DeepMM: Deep Learning Based Map Matching with Data Augmentation

Kai Zhao[1], Jie Feng[1], Zhao Xu[1], Tong Xia[1], Lin Chen[2], Funing Sun[3], Diansheng Guo[3], Depeng Jin[1], Yong Li[1]
[1]Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China
[2]Department of Information and Communication Engineering, Tongji University, Shanghai, China
[3]Tencent Inc., Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Map matching is important in many trajectory based applications like route optimization and traffic schedule, etc. As the widely used methods, Hidden Markov Model and its variants are well studied to provide accurate and efficient map matching service. However, HMM based methods fail to utilize the value of enormous trajectory big data, which are useful for the map matching task. Furthermore, with many following-up works, they are still easily influenced by the noisy records, which are very common in the real system. To solve these problems, we revisit the map matching task from the data perspective, and propose to utilize the great power of data to help solve these problems. We build a deep learning based model to utilize all the trajectory data for joint training and knowledge sharing. With the help of embedding techniques and sequence learning model with attention enhancement, our system does the map matching in the latent space, which is tolerant to the noise in the physical space. Extensive experiments demonstrate that our model outperforms the widely used HMM based methods more than 10% (absolute accuracy) and works robustly in the noisy settings in the meantime.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Data mining**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; *Ubiquitous and mobile computing design and evaluation methods.*

KEYWORDS

map matching, deep learning, data driven system

ACM Reference Format:

Kai Zhao[1], Jie Feng[1], Zhao Xu[1], Tong Xia[1], Lin Chen[2], Funing Sun[3], Diansheng Guo[3], Depeng Jin[1], Yong Li[1] . 2019. DeepMM: Deep Learning Based Map Matching with Data Augmentation. In *27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19), November 5–8, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3347146.3359090

SIGSPATIAL '19, November 5–8, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-6909-1/19/11.
https://doi.org/10.1145/3347146.3359090

1 INTRODUCTION

As a fundamental component in map service, map matching is to align the collected discrete user trajectory records to the road segments. It is of great importance in many trajectory based applications of route optimization, traffic scheduling, etc.

The past years have witnessed the great efforts of the research community in improving the accuracy, efficiency, and practicability of the map matching system. Popular methods include geometrical analysis [1], Kalman Filter [7], Hidden Markov Model (HMM) [6], etc. Due to its applicability in sequential modeling and road network connectivity, HMM becomes the widely used benchmark method and lots of variants [2, 3, 5] have been adapted to fit different settings and applications. HMM regards the individual road segments as the states of the HMM and the recorded vehicle location measurement as the state measurement. After observing the locations sequence, HMM uses viterbi algorithm to find the best matching road segments. However, HMM methods have the following two shortcomings,

- **Failing to utilize the potential of historical trajectory.** HMM matches each trajectory individually without using the information of other trajectories. There are at least two kinds of valuable information which are omitted: 1) the historical trajectory records of the same vehicle; 2) the historical trajectory from other vehicles going through the identical or similar road network.
- **Sensitive to the noisy records in the trajectory.** Due to the limitation of the positioning technique and the complicated and changeable environment, the raw trajectories usually contain considerable positioning noise and sparsity issue. As a distance based method, HMM [6] is easily affected by the noisy records in the physical world. Moreover, due to the complexity and randomness of the noisy records, the advanced methods [2, 4, 6] for map matching is still sensitive to the noisy records in the trajectory.

In summary, HMM methods are model based methods, which are failing to utilize the value of enormous trajectory big data and easily influenced by the noisy records in the trajectory.

In this paper, we revisit the map matching task from the data perspective, and propose to utilize the great power of data to help solve the aforementioned problems that obstruct the conventional methods. With the popularity of position devices, enormous trajectory data are continuously generated and collected. These trajectory data records the mobility of different vehicles, which reveal the mobility pattern of vehicles and the noise distribution of positioning techniques. With mining such knowledge from these trajectory data, we are able to complete the map matching task in an intelligent way, which surpasses the simple matching only from the nearby road candidates. We propose to utilize the power of deep learning to design a data-driven model for map matching.

