



TCL: Tensor-CNN-LSTM for Travel Time Prediction with Sparse Trajectory Data

Yibin Shen¹, Jiaxun Hua², Cheqing Jin¹, and Dingjiang Huang¹(✉)

¹ School of Data Science and Engineering,
East China Normal University, Shanghai, China
ybshen@stu.ecnu.edu.cn, {cqjin,djhuang}@dase.ecnu.edu.cn

² School of Computer Science and Software Engineering,
East China Normal University, Shanghai, China
vichua@stu.ecnu.edu.cn

Abstract. Predicting the travel time of a given path plays an indispensable role in intelligent transportation systems. Although many prior researches have struggled for accurate prediction results, most of them achieve inferior performance due to insufficient extraction of travel speed features from the *sparse trajectory data*, which confirms the challenges involved in this topic. To overcome those issues, we propose a deep learning framework named Tensor-CNN-LSTM (*TCL*) in this paper, which can extract travel speed effectively from historical sparse trajectory data and predict travel time with better accuracy. Empirical results over two real-world large-scale datasets show that our proposed *TCL* can achieve significantly better performance and remarkable robustness.

1 Introduction and Related Work

Thanks to the popularity of GPS-embedded devices, much more trajectory data has been generated, by analyzing which municipal authorities may identify and optimize the traffic congestion in advance. However, predicting an accurate travel time is still very challenging as the travel time is affected by many dynamic factors, such as dynamic departure time, dynamic traffic conditions and dynamic driver behavior. All these ‘dynamics’ make it intractable to predict future pattern of traffic with statistic model [6].

With rapid evolution of deep learning, some studies adopt embedding technologies to solve the challenge of dynamics [3, 7]. They transform departure time, drivers, weather and locations into low-dimensional learnable real vectors, and construct a deep neural network to predict the travel time. Nevertheless, most of them don’t extract travel speed features adequately from *sparse trajectory data* because trajectory data isn’t necessarily generated on all road segments at every moment¹, which results in poor performance.

¹ In our experiments, there are only 1.00% and 1.56% roads in *Beijing* and *Shanghai* can satisfy this condition, respectively.

Meanwhile, tensor/matrix decomposition algorithms have been adopted to solve the data sparsity [5, 8]. However, these decomposition methods often take several minutes to restore the travel time/speed on a road, which is almost infeasible in reality. Even worse, tensor/matrix decomposition algorithms can only estimate the previous travel time/speed of a road because there's no future data in the tensor/matrix. Consequently, it cannot be directly applied to the problem of travel time prediction².

With the aim of solving the aforementioned challenges, we propose a novel deep learning framework named Tensor-CNN-LSTM (*TCL*) for travel time prediction, which can extract travel speed features effectively from historical sparse trajectory data and predict the travel time of a given path with better accuracy.

2 Model Architecture

In this section, we introduce the framework of our proposed *TCL* model, as is shown in Fig. 1. *TCL* is comprised of three major components: non-negative tensor decomposition, long-short-term speed CNN and LSTM prediction model.

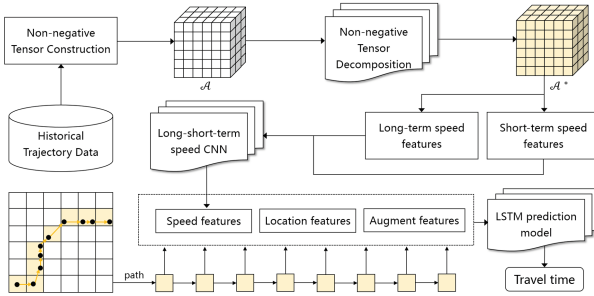


Fig. 1. The framework of *TCL* model.

Non-negative Tensor Decomposition. In the module of non-negative tensor decomposition, we partition an hour into M time slots, and construct three homogenous matrices $A_h, A_m, A_r \in \mathbb{R}^{N^2 \times M}$, where $A_r(i, j) = a$ denotes the i -th grid with travel speed a in time slot j , and A_h is constructed based on historical trajectories over a long period of time (e.g. a week). A_m is a mixed matrix to combine A_r and A_h . After constructing these matrices, we concatenate them together to construct a 3D non-negative tensor $\mathcal{A} \in \mathbb{R}^{N^2 \times M \times 3}$. We fill the missing value in \mathcal{A} by using a fast non-negative CP decomposition algorithm [2].

² Once future information is added, such as the real travel speed, the problem will no longer be travel time prediction [1].

Long-short-term Speed CNN. In the module of long-short-term speed CNN, we extract the long/short-term speed features from a given path, based on \mathcal{A}^* , where the long-term speed features are the travel speed values of the target grid in the past 7 days, and the short-term speed features are the speed distributions in that grid and relevant grids in the previous hour³. Afterwards, we construct a CNN to obtain the whole speed features, as is shown in Fig. 2.

LSTM Prediction Model. The LSTM prediction model consists of two parts: feature extraction layer and prediction layer, as is shown in Fig. 3. The former extracts useful features from the path, such as augment features (the driver ID, the departure time, the day of the week, the travel distance and holiday indicator) and location features (the latitude, the longitude and the grid ID). The prediction layer predicts travel time of the path, which consists of a 2-layer LSTM and a multi-layer perceptron (MLP). The loss function of our model is *Mean Absolute Percentage Error*(MAPE), and the optimizer is Adam.

