9-2016

Is only one GPS position sufficient to locate you to the road network accurately?

Hao WU

Weiwei SUN

Baihua ZHENG
Singapore Management University, bhzheng@smu.edu.sg

Citation

WU, Hao; SUN, Weiwei; and ZHENG, Baihua. Is only one GPS position sufficient to locate you to the road network accurately?. (2016). *UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Heidelberg, Germany, September 12-16, 2016*. 740-751.
Available at: https://ink.library.smu.edu.sg/sis_research/3585

Published in UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing:
Heidelberg, Germany, September 12-16, 2016, Pages 740-751.
http://doi.org/10.1145/2971648.2971702

Is Only One GPS Position Sufficient to Locate You to The Road Network Accurately?

Hao Wu[†] **Weiwei Sun**[†] **Baihua Zheng**[‡]
[†]School of Computer Science, Fudan University, Shanghai, China
[†]Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China
[‡]Singapore Management University, Singapore
{wuhao5688, wwsun}@fudan.edu.cn, bhzheng@smu.edu.sg

ABSTRACT

Locating only one GPS position to a road segment accurately is crucial to many location-based services such as mobile taxi-hailing service, geo-tagging, POI check-in, etc. This problem is challenging because of errors including the GPS errors and the digital map errors (misalignment and the same representation of bidirectional roads) and a lack of context information. To the best of our knowledge, no existing work studies this problem directly and the work to reduce GPS signal errors by considering hardware aspect is the most relevant. Consequently, this work is the first attempt to solve the problem of locating one GPS position to a road segment. We study the problem in a data-driven view to make this process ubiquitous by proposing a tractable, efficient and robust generative model. In addition, we extend our solution to the real application scenario, i.e., taxi-hailing service, and propose an approach to further improve the result accuracy by considering destination information. We use the real taxi GPS data to evaluate our approach. The results show that our approach outperforms all the existing approaches significantly while maintaining robustness, and it can achieve an accuracy as high as 90% in some situations.

ACM Classification Keywords
H.2.8 Database Applications: Spatial databases and GIS.

Author Keywords
Location-based services; GPS; positioning; map matching

INTRODUCTION

With the development of the mobile technology, GPS positioning is now widely applied in location-based services (LBSs) [37, 38] such as discovering urban functional zones [34, 36], human mobility prediction [10, 25, 30], and ride-sharing [6, 16]. Locating a GPS position to the road network without any context information serves as a building block for many LBSs including taxi-hailing, check-in/geo-tagging in location-based social networks (e.g., Foursquare, Twitter), etc. For convenience, we name the problem of mapping a GPS position to a road of a digital map without context information as *Single-point Map Matching (SMM)* problem.

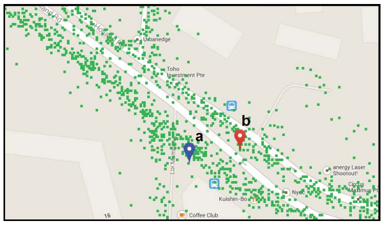
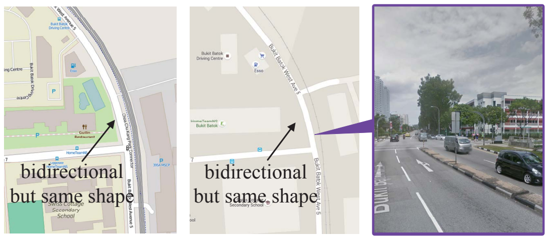


Figure 1. GPS points on a digital map

Let us take a quick view of taxi-hailing service, one of the application scenarios of SMM. When a commuter is calling for a taxi, if the APP simply maps the GPS point to the nearest road which may be the wrong answer, it will bring inconvenience to the driver as she/he might not be able to find the commuter. It could become worse during rush hour as it might take some time for the taxi to reach another road. Even though one can leverage the moving GPS position series of the commuter in a certain time interval to get a more precise answer, we should make this process ubiquitous for every scenario including those commuters standing steadily with only one position available.

SMM is a challenging problem mainly because of following four reasons. First, GPS signals have inevitable errors, e.g., the combination of noise, bias and blunders [8]. The bias is mainly resulted by multipath effect [9, 11, 19]. Multipath occurs when a radio signal is split by obstacles, e.g. high buildings, which is very common in urban areas [7, 27]. Figure 1 visualizes GPS points on a road network (Google Map). It is not hard to observe that the distribution of GPS points is not consistent with the road segments shown in the map. For given GPS points *a* and *b* in the figure, if we simply map them to the nearest road segment, the answer may not be correct.

Second, besides the GPS errors, the errors on the digital map also lead to the difficulty of SMM problem. Most of maps, including both commercial and noncommercial ones, are generated manually. Consequently, some roads displayed in the map might not precisely align with the roads in the real world, some new roads might be missing from the map, and some closed roads might still present in the map [24, 29, 33]. Con-



(a) OpenStreetMap (b) Google map (c) Real photo

Figure 2. Examples of bidirectional roads on digital maps

sidering the relatively low update frequency of map data and the dynamic nature of real road network, map data in most cases are not exactly the same as real road networks in use.

Third, SMM problem cannot refer to any context information, i.e., the predecessors and successors in a trajectory. The only information SMM has is a *single* GPS position represented by a latitude and a longitude.

Forth, most digital maps do not differentiate the *spatial* difference between the two sides of a bidirectional road, i.e., they represent the two sides using the *same shape*. Figure 2(a) and 2(b) show a region in Singapore on Google map and OpenStreetMap respectively, and Figure 2(c) shows the photo. We can find out these roads shown in the maps are bidirectional in nature but they are all represented by the same shape in maps. More specifically, based on OpenStreetMap which is the largest open source digital map, we find out that 78.6% of roads in Singapore have the same road shape as their reverse sides, such as $\{r_1, r_2\}$, $\{r_3, r_4\}$ and $\{r_5, r_6\}$ in Figure 3. Thus, even we can confirm that a GPS position should be mapped to an edge of a bidirectional road, e.g., the edge between node v_2 and v_5 in Figure 3, the problem has not been fully solved as we need to figure out whether it is located on r_5 or r_6.

Although we understand all the challenges of SMM problem, we still believe that it is solvable because of following observations. First, the bias of GPS signals can be inferred given a large number of historical GPS samples. This could be also observed from the distribution of GPS points depicted in Figure 1. Second, the width of bidirectional roads is not that small, which means the distribution of historical GPS points w.r.t. one side of the road will be different from that w.r.t. the other side of the road. When we accumulate sufficient historical GPS points, we might be able to learn the difference of GPS points w.r.t. two different sides of a road.

To the best of our knowledge, there is no existing work *directly* solve SMM problem . The problem of GPS error reduction is the closest but is still different from SMM. In addition, all the works on GPS error reduction [9, 11, 27, 28] solve the problem from a hardware aspect. We tackle SMM problem using a data-driven approach because of following two main reasons. First, a data-driven approach is believed to be more general. Second, it is transparent from the application layer which makes it ubiquitously applicable in any device/application. In addition, studying SMM in a data-driven view does not conflict with those hardware based works. For example, in order to solve SMM problem, we can first adopt hardware based approaches to reduce the GPS errors, and then apply the data-drive approach to further improve the accuracy.

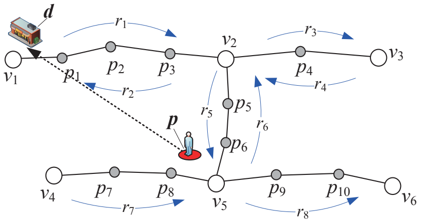


Figure 3. Example of road network and SMM/SMMD problem

This paper studies the distributions of historical GPS sampling points and tries to tackle SMM problem using a generative model. In addition, we show how to plug our SMM solution into real-life problems which may have more information available by an example of taxi-hailing service. More specifically, we extend SMM problem to SMMD where the location mapping is based on a GPS point and the destination of the journey. Our solution can be adjusted to online learning, which allows the parameters of the model to be twisted continuously based on the feedback from users. As a summary, we make four major contributions in this paper.

- We formalize SMM problem. This is the first work on SMM problem through a data-driven view. We implicitly catch the fixed bias of GPS points by a generative model. Our model is tractable and has a closed form solution which makes it easy-trainable.
- We show how to plug our SMM solution to SMMD problem in the taxi-hailing scenario. We model the tendency of users when calling a taxi to study the SMMD problem.
- We adapt our model to online learning framework which is useful in the real applications with continuous feedbacks.
- We conduct comprehensive experimental studies via a large amount of real-world taxi data. The results show that the proposed approach outperforms the existing approaches significantly and also demonstrate its robustness.

