future.

2) Best path search:

Once the candidate graph is defined and its nodes and edges are weighed according to the observation and transmission probabilities, respectively, a dynamic programming method is applied in order to find a path through the graph with the maximum weight. The path represents the most likely "true" path that the GPS trajectory was generated from.

The method proposed in (Lou et al., 2009) calculates the most likely path by recursively evaluating the following equation:

$$f(t,k) = N(c_{t,k}) + \max_j [f(t-1,j) \cdot V(c_{t-1},j \rightarrow c_{t,k})] \quad (\mathbf{3})$$

with $f(1,k)$ initialised to $f(1,k) = N(c_1,k)$. In the above equation, $f(t,k)$ represents the total weight of the most likely sequence of positions ending at position $c_{t,k}$, based on GPS observations at time steps $1{:}\,t$. Once the recursion reaches $t = T$, the obtained sequence of positions and the shortest paths between them form the most likely path given the GPS trajectory.

General HMM-Based Approach

Hidden Markov Model (HMM) is an established framework for probabilistic time-series modelling. It provides a principled way of representing uncertainty in measurements taken over time and as such is suitable for modelling uncertainty inherent to noisy GPS trajectories.

There have been numerous probabilistic approaches to map-matching based on HMMs (Goh et al., 2012; Jagadeesh & Srikanthan, 2014; Newson & Krumm, 2009). They typically use a HMM to represent possible "true" paths and their probabilities and then search for one or more paths with the highest probabilities as the map-matching output.

Drawing similarities to the ST-Matching algorithm, a HMM can be understood as a candidate graph from which the most likely path can be retrieved via a dynamic programming routine known as the Viterbi algorithm.

1) Candidate graph:

HMM provides a graph structure for storing possible paths in a probabilistic manner. Nodes of the graph are hidden states that can represent candidate positions at each time step. Edges are transitions between the hidden states and can represent possible paths taken between candidate positions at adjacent time steps. The structure of the graph in the context of map-matching is, in fact, equivalent to that of the candidate graph in ST-Matching (see Figure 2b for an example).

Nodes are assigned emission probabilities that quantify the likelihood of the observations given the hidden states at each time step. The emission probability is defined as the conditional probability $p(p_i|c_{i,j})$ of observing p_i given that the true state at time step i is $c_{i,j}$. In the map-matching context, it is equivalent to the observation probability given in Definition 4.

Edges are given so-called transition probabilities. The transition probability is a discrete conditional distribution $p(c_{i,k}|c_{i-1,j})$ that defines the probability of transitioning from hidden state $c_{i-1,j}$ at time step $i-1$ to another hidden state $c_{i,k}$ at time step i. In our context, it is the probability of following the shortest path between candidate positions corresponding to these hidden states. The transition probability can be any discrete distribution, such as the ST-Matching transmission probability in Definition 5, but more rigorously defined to ensure that basic rules of probability are satisfied. In particular, the following statement must hold:

$$\sum_k p(c_{i,k}|c_{i-1,j}) = 1 \quad (\mathbf{4})$$

2) Best path search:

The most likely path is inferred as the most likely sequence of hidden states using a dynamic programming technique known as the Viterbi algorithm (Bishop, 2006). The algorithm finds the state sequence with the highest joint probability over the states and the GPS observations, which is the product of the emission and transition probabilities along the sequence. It efficiently searches the space of all possible sequences by recursively evaluating the maximum joint probability for each state at each time step as follows:

$$w_{t,k} = p(p_t|c_{t,k}) \cdot \max_j [w_{t-1,j} \cdot p(c_{t,k}|c_{t-1,j})] \quad (\mathbf{5})$$

with $w_{t,k}$ initialised to $w_{t,k} = p(p_1|c_{1,k})$. In the above equation, $w_{t,k}$ represents the joint probability of the most likely sequence of hidden states until time step t. Once the recursion reaches time step $t = T$, the most likely path is formed by the most likely sequence of hidden states and the shortest paths between them.