First, to reduce the harmful effects of noisy records in physical space, we introduce the embedding techniques to represent the location and road segment, which are projected into a high dimensional latent space with basic relation reserved. In this way, the following matching process is done in the latent space which is more tolerant to the noise from the physical space. Second, we design an attention enhancement sequence to sequence model to learn the mapping function from the trajectory sequence to the road segment sequence from the enormous trajectory data. All the available trajectory data are used to jointly train a unified model with pattern sharing, which enable to utilize the information from other trajectories to help the individual trajectory map matching. Finally, we propose an effective trajectory data augmentation technique to enrich the trajectory data from a different perspective. The augmentation strategy accelerates the training procedure of the proposed model and obtain better performance in the real data.

2 PROBLEM DEFINITION

A trajectory is a series of spatial points in the order of time. There are two different kinds of trajectories: GPS-based trajectory and segment-based trajectory. The spatial point of a GPS-based trajectory is a GPS location $l = (lon, lat)$ with lon stands for longitude and lat stands for latitude. The trajectory can be represented as $P^l = [l_1, l_2, l_3, ...]$. A segment-based trajectory is a sequence of road segments. In the road network, each road is split into a series of segments and each segment s contains a series of GPS locations, *i.e.* $s = [l_1, l_2, l_3, ...]$. Therefore, segment-based trajectory can be represented as $P^s = [s_1, s_2, s_3, ...]$.

Map matching is matching a raw trajectory to the road network to get a matched trajectory. The raw trajectory is collected by mobile devices of a moving user or vehicle. Due to the limitations of trajectory collecting methods, trajectories may have very low quality in sample interval and noise. Our map matching problem mainly focuses on trajectories with large noise, like the station based trajectories and GPS trajectories with large spatial errors. Matching this kind of trajectories requires the model to be more robust to noise, which is difficult for those distance-based models to satisfy.

3 THE MAP MATCHING SYSTEM: DEEPMM

The framework of our DeepMM system is shown in Fig. 1. It can be divided into 3 parts: 1) Statistic-based trajectory augmentation, 2) Generation-based trajectory augmentation, 3) Attentional seq2seq map matching model. The statistic-based trajectory augmentation and generation-based trajectory augmentation enrich the training dataset by different ways. The third part utilizes the augmented data to train an attentional sequence to sequence map matching model, which maps the raw trajectories to segment-based trajectories.

3.1 Statistic-based Data Augmentation

In statistic-based data augmentation, we directly augment the trajectory dataset from the real trajectories. We have real raw trajectories and ground truth trajectories in pairs, but the size of the dataset is too small to train a deep learning model. We simulate the generating process of the real raw trajectory by down sampling the ground truth trajectories and adding spatial noise.

The sampling interval is the average time interval between adjacent trajectory points while the spatial noise is the spatial distance between the corresponding points of real raw trajectory and ground truth trajectory. We assume the spatial noise follows Gaussian distribution, as follows,

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}, \quad (1)$$

μ is the mean or expectation of the distribution and σ is the standard deviation, and both of them can be learned from the real data.

Suppose the average sampling interval of the real raw trajectories and ground truth trajectories are δ_t and δ'_t respectively. Then we randomly sample the generated ground truth trajectory points with a sampling rate of δ'_t/δ_t. After that, the learned Gaussian noise distribution is added to these subsampled trajectories.

3.2 Generation-based Data Augmentation

Augmenting trajectory dataset just by duplicating existing trajectories is not enough, because existing trajectories only covers a certain area of the road network. Other parts of the road network still lack training data. Thus, we can generate trajectories based on the road network to enrich the training data. The trajectory generation process can be divided into three steps. In the first step, we use a routing algorithm and road network to generate the shortest ground truth routes. In the second step, we use some techniques to perturb the shortest routes, making it more realistic. Finally, we implement the sampling interval and noise distribution from real trajectory data to generate the corresponding raw trajectories.