PRELIMINARY

In the following, we first introduce two important definitions on road network and road segment, and then formalize SMM and SMMD problems studied in this paper.

Definition 1. (***Road network.***) A road network is modelled as a directional graph $G(V, E)$, where V refers to the set of vertices (i.e., crossroads) and E refers to the set of edges (i.e., road segments).

Definition 2. (***Road segment.***) Given a road network $G(V, E)$, a road segment $r \in E$ is a directed edge from a source vertex $r.s \in V$ to an ending vertex $r.e \in V$, with the direction represented by $r.s \rightarrow r.e$. Unlike the definition in conventional road networks, here the road segment has a new field denoted as *r.shape* which is a list of intermediate points describing the shape of the road, i.e., polyline representation.

Problem 1. (**SMM problem.**) Given a road network G and a GPS position p generated by an object on a road segment r, the *Single-point Map Matching* (*SMM*) problem aims to find out the road segment $r \in G.E$ where p is actually located, without relying on any other information.

Problem 2. (**SMMD problem.**) In taxi-hailing scenario, the destination a commuter wants to go is usually available. Given a road network G, a GPS position p generated by commuter u on a road segment r, and a destination $d \in G.V$ that u wants to go, *Single-point Map Matching with Destination (SMMD)* problem aims to find out the road segment $r \in G.E$ where p is actually located.

Figure 3 plots an example road network with $V = \{v_1, v_2, \cdots, v_6\}$, and $E = \{r_1, r_2, \cdots, r_8\}$. Take road segment r_1 as an example. $r_1.direction = v_1 \rightarrow v_2$ and $r_1.shape = \{v_1, p_1, p_2, p_3, v_2\}$. The road between v_1 and v_2 is a bidirectional road and the reverse side of r_1 is r_2 with the same shape but different direction, i.e., $r_2.direction = v_2 \rightarrow v_1$. For the SMM problem, e.g., given point p in Figure 3, road segments r_5, r_6, r_7 and r_8 are the candidate road segments where p is located and SMM needs to locate one road segment that is most likely for p to be located on. If we know the destination information, i.e., d, of the journey started at point p and we want to locate p to a road segment, which is an example of SMMD.

SOLUTION TO SMM PROBLEM

We understand from Introduction Section that some biases for GPS positioning do exist in many places, with some caused by the multipath effect and others caused by the in-correctness of the digital map. In other words, we can leverage these biases using the historical data to tackle problem SMM. If we can capture all the biases in the map, we can locate the GPS sample p closer to its actual position by correcting p according to the bias near p. Unfortunately, it is not easy to capture and express the bias explicitly since bias is changing in terms of both *direction* and *scale* everywhere.

In order to avoid working on those intractable biases explicitly and directly, we intend to tackle the SMM problem from a probabilistic viewpoint. Let r_i be one of the candidate road segments that p might be mapped to, and $\mathbb{C}$ be the candidate set that preserves all the candidate road segments. Finding out the road segment r^* which p has the highest probability to be on, is equivalent to solve a Maximum-a-Posteriori (MAP) problem or a classification/prediction problem, where p is the object we need to predict and $r_i \in \mathbb{C}$ can be regarded as the class labels. i.e., $r^* = \arg\max_{r_i \in \mathbb{C}} P(r_i|p)$. In the following, we propose a new generative model, namely PSMM, as a solution. The quantitative comparison of existing classifiers and PSMM will be presented in the Evaluation Section.

Obtaining The Ground Truth

For all classification (supervised learning) problems, labels/ground truths should be attained. As manual labeling is obviously impractical for SMM problem, we explain how to get the label of each historical GPS sample automatically. To do so, a dataset of GPS series, i.e., trajectories, should be available. Although mapping a single GPS position to the road network is difficult, mapping a trip with series of GPS positions will be much easier and more reliable, handled by a well-studied technology named as *map matching*. Among all of the map-matching approaches, hidden Markov Model (HMM) based approaches are the best choice [1]. As stated in [20], it is very robust to noise and sampling rate, e.g., it achieves around 98% accuracy when sampling rate is 30s and the noise standard deviation is 15m. The accuracy is sufficiently high to get the label of each GPS position in a trajectory and we pick up these GPS positions as training data.

```
For each road $r$ drawn from $\mathcal{C}(\pi)$:
    For every road position $\tau$ drawn from $\mathcal{P}(\omega_r)$:
        Draw GPS position $p \sim \mathcal{N}\left(\begin{pmatrix} f_{r,x}(\tau)+b_{r,x}(\tau) \\ f_{r,y}(\tau)+b_{r,y}(\tau) \end{pmatrix}, \Sigma_r(\tau)\right)$;
```

Figure 4. Exact generation process of a GPS sample

Generation Process of GPS Point

Before presenting our model, let us use a parameterized form to represent a polyline. That is to say, for a 2-d polyline $r.shape$, any point (x, y) on the polyline satisfies that

$$x = f_{r,x}(\tau), y = f_{r,y}(\tau), 0 \leq \tau \leq 1$$

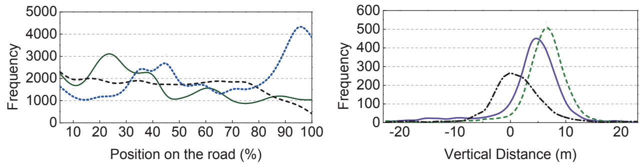
This means the coordinates (x, y) of intermediate points of a 2-d polyline $r.shape$ can be expressed as functions $f_{r,x}(\tau)$, $f_{r,y}(\tau)$ of parameter τ. In other words, when an object is moving along $r.shape$ from $r.s$ to $r.e$, the movement can be expressed by altering the parameter τ from 0 to 1.

Figure 4 shows the exact generation process of a GPS position. First, an object is located on one road segment r, which can be modelled to be drawn from a categorial distribution $\mathcal{C}(\pi)$. Then the true position of this object can be represented by the offset, i.e., τ in the parameterized representation, in r. Thus, τ should be drawn from certain distribution $\mathcal{P}(\omega_r)$ with parameter ω_r. After deciding the true position $(f_{r,x}(\tau), f_{r,y}(\tau))$ of the object, the object pushes a GPS position request and GPS is returned with some biases and noises if any. Consequently, the GPS position can be assumed to be drawn from a 2-d Gaussian with mean $\begin{pmatrix} f_{r,x}(\tau) + b_{r,x}(\tau) \\ f_{r,y}(\tau) + b_{r,y}(\tau) \end{pmatrix}$ and covariance matrix $\Sigma_r(\tau)$, where $\begin{pmatrix} b_{r,x}(\tau) \\ b_{r,y}(\tau) \end{pmatrix}$ can be regarded as the fixed bias and $\Sigma_r(\tau)$ models the random noise. We assume that every place should have its own constant bias and random noises due to the difference of the environment, and that's why we parameterize both mean and covariance of the 2-d Gaussian by τ. Note that, as mentioned above, the identical representation of bidirectional road is a problem we need to resolve. However, as the width of the road in the real world cannot be ignored, thus the distributions of the points corresponding to two directions of a bidirectional road is different. As a result, we can regard them as biases and the bidirectional road problem can be solved under the solution of our generative model.

Given a point p and a candidate set $\mathbb{C}$ of road segments, the model tries to find the road $r \in \mathbb{C}$ that maximizes the posterior $P(r|p)$ (equally speaking, joint distribution $P(r, p)$ as $P(p)$ is a constant w.r.t different rs), i.e., $r^* = \arg\max_{r \in \mathbb{C}} P(r, p)$. In detail,

$$\begin{aligned} P(r, p) &= P(r) \int P(p|r, \tau) P(\tau|r) \mathrm{d}\tau \\ &= C(r|\pi) \int \mathcal{N}\left(p \middle| \begin{pmatrix} f_{r,x}(\tau) + b_{r,x}(\tau) \\ f_{r,y}(\tau) + b_{r,y}(\tau) \end{pmatrix}, \Sigma_r(\tau)\right) \mathcal{P}(\tau|\omega_r) \mathrm{d}\tau \end{aligned}$$

Although this generation process can properly model the real generation process of a GPS position, it is not a tractable model for SMM because of three reasons. First, it is difficult



(a) Distribution of τ of different roads (b) Distribution of vertical distances

Figure 5. Some statistical studies of historical data

to use one distribution to approximate the distribution of τ in all road segments. Based on Figure 5(a), we can figure out that the distribution of τ in different road segments differs tremendously. Second, this model includes a latent variable τ and the joint distribution $P(r, p)$ needs the integral of τ which makes the parameter inference hard. Although it can be solved by Expectation-Maximization (EM) algorithm, it needs iterations and the inference will be slow if a large amount of training data is given. Third, it cannot be adopted to online learning which is useful in real applications especially for SMM problem. The detail of online learning will be introduced later.