Notice that the Viterbi algorithm is almost equivalent to the ST-Matching algorithm in (3). If one replaced the observation and transmission probabilities in (3) with the more general emission and transition probabilities of the Viterbi algorithm, respectively, the ST-Matching algorithm would only differ in the way it applies the most recent emission probability to the result of the *max* operation (addition instead of multiplication). However, it lacks the probabilistic treatment of the Viterbi approach which not only finds the most likely path but also quantifies the likelihood that it is indeed the true path taken using its joint probability.

Probabilistic ST-Matching Algorithm

Having introduced the ST-Matching algorithm and a general HMM-based approach, it has become apparent that the two approaches share a lot of similarities. In this section, we formalise the observation and outline modifications required to make the ST-Matching algorithm fit the probabilistic framework, thus giving it the ability to express map-matching confidence in a probabilistic manner. We term the proposed modification the probabilistic ST-Matching (PST-Matching) algorithm.

1) Candidate graph:

Candidate graph of PST-Matching is very similar to the original ST-Matching graph. It shares the same graphical structure and defines the observation probability according to the same formula in (1). It requires a modified transmission probability, however, as the original definition in (2) does not satisfy basic rules of conditional probabilities, such as the summation rule in (4). We satisfy the requirement by proposing a normalised transmission probability:

$$\mathrm{V_{pst}}(\mathrm{c_{i-1,j}} \rightarrow \mathrm{c_{i,k}}) = \frac{\mathrm{V}(\mathrm{c_{i-1,j}} \rightarrow \mathrm{c_{i,k}})}{\sum_k \mathrm{V}(\mathrm{c_{i-1,j}} \rightarrow \mathrm{c_{i,k}})} \quad (\mathbf{6})$$

2) Best Path Search:

The dynamic programming routine of PST-Matching is a modification of that of ST-Matching (3) that turns it into a Viterbi algorithm. The modification simply requires replacing the addition operation in (3) with multiplication. When applied to the candidate graph outlined above, the proposed algorithm takes the following recursive form:

$$f_{pst}(t,k) = N(c_{t,k}) \cdot \max_j [f_{pst}(t-1,j) \cdot V_{pst}(c_{t-1}, j \rightarrow c_{t,k})] \quad (\mathbf{6})$$

with $f_{pst}(1,k)$ initialised to $f_{pst}(1,k) = \mathrm{N}(\mathrm{c_1}, k)$. As in any Viterbi algorithm, the quantity stored in $f_{pst}(t,k)$ at the final time step is the joint probability of the most likely path. It serves as a measure of map-matching confidence that the original ST-Matching algorithm is lacking.

METHOD VALIDATION

Data

The dataset used for validating the proposed algorithm is a complete GPS trajectory of a police patrol vehicle during its night shift (9pm to 7am) in the London Borough of Camden on February 9th 2015. The dataset contains 4,800 GPS points that were emitted roughly every second when moving.

Further datasets of degraded quality are artificially created from the acquired data in order to test the robustness of the proposed algorithm on a range of GPS trajectories of varied sampling rates and levels of noise. Their sampling rate is manipulated by removing GPS points at chosen intervals. Their level of noise is controlled by perturbing GPS point by Gaussian noise with zero mean and a chosen standard deviation. There is already some random Gaussian noise inherent to the data. However, since Gaussian distributions are additive, i.e. adding two Gaussian random variables results in another Gaussian random variable with mean and variance equal to the sum of the added means and variances, any additional amount of noise can be simulated once the standard deviation of the original noise distribution is empirically found.

Accuracy Testing

Since there is no ground truth available, we propose a validation framework based on the well-established technique of cross-validation (Barber, 2012). We split available GPS observations into training and test sets according to the split ratio of 9:1, i.e. 90% training and 10% testing. In practice, this equates to us removing every 10th GPS point from each GPS trajectory for training (see Figure 3). We proceed by aligning the trajectory of training GPS points with the road network using our proposed PST-Matching algorithm. We then record how far off the predicted path each test point is. The more off, the more erroneous our map-matching proposal. We use the average distance across all test points as the measure of map-matching error made.