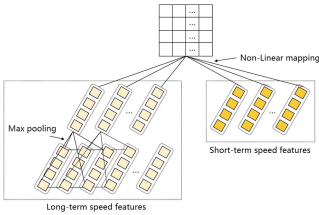


Fig. 2. Architecture of CNN model.

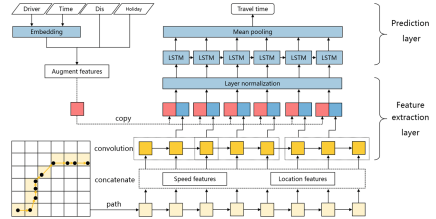


Fig. 3. Architecture of LSTM model.

3 Experiments

Datasets: We evaluate the performance of our model on two real-world trajectory datasets, namely *Beijing* and *Shanghai*. The Beijing dataset contains 3,384,847 trajectories of 10,039 drivers from Oct. 1st to Oct. 31st in 2013. The Shanghai dataset contains 9,727,798 trajectories of 13,622 drivers from Apr. 1st to Apr. 30th in 2015. For each dataset, we split the trajectories generated in the last 7 days as the test set and the rest as the training set⁴.

Results: We select *TEMP* [4], *XGBoost*, *DeepTTE* [3] as baseline methods for comparison with our model. For *DeepTTE* and *TCL*, we train these models for 50 epochs and repeat each experiment 3 times. The mean and the standard deviation are calculated, and the results are shown in Table 1.

³ The top-*k* most relevant grids are calculated by time-shifting KL-divergence.
⁴ The sampling rates on two datasets are different, *Beijing* has low sampling rates (sampling interval is 60 s), and *Shanghai* owns higher sampling rates (10 s).

Table 1. Performance comparison of travel time prediction.

Dataset	Beijing			Shanghai		
Metrics	MAPE (%)	MAE (sec)	RMSE (sec)	MAPE (%)	MAE (sec)	RMSE (sec)
TEMP	23.92	247.20	395.08	23.05	142.25	228.06
XGBoost	18.53	179.41	274.73	21.80	124.84	192.16
DeepTTE	13.47 ± 0.15	137.32 ± 1.03	221.03 ± 1.22	15.11 ± 0.08	99.77 ± 1.55	164.47 ± 3.83
TCL	12.40 ± 0.10	124.58 ± 0.99	200.06 ± 0.38	13.08 ± 0.05	85.96 ± 0.23	142.75 ± 1.07

As we can see, *TEMP* only considers the information of starting points and destinations, resulting in the worst performance. *XGBoost* performs better than *TEMP* on both datasets because the feature selection of *XGBoost* is consistent with our model. However, *XGBoost* can not handle sequence data, so fixing the length of a path will lose some information. *DeepTTE* consider various factors which may affect the travel time, the performance of *DeepTTE* are much better than aforementioned methods. Our proposed *TCL* captures the travel speed accurately, which is the most important factor affecting travel time. *TCL* scores 12.40% and 13.08% (on two datasets respectively) in MAPE, and also outperforms other models in other metrics.

4 Conclusion

In this paper, we propose a novel deep learning framework, namely *TCL*, to predict the travel time of a given path. Specifically, *TCL* can extract travel speed effectively from historical sparse trajectory data and predict travel time with better accuracy. *TCL* achieves satisfying performance on two real-world datasets, which means that we have conquered the challenges of dynamics and sparsity in the trajectory data.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (U1711262, U1811264, 11501204).

References

1. Achar, A., Sarangan, V., Regikumar, R., Sivasubramaniam, A.: Predicting vehicular travel times by modeling heterogeneous influences between arterial roads. In: AAAI, pp. 2063–2070 (2018)
2. Kim, J., Park, H.: Fast nonnegative tensor factorization with an active-set-like method. In: Berry, M., et al. (eds.) High-Performance Scientific Computing - Algorithms and Applications, pp. 311–326. Springer, London (2012). https://doi.org/10.1007/978-1-4471-2437-5_16
3. Wang, D., Zhang, J., Cao, W., Li, J., Zheng, Y.: When will you arrive? Estimating travel time based on deep neural networks. In: AAAI, pp. 2500–2507 (2018)
4. Wang, H., Kuo, Y., Kifer, D., Li, Z.: A simple baseline for travel time estimation using large-scale trip data. In: ACM SIGSPATIAL, pp. 61:1–61:4 (2016)

5. Wang, Y., Zheng, Y., Xue, Y.: Travel time estimation of a path using sparse trajectories. In: SIGKDD, pp. 25–34 (2014)
6. Wang, Z., Fu, K., Ye, J.: Learning to estimate the travel time. In: SIGKDD, pp. 858–866 (2018)
7. Zhang, H., Wu, H., Sun, W., Zheng, B.: Deeptravel: a neural network based travel time estimation model with auxiliary supervision. In: IJCAI, pp. 3655–3661 (2018)
8. Zhou, X., Luo, Q., Zhang, D., Ni, L.M.: Detecting taxi speeding from sparse and low-sampled trajectory data. In: Cai, Y., Ishikawa, Y., Xu, J. (eds.) APWeb-WAIM 2018. LNCS, vol. 10988, pp. 214–222. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96893-3_16