Our Generative model: PSMM

We then propose Probability-based SMM (in short PSMM), a model that borrows the general idea from the generation process of GPS point discussed above but is modified to be tractable and applicable for SMM problem.

The main change we introduce to PSMM model compared with previous generation process, is a new concept, namely *slot*. It refers to a short line segment along a road segment. In other words, we break a road segment which is represented by a polyline into many short slots. We assume positions along one slot share following common properties because of locality. First, they share common bias and noise. Second, given a large set of GPS points, the density of GPS points at different positions of the same slot is constant. Accordingly, instead of using τ to represent the offset of a GPS point p from the start vertex of a road segment, we use the position τ_s of the head of the slot s and the offset $\Delta\tau$ in the slot s, i.e., $\tau = \tau_s + \Delta\tau$.

In addition, in order to avoid integral, we also make some changes to the generation of the GPS position such that the generated GPS position from the true position becomes tractable. Recall that in the previous generative model, we assume that all the positions on a road segment have certain probability density to generate a given GPS position p. Similarly, it has been assumed that all the positions on a road segment make contribution to the generation of p. For example, for a GPS sample p located in the middle of the road, the position $\tau = 0$ (i.e., the start of the road segment) has the probability density to generate p, although with a relatively low probability, and p's projection on the road can also generate p, however with a relatively high probability. Thus, all the positions on the road have the probability density to generate p which leads to the integral of τ and further contributes to the difficulty of the previous model. It can be observed that each GPS position is generated quite near to its true position. Figure 6 plots an example. For the sake of exponential component in the Gaussian, the probability density for a true position τ on the road exponentially decreases with the increase of the distance between τ and p. Positions near τ_4 call for a high probability to generate p, such as τ_3, τ_4 and τ_5. Moreover, we do not consider the difference between the probability density of τ_3 generating p and that of τ_4 (or τ_5), because τ_3, τ_4 and τ_5 are spatially close to each other. Consequently, we make an assumption that a GPS position is generated *only* by its projection position on the road segment. Back to example shown in Figure 6, we assume p is generated only by τ_4. Consequently, the generation of the GPS position from the true position is now tractable because p is generated only by one τ rather than the infinite $\tau \in [0, 1]$ to avoid integral.

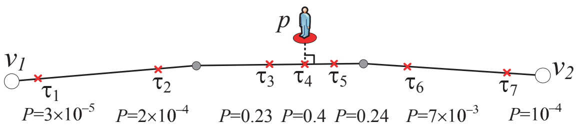


Figure 6. Example of intuitive generation process

Based on the assumption that a GPS point p is generated by its projection position τ on a road segment r, we can use 1-d Gaussian to model the distribution of the distance from p to τ, i.e., the vertical distance from p to r. Figure 5(b) shows a histogram of the projection distances from history GPS samples to the road segment for three randomly selected segments. It demonstrates that the distance between p and its projection on r also follows a near-Gaussian distribution. In other words, the noise and bias of a GPS point can be restricted to the vertical line from p to the slot. From Figure 5(b), we can find out that the mean of the Gaussian is not zero which implies that the bias and the variance are also different on different segments. We use notation δ to denote the *vertical distance* from the GPS position p to the road r, or equivalently speaking, slot s. Thus, the generation of p in the 1-d Gaussian model is equivalent to the generation of δ on $\tau_s + \Delta\tau$.

Incorporating the concept of slot and the proposed 1-d Gaussian distribution for δ, we propose PSMM model, as presented in Figure 7. For the true position $\tau_s + \Delta\tau$, the slot s is drawn from a categorical distribution $C(\zeta_r)$ and then the offset $\Delta\tau$ is generated uniformly in the slot s. For the vertical distance δ, it is drawn from 1-d Gaussian with mean $b_r(\tau_s + \Delta\tau)$ and variance $\sigma_r^2(\tau_s + \Delta\tau)$.

To facilitate the understanding of PSMM model, an example is plotted in Figure 8. We assume each line segment in the figure refers to one slot and the model first selects the slot $s = 4$ in the figure (suppose the index of slot starts from 1). It then decides the in-slot offset $\Delta\tau$. The cross mark represents the true location and the plus signs visualize the distribution of the GPS signals when the true position is at $\tau_s + \Delta\tau$. In this example, we can figure out that the bias is above the segment which is captured by $b_r(\tau_s)$ and the noise level is modelled by $\sigma_r^2(\tau_s)$. Note that we plot a straight road for simplicity but PSMM model works even when the road segment is a polyline with a minor modification in the angle part. Due to the space limitation, we skip the detail.

```
For each road $r$ drawn from $\mathcal{C}(\pi)$:
    For each slot $s$ drawn from $\mathcal{C}(\zeta_r)$:
        For each offset $\Delta\tau$ drawn from $\mathcal{U}(0, \tau_s.len)$:
            Draw projection distance $\delta \sim \mathcal{N}(b_r(\tau_s+\Delta\tau), \sigma^2(\tau_s+\Delta\tau))$;
            Get $p$ starting from $(f_{r,x}(\tau_s+\Delta\tau), f_{r,y}(\tau_s+\Delta\tau))$
            and moving distance $\delta$ along the vertical direction of $s$;
```

Figure 7. Generation process of PSMM model

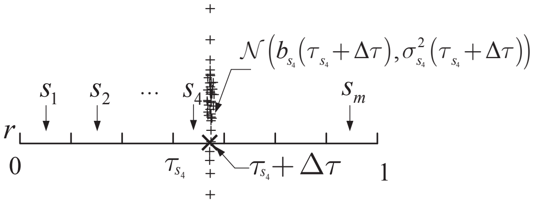

$\mathcal{N}\left(b_{s_4}(\tau_{s_4}+\Delta\tau),\sigma^2_{s_4}(\tau_{s_4}+\Delta\tau)\right)$
s_1
s_2
$\cdots$
s_4
s_m
r
0
τ_{s_4}
$\tau_{s_4}+\Delta\tau$
1

Figure 8. Example of the generation process of PSMM

Under this PSMM model, the joint distribution $P(r, p)$ can be expressed by Equation 1. Note that now the model is free of both latent variable and integral and meanwhile the modelling of $\mathcal{P}(\tau)$ is not necessary.

$$\begin{aligned} P(r, p) &= P(r)P(s|r)P(\Delta\tau|s, r)P(\delta|s, r, \Delta\tau) \\ &= C(r|\pi) \times C(s|\zeta_r) \times \mathcal{U}(\Delta\tau|0, \tau_s.len) \\ &\quad \times \mathcal{N}\left(\delta|b(\tau_s + \Delta\tau), \sigma^2(\tau_s + \Delta\tau)\right) \end{aligned} \tag{1}$$

Parameter Estimation

PSMM model relies on a few of parameters such as π for road segments distribution and ζ_r for slots distribution. In the following, we explain how to determine the values of these parameters. The main idea is to use the log joint likelihood of training samples as the cost function. For the training dataset with m training GPS samples $\mathcal{T} = \left\{\left(p^{(1)}, r^{(1)}\right), \left(p^{(2)}, r^{(2)}\right), \cdots, \left(p^{(m)}, r^{(m)}\right)\right\}$, the log joint likelihood can be computed by $\mathcal{L} = \sum_{i=1}^{m} \log P\left(r^{(i)}, p^{(i)}\right)$. Since we assume the in-slot offset follows the uniform distribution that is only relevant to the length of the slot τ_s and all the positions in the same slot share the same bias and noise, PSMM model needs to decide the values for parameters π, ζ_r, $b_r(\tau_s)$ and $\sigma_r(\tau_s)$. By setting the derivative corresponding to the object parameter to 0, we can get the closed form estimation of that parameter through training data, which are,

$$\begin{aligned} \pi(r) &= \frac{\sum_{i=1}^{m} 1\{r^{(i)} = r\} + 1}{m + |\{r|r \in G.E\}|} \\ \zeta_r(s) &= \frac{\sum_{i=1}^{m} 1\{r^{(i)} = r, \tau_{s^{(i)}} = \tau_s\} + 1}{\sum_{i=1}^{m} 1\{r^{(i)} = r\} + |\{s|s \in r\}|} \\ b_r(\tau_s) &= \frac{\sum_{i=1}^{m} 1\{r^{(i)} = r, \tau_{s^{(i)}} = \tau_s\}\delta^{(i)}}{1\{r^{(i)} = r, \tau_{s^{(i)}} = \tau_s\}} \\ \sigma_r(\tau_s) &= \frac{\sum_{i=1}^{m} 1\{r^{(i)} = r, \tau_{s^{(i)}} = \tau_s\}(\delta^{(i)} - b_r(\tau_s))^2}{1\{r^{(i)} = r, \tau_{s^{(i)}} = \tau_s\}} \end{aligned}$$

where $1\{condition\}$ is the indicator function that returns 1 when condition is true and 0 otherwise. Note that for estimating parameter $\pi(r)$ and $\zeta_r(s)$, Laplace smoothing [17] is adopted because of data sparsity problem which may lead them to be zero.