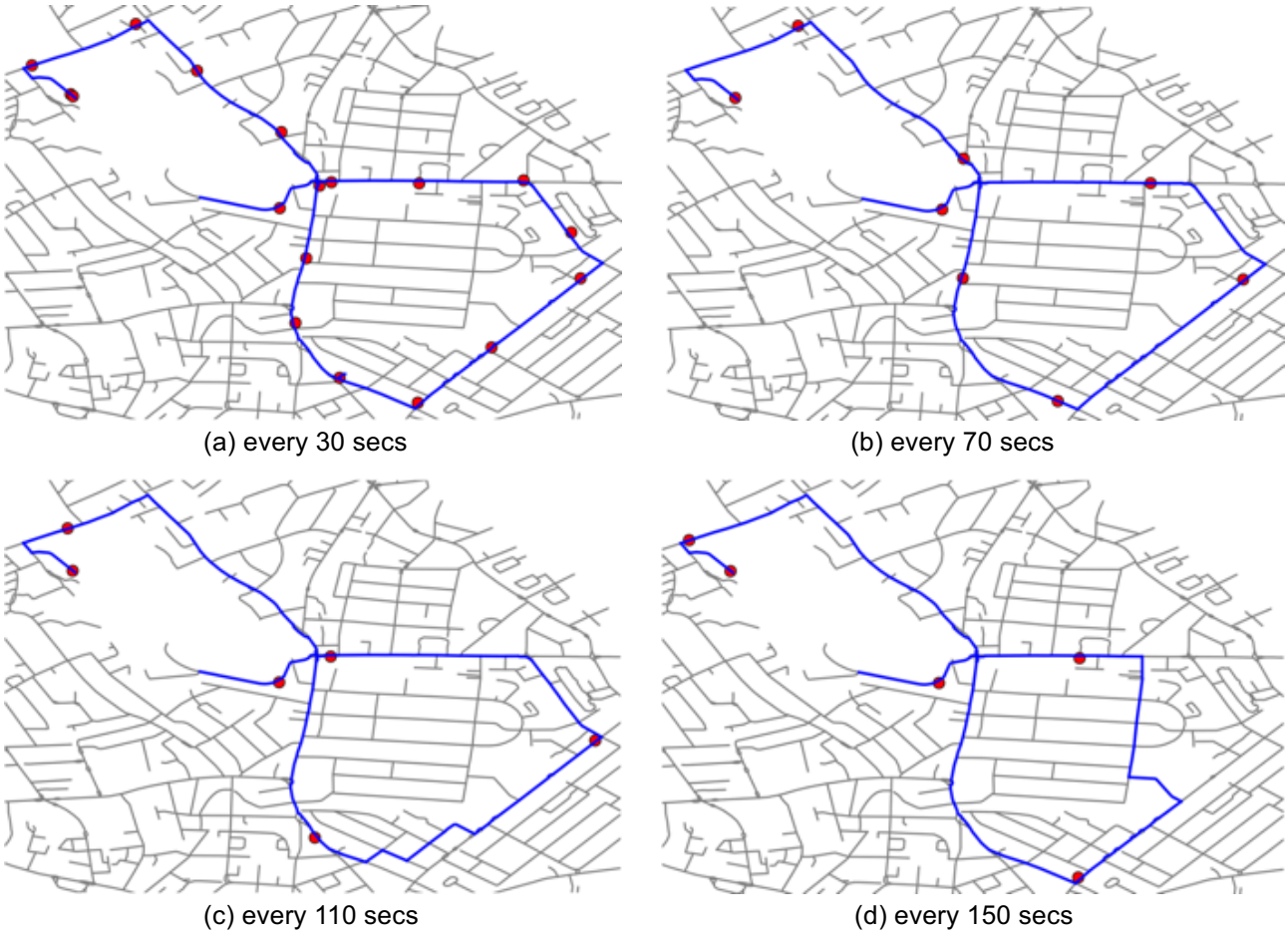
Fig. 3. Exemplary GPS trajectory with points split into training and test sets.

RESULTS

We tested the proposed PST-Matching algorithm on datasets of varied quality and compared its accuracy against that of the original ST-Matching algorithm. See Figure 4 for exemplary map-matching outputs and Figure 5 for a summary of the algorithm's performance across all datasets. Similarly to ST-Matching, the algorithm shows high accuracy on datasets with noise as high as 30 meters standard deviation and sampling rates of up to 90 seconds. Such extreme conditions are rarely found in real datasets; hence the algorithm should be successful on real GPS trajectories without the need for any prior adjustments or parameter fitting.