3.2.1 Shortest Ground Truth Trajectory. First, we use a route planning algorithm to generate trajectories. To plan a route, we only need to give the origin and destination. Empirically, more training data derive a better result. Thus, we uniformly choose the origin and destination from the region of the map so that there are enough data to learn about each part of the map. Given the origin and destination, we use a route planning algorithm [1] based on the Dijkstra algorithm, which generates the shortest route between these two locations.

3.2.2 Perturbed Ground Truth Trajectory. In practice, people do not always follow the shortest route. Thus, we propose a trajectory perturbation method to simulate the real situation. The main idea is adding spatial noise at some waypoints of the shortest trajectory. First, we generate the shortest route R by a routing algorithm. Second, according to the route length d, we randomly choose the number of split waypoints, which is used to split the trajectory. It is calculated as $N_W = d/\alpha * p$, where α is a parameter to control the average number of waypoint and $p \in (0, 1)$ is a random perturbation. From the equation, we can see that the split waypoint number N_W is directly proportional to the length of the route d. Because longer route contains more uncertainty. Third, we randomly choose N_W

[1]https://github.com/graphhopper/graphhopper

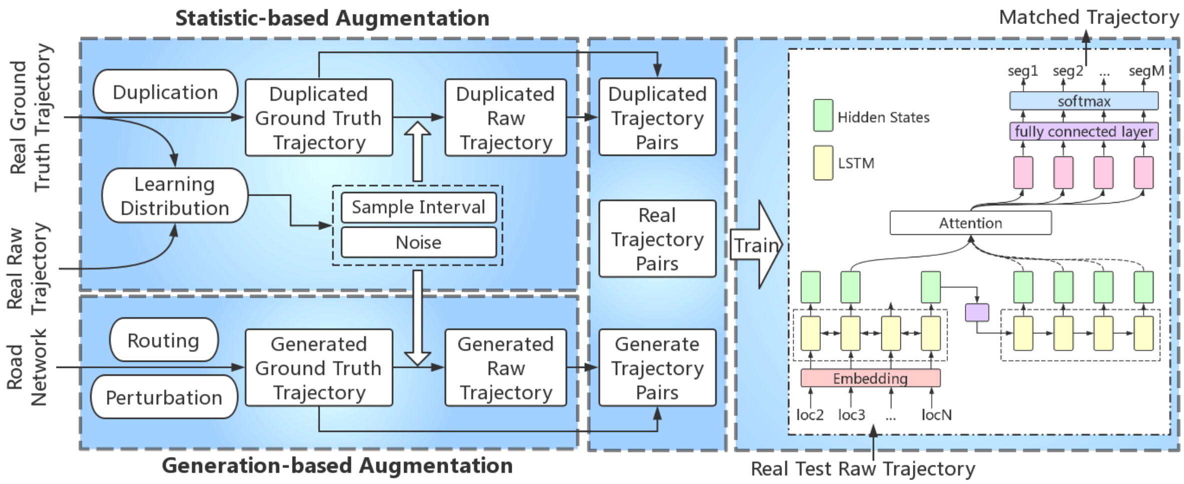


Figure 1: DeepMM map matching system framework

waypoints from the shortest route R and add different noise at each waypoint.

3.2.3 Generate Raw Trajectory. The final step is to generate raw trajectory from the generated ground truth data. Same as the statistic-based augmentation, we subsample the generated ground truth trajectory and add Gaussian noise.

3.3 Attentional seq2seq Model

We model the map matching problem as a sequence to sequence problem. The input sequence is the raw trajectory and output sequence is the segment-based trajectory. We divide the whole map into 100 meters * 100 meters square locations and translate each location ID into a one-hot vector. The one-hot vector is supposed to capture the geographical information of the location and other useful information. The architecture of our map matching model is described Fig. 1. The first core component is location embedding of the input sequence. And then we use a sequence to sequence model [8], which is a general end-to-end approach to map sequences to sequences. It uses multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Further, we apply the attention mechanism to strengthen its capability to capture the long and complicated dependencies.