Considering Large Bias and Misalignment

When a road segment has a large misalignment or the bias is large, directly applying PSMM model may face some issues. This is because PSMM model is based on the assumption that all the positions on the same slot share the same point distribution, same bias and same noise. When the bias is large or the misalignment is severe, this assumption is no longer valid. To address this problem, we adopt a principal curve fitting algorithm [14] to fit the historical GPS points via a polyline. Principal curves are curves that pass through the *middle* of a dataset, providing a nonlinear summary of the data [12]. We use the fitted polyline as the true position of road r, namely *latent road segment* and denoted as $\hat{r}$ in Figure 9. Then, we substitute all the r_is involved in PSMM model by $\hat{r}_i$. Via this strategy, the model will be trained by the latent road segments formed by historical points and hence become irrelevant to the in-correctness of digital map. Note that there is no need for $\hat{r}$ to be located in the middle of the historical points exactly, as our model can capture the biases precisely when the biases are not too large.

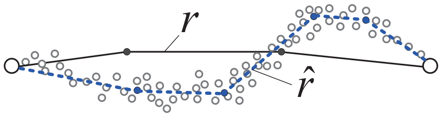


Figure 9. Example of latent road

ONLINE LEARNING

Online learning is used in the case where the data becomes available in a sequential fashion, in order to perform the classification task. In online learning, the mapping between the data point and the labels is updated after the arrival of every new data point in a scalable fashion; whereas traditional offline learning techniques are used when one has access to the entire training dataset at once. Because of the nature of SMM problem, online learning is very applicable. Considering following situation. When a user pushes an SMM request, the model predicts the road segment the user is on and then user can feedback on whether the predication is correct. The model can twist the parameters based on the user feedback to further improve the system. This process can continue which is exact the same as online learning.

The main benefits of online learning for SMM lay on three aspects. First, the biases, noises and the road shapes might change over the time. Thus, online learning provides a chance for the model to adjust the parameters in order to cater for new changes. Second, the proposed model relies on a large sample set to decide the proper values of parameters and the size of the sample set has a direct impact on the quality of the parameters and hence the performance of the model. However, in some application scenarios, the amount of samples available is limited. Online learning provides an alternative to gradually improve the accuracy of the predication when more and more samples and feedback are collected. Third, online learning has much lower memory requirements in the sense that it only requires storage of the current parameters of the model and the next data point (p, r). This is because online learning will incrementally update the parameter with the prediction result of the next data point. For PSMM, when the ith feedback sample $\left(r^{(i)} = r, p^{(i)}\right)$ arrives, we use superscript of the parameter, e.g., $\pi^i(r)$, to indicate the adjusted parameter after seeing the ith feedback. As our parameter can be estimated in the closed form, it is not hard to get the update criteria, as shown in Equation (2), where $c_r = |\{r|r \in G.E\}|$, $q_{r,s} = |\{s|s \in r\}|$, $N^{i-1} = i - 1$, $N_r^{i-1} = \sum_{k=1}^{i-1} 1\{r^{(k)} = r\}$, $N_{r,s}^{i-1} = \sum_{k=1}^{i-1} 1\{r^{(k)} = r, \tau_{s^{(k)}} = \tau_s\}$. Note that the update of all the components has a constant time complexity.

$$\pi^i(r) = \frac{N_r^{i-1} + 2}{N^{i-1} + 1 + c_r}, \;\; \zeta_r^i(s) = \frac{N_{r,s}^{i-1} + 2}{N_r^{i-1} + 1 + q_{s,r}}$$

$$b_r^i(\tau_s) = \frac{1}{N_{r,s}^{i-1} + 1} \left(N_{r,s}^{i-1} b_r^i(\tau_s) + \delta^{(i)} \right)$$

$$\sigma_r^i(\tau_s) = \frac{N_{r,s}^{i-1}}{N_{r,s}^{i-1} + 1} \left(\left(b_r^{i-1}(\tau_s) \right)^2 - \left(\sigma_r^{i-1}(\tau_s) \right)^2 \right) + \frac{\left(\delta^{(i)} - b_r^i(\tau_s) \right)}{N_{r,s}^{i-1} + 1}$$
$$+ \frac{N_{r,s}^{i-1}}{\left(N_{r,s}^{i-1} + 1 \right)^3} b_r^i(\tau_s) \left(b_r^i(\tau_s) - 2 \left(N_{r,s}^{i-1} - 1 \right) \right) \qquad (2)$$

SOLUTION TO SMMD PROBLEM

In this section, we will show how to plug the SMM solution to the problem with additional input information when locating the GPS position to the road network. Generally, let $\mathcal{I}$ denote other information available. Given input $\{p, \mathcal{I}\}$, locating the real road segment r^* is also equivalent to find the road segment r that can maximize the posterior, i.e, $P(r|p, \mathcal{I})$. By Bayes' theorem, $P(r|p, \mathcal{I}) \propto P(p, \mathcal{I}|r)P(r) = P(p|r)P(\mathcal{I}|r, p)P(r) = P(p, r)P(\mathcal{I}|r, p)$. Recall that $P(p, r)$ is the joint probability in SMM problem which can be derived by Equation (1). That is to say our PSMM model can be plugged here without any modification and the only remaining task is to model the component $P(\mathcal{I}|r, p)$.

Specifically, in taxi-hailing service, the destination where a user wants to go is usually available, i.e., $\mathcal{I} = d$. To differentiate this from the original SMM problem where the destination d is not available, we name the problem of locating a user currently located at a GPS position p to a road segment given the destination d the user wants to go as *SMMD*, as presented in Problem 2 in Preliminary Section. In the following, we introduce how to model $P(d|r, p)$.

Note that, once r is given, the generation of d is irrelevant to the GPS position p since p is also generated by r, thus $P(d|r, p) = P(d|r)$. Intuitively, we can use categorical distribution to model $P(d|r)$ since d is a discrete variable, i.e., d is drawn from $C(\Phi_r)$ and the probability of choosing the destination d given r is denoted by $\Phi_r(d)$. Through maximum likelihood estimation (MLE), $\Phi_r(d)$ can be estimated by $\frac{|\{T|T.s=r \wedge T.e=d\}|}{|\{T|T.s=r\}|}$, where $|\{T|T.s = r \wedge T.e = d\}|$ refers to the number of trips that are from road segment r to d, and $|\{T|T.s = r\}|$ refers to the number of trips started from road segment r. However, this naive method suffers from a data sparsity problem. We find that given certain road r, most value of $|\{T|T.s = r \wedge T.e = d\}|$ is 0 for $d \in G.V$. Thus, the destination information will be eventually useless since for all the destinations with $|\{T|T.s = r \wedge T.e = d\}| = 0$, $\Phi_r(d)$ will be assigned an identical value. Motivated by this, we intend to propose *Probability-based SMMD* (in short PSMMD) to model $P(d|r)$ which does not suffer from the data sparsity problem.

Note that $P(d|r)$ is the probability of the event that the user wants to go to d if he/she is standing at road r. Intuitively, the direction of the candidate road segment w.r.t. d (e.g., whether it is towards the destination or not) may affect user's choice of road segments. As a result, we first introduce the *cosine direction angle* $\varphi(r, d)$, as shown in Equation (3), to quantify how a road segment r is towards to the destination and then show how it can be transferred to model $P(d|r, p)$.

$$\varphi(r, d) = \cos(\theta(r, d)) = \frac{\langle r.e - r.s, d - r.e \rangle}{|r.e - r.s| \cdot |d - r.e|} \qquad (3)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product between two vectors and $|\cdot|$ infers the norm of a vector.