We noticed a slight drop in performance at very low sampling rates of 1-2 seconds (see Figure 5a). This is likely caused by the fact that frequent, noisy observations tend to pull rather violently towards different path proposals. The algorithm also gradually deteriorates at higher levels of measurement noise, unlike the original

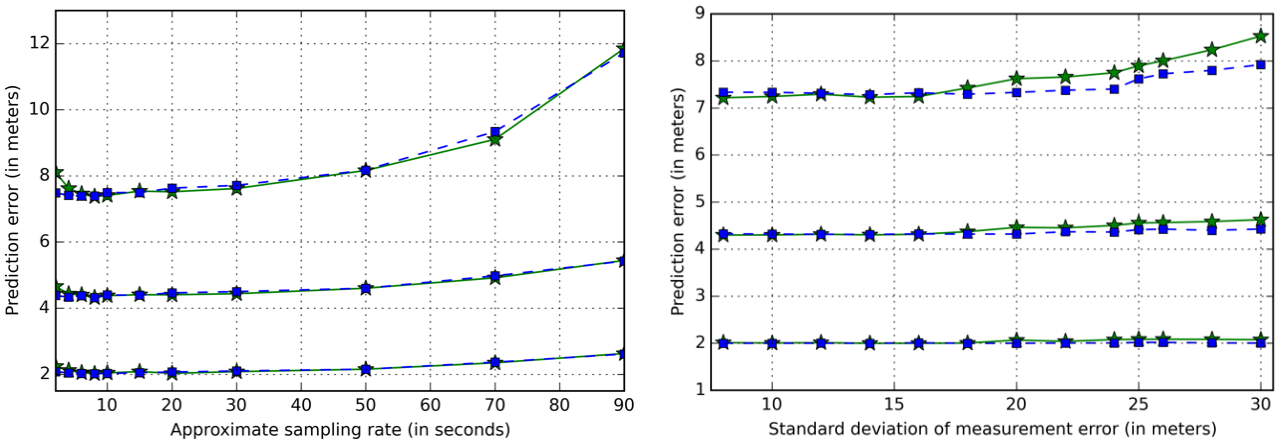
ST-Matching algorithm (see Figure 5b). This is due to the normalisation of the transmission probability according to (6), which fails to penalise candidate points that are clearly off the "true" path. As a result, as the measurement noise increases, there are more points off the path that PST-Matching accidentally includes in the most likely path.



(a) every 30 secs (b) every 70 secs

(c) every 110 secs (d) every 150 secs

Fig. 4. Exemplary PST-Matching solutions at different sampling rates.