4 PERFORMANCE EVALUATION

4.1 Evaluation Setup

4.1.1 Dataset and Preprocess. We experiment on a vehicle trajectory dataset which is a vehicle GPS trajectory dataset collected in a large part of urban area of Beijing. It covers a rectangular area from (116.36, 39.89) to (116.46, 39.96) which is 8.4 km long and 7.1 km wide. The dataset contains 12436 high-quality trajectories. Every trajectory has a length of at least 2 kilometers and the average sampling interval is 11 seconds.

We first use the state-of-the-art algorithm, HMM map matching algorithm [6], to map the trajectories to the road network. Newson and Krumm [6] shows that with a sampling interval of 10 seconds, the map matching accuracy can reach as high as 99%, which means the matched result is enough to be the ground truth. Same as the processing method in [6], we simulate raw trajectories from the ground truth trajectory by removing points and adding Gaussian random noise. We simulated sampling intervals of 30, 40, 60, 80, 100, and 120 seconds. The random Gaussian noise has standard deviations of 10, 20, 40, 60, 80, 100, and 120 meters and zero mean. After this, each raw trajectory is paired with a ground truth trajectory.

4.1.2 Baselines and Metric. We compare the performance of our model with two state-of-the-art baselines **HMM** [6] and **CTS** [3].

Our evaluation metric is **accuracy** [9, 10] which is defined as follows,

$$accuracy = \frac{len(P^{sm} \cap P^{sg})}{max\{len(P^{sm}), len(P^{sg})\}} \tag{2}$$

where P^{sm} is the matched trajectory and P^{sg} is the ground truth trajectory. They are all segment-based trajectories. $len()$ calculates the length of trajectory. $P^{sm} \cap P^{sg}$ is the correctly inferred road segments. $max\{len(P^{sm}), len(P^{sg})\}$ penalize a long inferred route as the longer the route, the higher the chance that it contains the correct road segments.

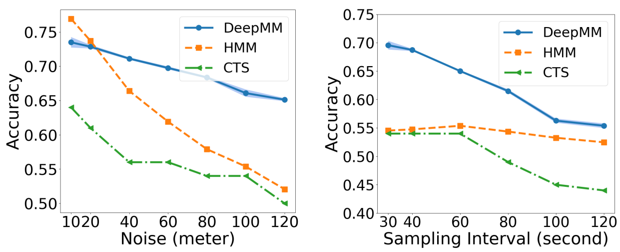
Table 1: The performance of our model and baselines.

	HMM	CTS	DeepMM no-attention	DeepMM
Accuracy	0.55	0.54	0.63	0.66
Improve	0%	-1.82%	+14.54%	+20.00%

4.2 DeepMM Performance

4.2.1 Overall Performance. The result is shown in Table 1, with sampling interval of 60 seconds and the standard deviation of Gaussian noise of 100 meters. Our DeepMM system outperforms two baselines HMM and CTS with the highest accuracy of 0.66. The reason might be that our model can not only learn the distance between the raw locations and the road segments but also the history of all the trajectories. By the way, the accuracy of CTS is a little bit lower than HMM's, which is probably because the sector information is not provided. Besides, the DeepMM model improves 3% in accuracy compared to the no-attention version, which means the attention module works. Each predicted segment is probably correlated with several points of the raw trajectory.

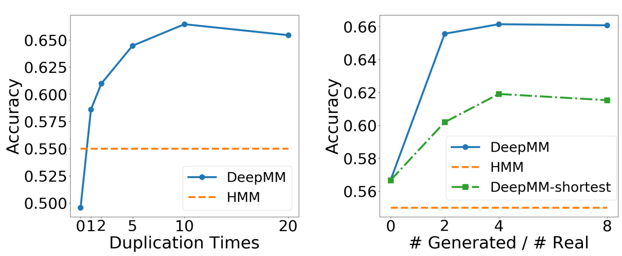
4.2.2 The Influence of Data Quality. We test our model under different sampling interval and noise. In Fig. 2(a), the sampling interval is 60 seconds and the noise varies from 10 meters to 120 meters. DeepMM performs much better than baselines when the noise grows larger, which means DeepMM is more robust to noise. In Fig. 2(b), standard deviation of Gaussian noise is 100 meter and the sampling interval varies from 30 seconds to 120 seconds. The accuracy of DeepMM is always higher than the baselines'.