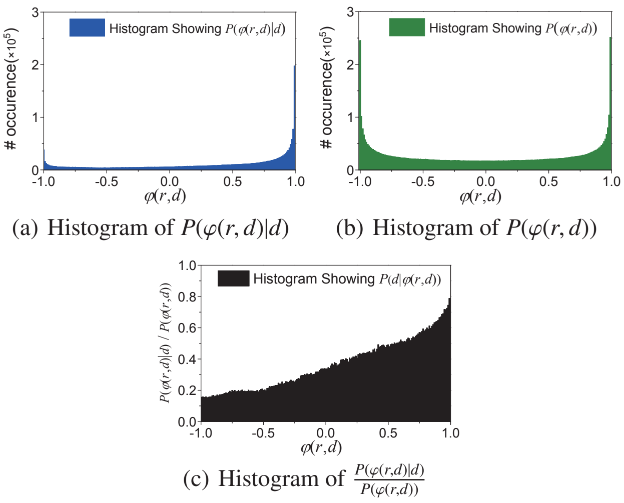
We intend to model $P(d|r)$ based on $P(d|\varphi(r, d))$. The reason is that, from the above intuition we can find that given a destination d and two road segments r_1 and r_2 such that $\varphi(r_1, d) \approx \varphi(r_2, d)$, the choice of road segments made by users on r_1 and that made by users on r_2 tend to be similar. Under this observation, we cluster (r, d) pairs with similar $\varphi(r, d)$ values together and assume they have the same selection probability. That is to say, $\varphi(r_1, d_1) = \varphi(r_2, d_2) \rightarrow P(d_1|\varphi(r_1, d_1)) = P(d_2|\varphi(r_2, d_2))$. Obviously, this can address the data sparsity problem since those (r, d) pairs with the same cosine value φ will be considered together. By Bayes' theorem, $P(d|\varphi(r, d))$ can be transformed as shown in Equation (4). This means that $P(d|\varphi(r, d))$ becomes tractable because $P(\varphi(r, d)|d)$ and $P(\varphi(r, d))$ can be statistically estimated. In the following, we explain how to derive $P(\varphi(r, d)|d)$ and $P(\varphi(r, d))$.

$$P(d|r) = P(d|\varphi(r, d)) = \frac{P(\varphi(r, d)|d)P(d)}{P(\varphi(r, d))} \propto \frac{P(\varphi(r, d)|d)}{P(\varphi(r, d))} \quad (4)$$

To estimate $P(\varphi(r, d)|d)$, a simple method is to draw the histogram of $\varphi(r, d)$ using all trips that are ended in d. Note that for those destinations d and d' that are spatially close, trips ended in d' can also be used in computing the histogram of $\varphi(r, d)$ since spatially close destinations are expected to share similar distribution of $\varphi(r, d)$. We partition the whole map into grids and assume the destinations in the same grid share the same distributions. Figure 10(a) shows an example histogram of $P(\varphi(r, d)|d)$ w.r.t. a grid located in the middle of the map.

To estimate $P(\varphi(r, d))$, we draw the histogram by traversing the whole map. As shown in Figure 10(b), the distribution of $\varphi(r, d)$ is nearly symmetric with 0.0 which is consistent with our expectation because the grid we choose is in the central of Singapore. Thus, the distribution of the cosine angle formed by the roads on map and grid g is symmetric. In addition, the density of $\varphi(r, d) = -1.0$ and $\varphi(r, d) = 1.0$, or equivalently speaking, $\theta(r, d) = 0$ and $\theta(r, d) = \pi$, looks much larger than the middle ones. This is because the mainland of Singapore measures 50 kilometers from east to west and 26 kilometers from north to south, in a rectangular shape with length longer than the breadth.

After plotting the histograms of $P(\varphi(r, d)|d)$ and $P(\varphi(r, d))$, we can divide them corresponding to each interval of $\varphi(r, d)$ to get the histogram of $P(d|\varphi(r, d))$. Figure 10(c) shows the result by dividing the histogram of Figure 10(a) and that of Figure 10(b). The distribution is consistent with our expectation. $\varphi(r, d)$ increases from -1.0 to 1.0 (i.e., $\theta(r, d)$ increases from $-\pi$ to π) monotonously, which means the direction of road r is increasingly facing towards the destination and the probability a person tends to choose r is also increased. Based on the distribution of $P(d|\varphi(r, d))$, we use exponential distribution to model it and finally the modelling of $P(d|r, d)$ is addressed. Note that we do not intend to discuss other input information



(a) Histogram of $P(\varphi(r,d)|d)$ (b) Histogram of $P(\varphi(r,d))$

(c) Histogram of $\frac{P(\varphi(r,d)|d)}{P(\varphi(r,d))}$

Figure 10. Example histograms

such as temporal information due to the limitation of space and it can be easily taken into consideration under our solution framework.

EXPERIMENT

Dataset. We conduct a comprehensive experimental study based on real trajectory dataset from Singapore. The dataset contains 225,000 trajectories generated by about 15,000 taxis from Jan. 1, 2011 to Jan. 15, 2011. The average sampling interval of the trajectories is 30s. Please refer to [2] for a detailed description of the dataset. The digital map we use is from OpenStreetMap[1]. We conduct the experiments in an area consisting of 577 road segments. The reason that we do not use the full Singapore area is that some existing classifiers, e.g., ANN and SR, are not able to get the answer if we consider the whole area. On the other hand, we do not observe any significant performance drops of our solution when we increase the area to the full landscape of Singapore. In addition, SMMD problem only restricts the origin of the trip to be located within the area but not the destination.

For SMM, 10,000 GPS samples picked randomly from the trajectories serve as test set. The ground truth is obtained by the strategy introduced in Obtaining the Ground Truth Section. For SMMD, we select the trajectory with BUSY status that corresponds to a taxi journey and generate the dataset containing the information of the pick-up position, drop-off position, and the complete trajectory. The pick-up as well as drop-off positions are input for SMMD, and the complete trajectory is used for getting the ground truth of the road segment w.r.t. the pick-up position. For SMMD problem, in total 10,000 records are tested. In the following, we present the competitors of PSMM and PSMMD models.

Classification Algorithms. As SMM/SMMD problem can be formalized into a multi-class classification problem, we implement representative classification algorithms as competitors, including softmax regression (SR) [4], Naive Bayes (NB) [4], support vector machine with linear kernel (SVM$_{linear}$) [4] and that with radial basis function (SVM$_{RBF}$) kernel, artificial neural network with one hidden layer (ANN) [4], decision tree (DT) [21] and k-nearest neighbor classification (kNN) [4]. For ANN, the number of nodes in the hidden layer is set to 50. For SVMs, the multi-class classification is implemented by "one-against-one" approach; and for SVM$_{RBF}$, the parameter γ is set to 1.0. For DT, the split criteria is Gini impurity [21]. For kNN, we set $k = 20$. We use the latitude and longitude of the GPS position as the input feature. We would like to highlight that, in addition to the position information, we have tried to include other features such as the vertical distance from the GPS point to each road segment to make the input feature more informative. However, the results do not differ too much from the original ones where we only consider GPS point. Because the high dimension of the input feature will vastly increase the training time of some classifiers, we only consider GPS point feature in our experiments.

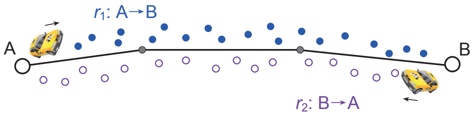


Figure 11. GPS samples on a bidirectional road segment

Simple Baseline Approach (SBA). Given a position p, SBA locates p to the *nearest* road segment r. Given the fact that a bidirectional road is represented by two segments with exactly the same shape but opposite directions, two segments might share the same shortest distance to p. If this is the case, one of them is returned randomly with the same probability.

Baseline Approach (BA). Although a bidirectional road is represented by two road segments with the same shape, in reality, the width of bidirectional road segments is actually *non-negligible*. Intuitively, the GPS samples on r tend to lay on the right (left) side of r in countries with right-hand (left-hand) traffic. Accordingly, in countries with right-hand (left-hand) traffic, BA returns the road segment r of which p is on the right (left) side. As shown in Figure 11, for all the solid points that are on the left side of r_1 in Singapore, r_1 will be reported as the answer. Similarly, for hollow points, r_2 will be returned.

Baseline Approach with Destination (BAD). BAD is designed as a baseline for SMMD. Similar to SBA/BA, it returns the nearest road of p. However, when the nearest road is a bidirectional road, e.g., $\{r_1, r_2\}$, it returns the segment that is towards the destination, i.e., $r^* = \arg\max_{r \in \{r_1, r_2\}} \varphi(r,d)$.

Comparison of Accuracy

We first evaluate the accuracy of different approaches that can support SMM or SMMD problems. We report the *accuracy*, i.e., the ratio of the number of correctly-matched test samples to the total number of test samples, in Table 1.