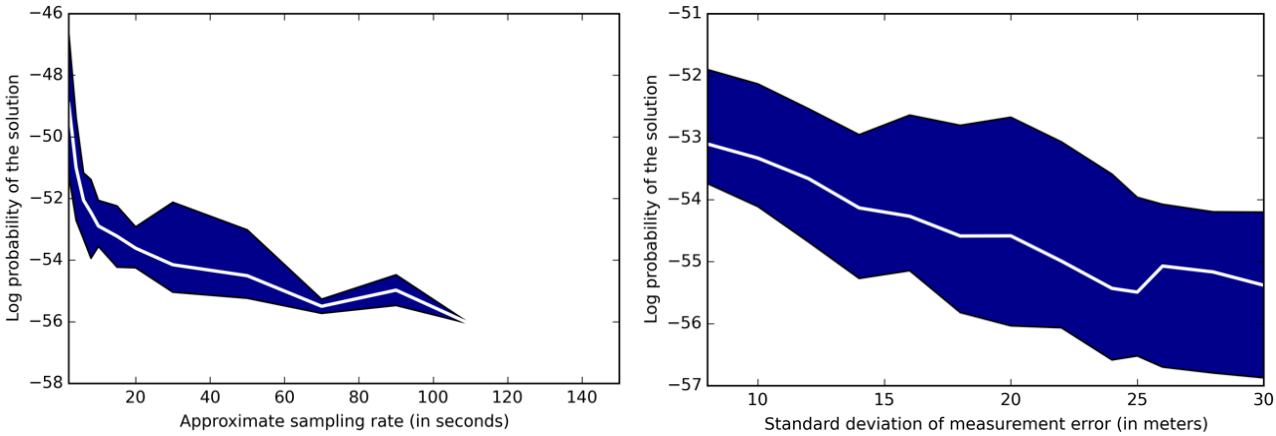


(a) sampling rate (b) measurement noise

Fig. 5. Accuracy of PST-Matching (green) and ST-Matching (blue) on datasets with varied GPS sampling rates and noise represented as 25th, 50th and 75th percentiles of map- matching errors.

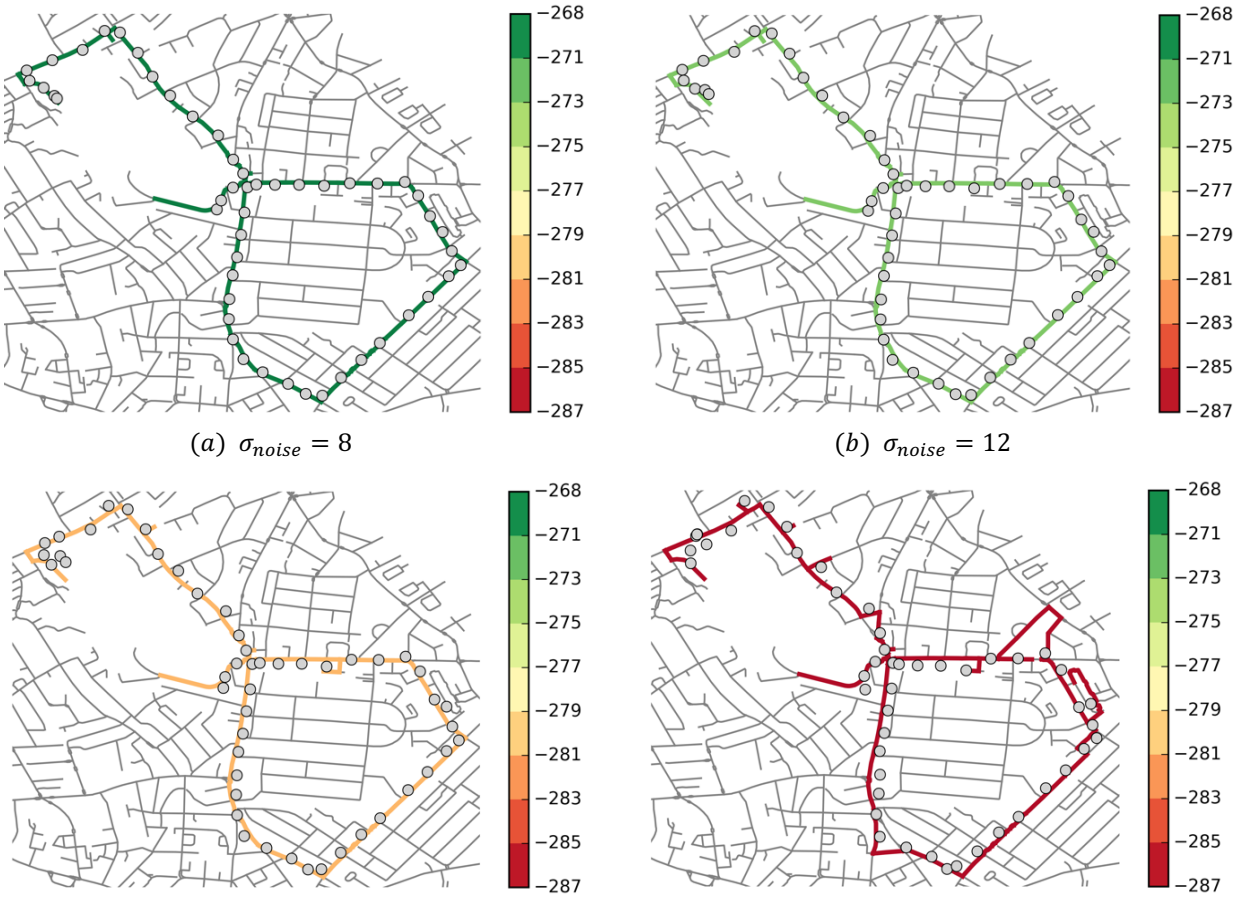
We investigated how confidence of PST-matching solutions, expressed as joint probabilities, changes with the sampling rate and the level of noise of GPS data. Since the joint probability of a solution is the product of observation and transmission probabilities along the most likely sequence (see (7)), its value depends on the