(a) Noise (b) Sampling Interval

Figure 2: The performance of DeepMM with different data quality.

4.2.3 The Effects of the Data Augmentation. The data augmentation consists of two parts: statistic-based augmentation and generation-based augmentation. We evaluate these two parts respectively. Here we incorporate the best baseline HMM for comparison.



(a) Statistic-based Augmentation (b) Generation-based Augmentation

Figure 3: The performance of data augmentation methods.

The duplication times of statistic-based augmentation range from 1 to 20. 0 means do not apply statistic-based data augmentation. Meanwhile, the size of the generated trajectories is set to a fixed number of 700 thousand. Fig. 3(a) shows that when increasing the duplication times, the accuracy improves enormously from 0.5 to 0.66. The best performance is reached when the original real training dataset is expanded by 10 times. Besides, doubling the original dataset exceeds the best performance of baselines.

To evaluate the performance of generation-based augmentation, we set the statistic-based augmentation multiples to 10 and vary the generated data from 0 times to 8 times of the real data. From Fig. 3(b) we can see that adding only twice the real data, accuracy increases from 0.566 to 0.656, which means the generated data is effective. Increasing the size of the generated dataset, the accuracy continues to rise. The accuracy reaches 0.66 at 4 and remains steady to 8 times of real data.

In conclusion, both augmentation methods bring large performance improvement to our deep learning model.

5 CONCLUSION

In this paper, we investigated the task of map matching by leveraging the power of deep learning. The proposed DeepMM system employs two trajectory augmentation methods to enrich the training dataset and an attentional seq2seq model to map trajectories to road networks. Extensive experiments on a mobility dataset show that our system significantly outperforms existing algorithms. Our work is a preliminary attempt to solve the map matching problem using deep learning methods. We believe deep learning has a great prospect in this area in the future.

6 ACKNOWLEDGEMENT

This work was supported in part by The National Key Research and Development Program of China under grant SQ2018YFB180012, the National Nature Science Foundation of China under 61971267, 61972223, 61861136003, and 61621091, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

REFERENCES

[1] Daniel Chen, Anne Driemel, Leonidas J Guibas, Andy Nguyen, and Carola Wenk. 2011. Approximate map matching with respect to the Fréchet distance. In *ALENEX*. SIAM, 75–83.
[2] Yue-Jiao Gong, En Chen, Xinglin Zhang, Lionel M Ni, and Jun Zhang. 2018. AntMapper: An ant colony-based map matching approach for trajectory-based applications. *IEEE Transactions on Intelligent Transportation Systems* 19, 2 (2018), 390–401.
[3] Xingyu Huang, Yong Li, Yue Wang, Xinlei Chen, Yu Xiao, and Lin Zhang. 2018. CTS: A cellular-based trajectory tracking system with GPS-level accuracy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 140.
[4] George Rosario Jagadeesh and Thambipillai Srikanthan. 2014. Robust real-time route inference from sparse vehicle position data. In *ITSC*. IEEE, 296–301.
[5] Reham Mohamed, Heba Aly, and Moustafa Youssef. 2017. Accurate real-time map matching for challenging environments. *IEEE Transactions on Intelligent Transportation Systems* 18, 4 (2017), 847–857.
[6] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *SIGSPATIAL*. ACM, 336–343.
[7] Dragan Obradovic, Henning Lenz, and Markus Schupfner. 2006. Fusion of map and sensor data in a modern car navigation system. *Journal of VLSI signal processing systems for signal, image and video technology* 45, 1-2 (2006), 111–122.
[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.
[9] Hao Wu, Jiangyun Mao, Weiwei Sun, Baihua Zheng, Hanyuan Zhang, Ziyang Chen, and Wei Wang. 2016. Probabilistic robust route recovery with spatio-temporal dynamics. In *KDD*. ACM, 1915–1924.
[10] Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. 2012. Reducing uncertainty of low-sampling-rate trajectories. In *ICDE*. IEEE, 1144–1155.