From the result, we observe that although SMM/SMMD can be transferred into a classification problem, most of traditional classification algorithms do not work well. Among the classification algorithms we implement, ANN and SR suffer from low accuracy. They both classify the objects by using a function to approximate the posterior probability of each class. In the training step, they both want to find a function to maximize the likelihood (SR) or minimize the prediction error (ANN) over the whole training dataset. When the number of classes is large (e.g., 577), it is hard to find such a ’function’ to model the data which leads to the poor performance.

[1] http://www.openstreetmap.org/

	SMM	SMMD
ANN	3.17%	3.63%
SR	15.13%	17.23%
NB	50.77%	53.8%
SVM$_{linear}$	44.73%	45.21%
SVM$_{RBF}$	50.51%	50.89%
DT	54.53%	57.2%
*k*NN	62.08%	62.81%
SBA	44.20%	39.51%
BA	59.97%	66.99%
BAD	N/A	52.14%
PSMM	**76.52**%	77.59%
PSMMD	N/A	**82.58**%

Table 1. Comparison of Accuracy

NB performs better than SR and ANN. This is because NB is a generative model, which can capture more information such as the prior $P(r)$. SVM$_{linear}$ can only classify data via a hyperplane or line. In SMM, the distributions of historical GPS data on different road segments are not linear, which affects the accuracy. On the other hand, although the distribution of historical points can not be linearly separated, they are actually near-linear since the shape of a road segment does not twist heavily. When adopting SVM$_{RBF}$, the feature can be mapped into infinite kernel space which allows the data to be classified linearly in the high-dimension kernel space. Thus, SVM$_{RBF}$ improves the accuracy, as compared with SVM$_{linear}$.

DT performs a little bit better. The main reason is that DT uses horizontal and vertical boundaries to split the data and hence the performance of DT is not affected by the large number of classes. Because DT boundaries posed on the data are similar to the *k*-d tree's boundaries to sub-divide the dataset, for a bidirectional road, the decision boundary will be greatly affected by the road shape, which may be the key reason why it is inferior to *k*NN. *k*NN performs best out of all the traditional classification approaches. The main reason is that *k*NN classifies the data by returning the label which has maximum votes of its nearest neighbors. For the SMM/SMMD problem, the historical GPS points on the same road segment tend to be clustered together, which implicitly takes the bias and noise into consideration. However, as *k*NN highly depends on the dense distribution of the GPS points, the volume of historical data required by *k*NN $(400,000$ points) is much larger than that of other approaches $(< 100,000$ points).

SBA generates the accuracy slightly below 50%, which is below our expectation. If each road is bidirectional and the matched road r is nearest to p, the accuracy shall be exactly 50%. We guess the main reason that SBA has accuracy below 50% is that the GPS has bias and noise, and some matched road segments may not be the ones with the minimum distance to p. Actually, in this dataset, about 15.3% ground truths are not nearest to the GPS position. BA achieves a higher accuracy, as compared with SBA. This means the driving direction does provide useful information for SMM and SMMD problems. We also notice that BAD outperforms SBA but not BA. This means the destination indeed has an impact on the selection of road segments, but its impact is smaller than that of the distribution of points on a road segment.

PSMM and PSMMD models outperform all the existing approaches with significant improvement. An interesting observation is that most approaches perform better when the destination is provided, although the improvement of PSMM model is less significant. This is mainly caused by the difference among the datasets. Given a testing dataset, most of the approaches tend to fit the data of those dense areas. As the testing data is drawn from the same distribution of training data, the testing samples in dense areas will be predicted correctly which explains the improvement of the accuracy for most approaches. However, PSMM model focuses on the generation of a GPS point. This means whether the point is located in a dense area or a sparse area, it does not affect the way PSMM locates the road segment. In other words, the distribution of historical and testing data will not influence the performance of PSMM model.

	ANN	SR	NB	SVM	DT	PSMM
time t	312s	36.8s	2.2s	303s	1.6s	1.0s
training count m	30,000	50,000	90,000	50,000	70,000	80,000
$\frac{t}{m}$(unit: $10^{-5} s$)	1040	73.6	2.44	606	2.29	**1.25**

Table 2. Training Time

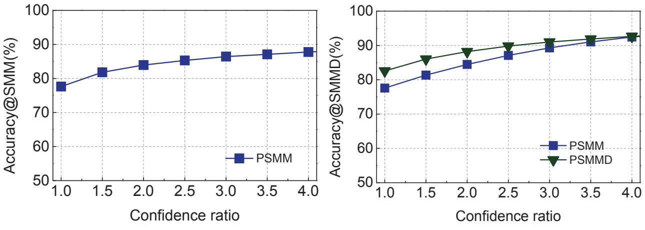
Training Time Comparison

In the second set of experiments, we report the training time consumed by different models in Table 2. Note that *k*NN, SBA, BA and BAD are excluded as *k*NN is a non-parametric method and SBA, BA as well as BAD are not classifiers. Thus, they do not require training process. Considering that different classifiers need different number of training samples, we train each classifier using different sizes of training samples which allow them to perform nearly best. We also include the loading time of training files. As PSMM model only needs to read the training data and can get the answer analytically, it requires very short training time, close to NB and DT which have no iterations to adjust the model parameters. Note that SVM and SR need iterations for training each road (such as SGD [4] and SMO [23] algorithms in their implementation), which extend the training time. ANN uses back-propagation to update the parameters, and hence incurs the longest training time.

Confidence Test

In the third set of experiments, we report the performance of our approaches under different confidence ratios. In our work, notations $p(r^{1st}|\chi)$ and $p(r^{2nd}|\chi)$ refer to the highest and second highest posterior probabilities of the candidate road segments respectively, computed by PSMM or PSMMD. Here, χ refers to the input information, i.e., p for SMM or (p, d) for SMMD. We define the *confidence ratio* of χ as $\rho(\chi) = p(r^{1st}|\chi)/p(r^{2nd}|\chi)$. Obviously, $\rho(\chi) \in [1, +\infty)$.

To study the performance under different confidence ratios, we generate the testing dataset by excluding certain testing samples from the testing dataset based on ρ values. To be more specific, for a given ρ_0, all the data $\chi^{(i)}$ with $\rho(\chi^{(i)}) < \rho_0$ will be excluded from the testing dataset. We then conduct experiments under this refined dataset and report its accuracy corresponding to ρ_0. In detail, we conduct this experiment by varying ρ_0 from 1.0 to 4.0. The reason to conduct such an experiment is that the accuracy under certain confidential ratio ρ_0 can be regarded as the *confidence* of our approach's result for a testing sample χ if $\rho(\chi) \geq \rho_0$. It is not hard to find that the accuracy under $\rho_0 = 1$ is the accuracy on the complete testing dataset without filtering any testing sample.



(a) Accuracy@SMM (b) Accuracy@SMMD

Figure 12. Accuracy vs. confidence ratio

First, we plot the accuracy of PSMM model in different confidence ratios under SMM dataset in Figure 12(a). We can observe that with the increase of ρ_0, the accuracy is improved which is consistent with our expectation. This is because the higher the ρ is, the larger the gap between $p(r^{1st}|\tau)$ and $p(r^{2nd}|\tau)$ will be. Consequently, the probability of returning the right answer r^{1st} becomes higher.

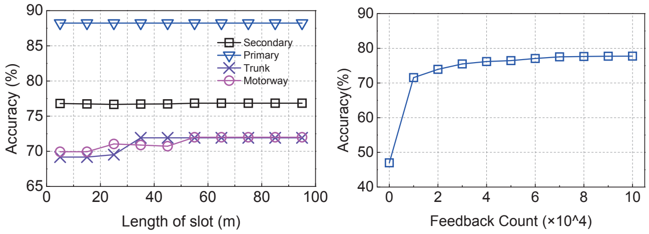
Next, we plot the accuracy of both PSMM and PSMMD for supporting SMMD problem under SMMD dataset in Figure 12(b). As PSMMD incorporates more information than PSMM, the accuracy is always superior to PSMM. We can observe that when the confidence ratio is larger than 2.5, the accuracy of PSMMD exceeds 90%. In addition, when the confidence ratio ρ increases, the gap between PSMMD and PSMM shrinks. The reason is that as ρ increases, $P(p, r)$ contributes more to the accuracy, as compared with $P(d|r)$ to the accuracy. Recall that although the destination has an influence on the decision of the road segment, the influence is still smaller than the distribution of historical points as the performance of BAD is inferior to BA. That explains why as ρ increases, the difference between these two algorithms becomes smaller. As the performance of PSMM and that of PSMMD are similar, when we perform further studies on PSMM model to be reported below, we only plot the result of PSMM model.