length of the input GPS sequence. We ensured that the dependence did not skew our analysis by applying PST-Matching to a sliding window (of length ten) over input GPS trajectories. The idea guaranteed that confidence scores were meaningful and gave the algorithm the ability to process GPS trajectories in an online manner. The obtained confidence scores are shown in Figure 6. On average, the scores decline as data become noisy and sparse. This trend is exemplified in Figure 7, where after adding noise to the data, the quality and confidence of the map-matching output gradually drops. These intuitive results show that the confidence scores are closely aligned with the quality of map-matching results and, as such, could prove indispensable when dealing with GPS data of unknown quality.



(a) sampling rate (b) measurement noise

Fig. 6. Map-matching confidence on datasets with varied GPS sampling rates and noise represented as 25th, 50th and 75th percentiles of log probabilities of map-matching outcomes.



$(a)\ \sigma_{noise} = 8$ $(b)\ \sigma_{noise} = 12$

$(c)\ \sigma_{noise} = 18$ $(d)\ \sigma_{noise} = 30$

Fig. 7. Exemplary PST-Matching outcome with confidence (expressed as log probability) on a GPS trajectory with varied noise standard deviation (in meters).

CONCLUSIONS

In this paper, we propose a new probabilistic map-matching algorithm called PST-Matching for aligning sparse and noisy GPS trajectories with a road network. The algorithm is a probabilistic extension of a popular deterministic algorithm called ST-Matching that has been shown to outperform more traditional map-matching algorithms on datasets of low sampling rates. The proposal brings high computational efficiency and accuracy of ST-Matching into the probabilistic world, hence giving it the ability to express confidence about its outputs. The measure of confidence is particularly important when dealing with traffic datasets of low accuracy. We validate the proposed algorithm on a range of GPS trajectories of varied quality to show that it has as high accuracy on low-frequency and noisy datasets as the original ST-Matching algorithm, yet with the added benefit of expressing beliefs about the quality of its output using probabilities.

REFERENCES

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Goh, C. Y., Dauwels, J., Mitrovic, N., Asif, M. T., Oran, A., & Jaillet, P. (2012). Online map-matching based on Hidden Markov model for real-time traffic sensing applications. In *2012 15th International IEEE Conference on Intelligent Transportation Systems* (pp. 776–781). IEEE.

Gonzalez, H., Han, J., Li, X., Myslinska, M., & Sondag, J. P. (2007). 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).

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

Kowalska, K., Shawe-Taylor, J., & Longley, P. (2015). Data-driven modelling of police route choice. In *Proceedings of the 23rd GIS Research UK conference*.

Kühne, R., Schäfer, R.-P., Mikat, J., & Lorkowski, S. (2003). New Approaches for Traffic Management in Metropolitan Areas. In *Proceedings of the 10th Symposium on Control in Transportation Systems*. Tokyo.

Li, Q., Zeng, Z., Zhang, T., Li, J., & Wu, Z. (2011). Path-finding through flexible hierarchical road networks: An experiential approach using taxi trajectory data. *International Journal of Applied Earth Observation and Geoinformation*, *13*(1), 110–119.

Liao, L., Patterson, D. J., Fox, D., & Kautz, H. (2006). Building personal maps from GPS data. *Annals of the New York Academy of Sciences*, *1093*, 249–65.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., & Huang, Y. (2009). Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09* (p. 352). inproceedings, New York, New York, USA: ACM Press.

Newson, P., & Krumm, J. (2009). Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09* (p. 336). New York, New York, USA: ACM Press.