Road Type and Size of Slot

In this set of experiments, we evaluate the impact of slot size in different road types. We evaluate four road types including motorway, trunk way, primary way and secondary way, which cover most of the traffic. Note that the type of road is obtained from OpenStreetMap. The accuracy of our model under different slot sizes and road types is reported in Figure 13(a). We observe that motorway and trunk way have the lowest accuracy among these four types. The reason is that motorway and trunk way are both express ways which have many lanes close to each other, which leads to the difficulty of locating a GPS position to a lane accurately. Note that the size of slots does not have a significant impact on the result corresponding to primary and secondary ways. The reason is that most of these ways in the testing area are straight lines and the bias tends to be consistent in these ways. For motorway, 55m is the best setting for the slot size. For trunk way, it is 35m. For primary way, the slot size does not affect the accuracy so much and for secondary way, 5m is the best.

Online Learning

We also conduct an experiment to demonstrate the effectiveness of online learning. Initially, the parameters of PSMM model are set to their defaults i.e., $b_r(\tau) = 0$, $\sigma_r(\tau) = 1$, $\pi(r_1) = \pi(r_2) = \cdots$, and $\zeta_r(s_1) = \zeta_r(s_2) = \cdots$. According to Equation (1), $P(r, p) = C(r|\pi) \times C(s|\zeta_r) \times \mathcal{U}(\Delta\tau|0, \tau_{s+1} - \tau_s) \times \mathcal{N}\left(\delta|b(\tau_s), \sigma^2(\tau_s)\right)$. As the parameters π and ζ_r of categorial distributions are all the same, $C(r|\pi)$ and $C(s|\zeta_r)$ can be regarded as constants. Given the fact that the uniform distribution is constant, the joint probability is only influenced by the Gaussian. As all the slots share the same default b and σ, PSMM without any training performs as SBA does and the result proves above analysis.



(a) Accuracy vs. slot length (b) Accuracy vs. feedback count

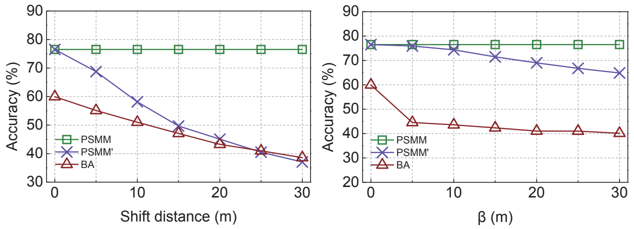
Figure 13. Accuracy v.s. slot size and performance via online learning

We start collecting the accuracy data when PSMM model has received 10,000 feedback, with the average density being $10,000/577 \approx 17$ feedbacks per road segment. The result is plotted in Figure 13(b). We can observe that PSMM is able to learn in a very fast speed. For example, when there are 17 feedbacks per road, the accuracy has already exceeded 70%. This shows that PSMM model does not suffer from the cold start problem and it can perform well after receiving a small number of feedbacks. When more feedbacks are received, the accuracy of PSMM improves and it is able to converge quickly.

Robustness Test

The sixth set of experiments is to demonstrate the robustness of our approach. Recall that PSMM generates a latent road segment based on the distribution of historical points to tackle the issue of large bias and misalignment of map. Hence the correctness of map does not affect its performance. However, the baseline algorithms, including BA, SBA and BAD, locate the answer road segment based on the position of p and its distances to different road segments. In other words, the accuracy of the digital map will affect the performance of those baseline algorithms. We also include PSMM model without using latent road segment, denoted as PSMM', to study the potential impact caused by the *map errors* on the accuracy. The performance of classification algorithms is excluded from this set of experiments because they train their models without considering the digital map and their result will not be affected by the quality of digital maps. As BA performs best among BA, SBA and BAD, we include BA as the representative of them.

The first type of in-correctness we introduce to the maps is map shifting. The shifting distance is varied in both horizontal and vertical directions. It is observed from Figure 14(a) that PSMM retains high accuracy under various shifting distances as PSMM relies on the latent road which is independent of the correctness of the digital map. PSMM' suffers from the shifting of the map as the GPS points are assigned to wrong slots which does not allow the generative model to work. Obviously, this finding well demonstrates the effectiveness of the latent road strategy. BA is very sensitive to the shifting distance too



(a) GPS point shift (b) Perturbation

Figure 14. Accuracy vs. digital map errors

because when map is shifted, the nearest road (unidirectional or bidirectional) may not contain the ground truth.

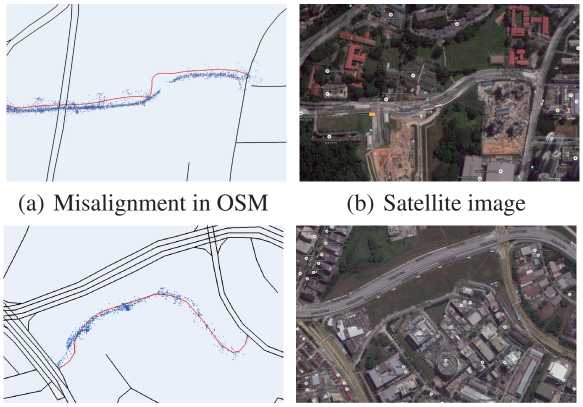
The second type of in-correctness we introduce to the maps is via perturbation, which simulates the situation of map misalignment. Parameter β is introduced as the perturbation coefficient. For each intermediate point $p_i \in r.shape$, it is moved by a distance drawn from $\mathcal{U}(-\beta,\beta)$, in both vertical and horizontal directions. The accuracy of different approaches under various β values is plotted in Figure 14(b). As β increases from 0 to 5m, the accuracy of BA reduces drastically, from nearly 60% to 45%. When β exceeds 5m, the trend of reduction of BA becomes gentle. This is because when the road shape is wrong, BA can not leverage the left/right-handed driving characteristic to determine the point on a bidirectional road. Thus, BA will reduce to SBA as it chooses the segment almost randomly as the shape of the road is wrong. The accuracy of PSMM' also decreases with the increase of β, which is consistent with our expectation as the misalignment of roads will lead to the difficulty of capturing the biases and noises. Recall that our model assumes each position on the same slot shares the same bias and noise and the in-correctness of the roads will violate this assumption when latent road is not used.

Finding Misalignment Roads

In the last set of experiments, we want to demonstrate that our model is also able to find out the misalignment roads. Recall that before training the model, generating the latent road $\hat{r}$ is useful for the accuracy of SMM/SMMD. Here, we make a small change, instead of training the exact PSMM, we train the PSMM' introduced in the previous section. After training the parameters, if a certain road segment r has several consecutive slots with parameter $b_r(\tau_s)$ larger than others, it is very likely that the segment r in the digital map is misaligned. By figuring out the roads with large bs, it is possible to help the map producer to find out the misaligned road. Figure 15 shows two roads with the largest bs. Figure 15(a) and Figure 15(c) plot the historical GPS points on the roads. From these two figures, we can observe that the distribution of those points is inconsistent with the shape of the road. By checking the satellite images from Google Map, i.e., Figure 15(b) and Figure 15(d), it can be proved that these two roads found out by our model are *actually misaligned* in the digital map (OpenStreetMap). This demonstrates that our model can improve the quality of the digital map especially non-commercial maps which are more likely to have more misalignments than commercial maps.

RELATED WORK

There is a series of research on map matching for *trajectories*. [3, 22, 32] map each GPS point with geometry information which often have low accuracy. [18] utilizes the Kalman Filter to map the trajectory to the map which is able to correct the GPS position according to context. HMM-based methods not only consider the context information but also the topology of the map and they can achieve a considerably high performance under high-sampling rate [20, 26, 31]. On the other hand, there are several research works tackling low-sampling rate trajectory data [15, 35] which are also based on the HMM framework. Since low-sampling rate trajectory data have lost a great deal of information, the accuracy is relatively low. [5] studies the effect of sensor errors in map matching and analyzes the percentage of many types of errors.



(a) Misalignment in OSM (b) Satellite image

(c) Misalignment in OSM (d) Satellite image

Figure 15. Examples of misalignment roads

Although there is no research directly addressing the SMM problem, there are some hardware-related approaches on reducing the error of GPS signals. [9] uses wavelet to remove the low to high-frequency GPS errors, where the low frequency error includes bias resulted by multipath, ionospheric and tropospheric delays, and the high frequency error refers to the random measurement error. Besides, [11, 28] try to reduce the bias generated by multipath. [27] also aims to mitigate the effect of multipath, but it is designed for trajectory rather than a single point. [13] corrects the bias for a vehicle through camera images by the vision system. We claim that our approach is the first try via a data-driven view and can also be adopted after applying these hardware-based approaches.

CONCLUSION

In this paper, we have studied a ubiquitous problem, i.e., SMM with historical data and proposed a generative model PSMM which is the first attempt. Our model is carefully designed so that it is able to model the fixed bias as well as random noises properly with a closed form solution which makes it easy to train and can be adopted to online learning. We also extend the SMM problem to SMMD problem to show how to plug our PSMM model into real applications with other information available. We conduct experiments using real-world dataset and the results validate the effectiveness and robustness of our models, as compared with existing classifiers and three baseline approaches. Our models can achieve the confidence over 90% when the confidential ratio is larger than 3.0.

ACKNOWLEDGEMENTS

This research is supported in part by National Natural Science Foundation of China (NSFC) under grant 61073001, Shanghai Natural Science Foundation under grant 14ZR1403100. Weiwei Sun is the corresponding author.

REFERENCES

1. Mohamed Ali, John Krumm, Travis Rautman, and Ankur Teredesai. 2012. ACM SIGSPATIAL GIS Cup 2012. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS'12)*. ACM, 597–600.

2. Rajesh Krishna Balan, Khoa Xuan Nguyen, and Lingxiao Jiang. 2011. Real-time Trip Information Service for a Large Taxi Fleet. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (Mobisys'11)*. ACM, 99–112.

3. David Bernstein and Alain L. Kornhauser. 1998. An Introduction to Map Matching for Personal Navigation Assistants. (1998).

4. Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

5. Wu Chen, Zhilin Li, Meng Yu, and Yongqi Chen. 2005. Effects of Sensor Errors on the Performance of Map Matching. *Journal of Navigation* 58, 02 (2005), 273–282.

6. Blerim Cici, Athina Markopoulou, Enrique Frias-Martinez, and Nikolaos Laoutaris. 2014. Assessing the Potential of Ride-sharing Using Mobile and Social Data: a Tale of Four Cities. In *Proceedings of the 2014 International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp'14)*. ACM, 201–211.

7. J. L. Cuevas-Ruiz, A. Aragón-Zavala, G. A. Medina-Acosta, and J.A. Delgado-Penin. 2009. Multipath Propagation Model for High Altitude Platform (HAP) Based on Circular Straight Cone Geometry. In *International Workshop on Satellite and Space Communications, 2009*. IEEE, 235–239.

8. K. R. Desai, P. T. Patil, R. H. Chile, and S. R. Sawant. 2015. A Model for Detection of Error Factors in GPS Signals. *SSRG International Journal of Electronics and Communication Engineering* 2, 2 (2015).

9. A. A. El-Ghazouly, M. M. Elhabiby, and N. M. El-Sheimy. 2013. Medium to High-frequency Static DGPS Error Reduction Using Multi-resolution De-noising vs. De-trending Procedures. *Journal of Geodetic Science* 3, 3 (2013), 224–239.

10. Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. 2015. CityMomentum: an Online Approach for Crowd Behavior Prediction at a Citywide Level. In *Proceedings of the 2015 International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp'15)*. ACM, 559–569.

11. Ronald L. Fante and John J. Vaccaro. 2003. Evaluation & Reduction of Multipath-induced Bias on GPS Time-of-arrival. *IEEE Trans. Aerospace Electron. Systems* 39, 3 (2003), 911–920.

12. Trevor Hastie and Werner Stuetzle. 1989. Principal Curves. *J. Amer. Statist. Assoc.* 84, 406 (1989), 502–516.

13. Kichun Jo, Keounyup Chu, and Myoungho Sunwoo. 2013. GPS-bias Correction for Precise Localization of Autonomous Vehicles. In *Proceedings of 2013 Intelligent Vehicles Symposium (IV'13)*. IEEE, 636–641.

14. Balázs Kégl, Adam Krzyzak, Tamás Linder, and Kenneth Zeger. 2000. Learning and Design of Principal Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 3 (2000), 281–297.

15. Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. 2009. Map-matching for Low-sampling-rate GPS Trajectories. In *Proceedings of the 17th International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS'09)*. ACM, 352–361.

16. Shuo Ma, Yu Zheng, and Ouri Wolfson. 2013. T-share: A large-scale Dynamic Taxi Ridesharing Service. In *Proceedings of the 29th International Conference on Data Engineering (ICDE'13)*. IEEE, 410–421.

17. Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, and others. 2008. *Introduction to Information Retrieval*. Vol. 1. Cambridge University Press.

18. Maan E. El Najjar and Philippe Bonnifait. 2005. A Road-Matching Method for Precise Vehicle Localization Using Belief Theory and Kalman Filtering. *Auton. Robots* 19, 2 (2005), 173–191.

19. Van Nee and Richard DJ. 1993. Spread-spectrum Code and Carrier Synchronization Errors Caused by Multipath and Interference. *IEEE Trans. Aerospace Electron. Systems* 29, 4 (1993), 1359–1365.

20. Paul Newson and John Krumm. 2009. Hidden Markov Map Matching Through Noise and Sparseness. In *Proceedings of the 17th International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS'09)*. ACM, 336–343.

21. Tan Pang-Ning, Michael Steinbach, and Vipin Kumar. 2006. *Introduction to Data Mining*. Vol. 1. Pearson Addison Wesley Boston.

22. Phuyal and P. Bishnu. 2002. Method and use of aggregated dead reckoning sensor and GPS data for map matching. In *Proceedings of the 15th Institute of Navigation-GPS Annual Conference (ION-GPS'02)*. 430–437.

23. John Platt. 1998. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Technical Report MSR-TR-98-14* (1998).

24. Zhangqing Shan, Hao Wu, Weiwei Sun, and Baihua Zheng. 2015. COBWEB: a Robust Map Update System Using GPS Trajectories. In *Proceedings of the 2015 International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp'15)*. ACM, 927–937.

25. Masamichi Shimosaka, Keisuke Maeda, Takeshi Tsukiji, and Kota Tsubouchi. 2015. Forecasting Urban Dynamics with Mobility Logs by Bilinear Poisson Regression. In *Proceedings of the 2015 International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp'15)*. ACM, 535–546.

26. Renchu Song, Wei Lu, Weiwei Sun, Yan Huang, and Chunan Chen. 2012. Quick Map Matching Using Multi-core CPUs. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS'12)*. ACM, 605–608.

27. Niko Sünderhauf, Marcus Obst, Gerd Wanielik, and Peter Protzel. 2012. Multipath Mitigation in GNSS-based Localization Using Robust Optimization. In *Proceedings of 2012 Intelligent Vehicles Symposium (IV'12)*. IEEE, 784–789.

28. Bryan R. Townsend and Patrick C. Fenton. 1994. A Practical Approach to the Reduction of Pseudorange Multipath Errors in a L1 GPS Receiver. In *Proceedings of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation(ION GNSS'13)*. 143–148.

29. Yin Wang, Xuemei Liu, Hong Wei, George Forman, Chao Chen, and Yanmin Zhu. 2013. CrowdAtlas: self-updating maps for cloud and personal use. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (Mobisys'13)*. ACM, 27–40.

30. Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. 2015. Regularity and Conformity: Location Prediction Using Heterogeneous Mobility Data. In *Proceedings of the 21th SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'15)*. ACM, 1275–1284.

31. Hong Wei, Yin Wang, George Forman, Yanmin Zhu, and Haibing Guan. 2012. Fast Viterbi Map Matching with Tunable Weight Functions. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS'12)*. 613–616.

32. Christopher E White, David Bernstein, and Alain L. Kornhauser. 2000. Some Map Matching Algorithms for Personal Navigation Assistants. *Transportation Research Part C: Emerging Technologies* 8, 1 (2000), 91–108.

33. Hao Wu, Chuanchuan Tu, Weiwei Sun, Baihua Zheng, Hao Su, and Wei Wang. 2015. GLUE: a Parameter-Tuning-Free Map Updating System. In *Proceedings of the 24th International on Conference on Information and Knowledge Management (CIKM'15)*. ACM, 683–692.

34. Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering Regions of Different Functions in a City Using Human Mobility and POIs. In *Proceedings of the 18th SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'12)*. ACM, 186–194.

35. Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guangzhong Sun. 2010. An Interactive-voting Based Map Matching Algorithm. In *Proceedings of the 11th International Conference on Mobile Data Management (MDM'10)*. IEEE, 43–52.

36. Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2015. Discovering Urban Functional Zones Using Latent Activity Trajectories. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2015), 712–725.

37. Yu Zheng. 2015. Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology* 6, 3 (2015), 29:1–29:41.

38. Yu Zheng and Xiaofang Zhou. 2011. *Computing with Spatial Trajectories*. Springer Science & Business Media.