Travel Time Estimation Based on Neural Network with Auxiliary Loss

Yunchong Gan
yunchong@pku.edu.cn
Peking University
Beijing, China

Haoyu Zhang
19990301@pku.edu.cn
Peking University
Beijing, China

Mingjie Wang
xiaowangiii@yahoo.com
Beijing Normal University - Hong Kong Baptist University United International College
Zhuhai, China

Abstract

Estimated Time of Arrival (ETA) plays an important role in various applications, for instance, scene of order dispatch, estimate price, travel time prediction, route decision, etc. In this project, we propose a new systematical Wide-Deep-Double-Recurrent model with Auxiliary loss (WDDRA), which involves Auxiliary Loss for Link Current Status prediction task. Our extensive evaluations show that WDDRA significantly outperforms the state-of-the-art learning algorithms. And our final ensemble model wins second place on the SIGSPATIAL 2021 GISCUP leaderboard without data augmentation. Our source code is available at:https://github.com/Phimos/SIGSPATIAL-2021-GISCUP-2nd-Place-Solution

CCS Concepts: • **Feature engineering** → **Data mining**; *Feature screening*; • **Model structure** → Deep neural network; Multi-Task learning.

Keywords: estimated time of arrival, link current state, deep neural network, ensemble learning

ACM Reference Format:
Yunchong Gan, Haoyu Zhang, and Mingjie Wang. 2021. Travel Time Estimation Based on Neural Network with Auxiliary Loss. In *SIGSPATIAL '21: ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, November 02–05, 2021, SIGSPATIAL, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3474717.3488238

1 INTRODUCTION

In recent years, the widely used car-sharing and online ride-hailing mobile apps are changing our life. It helps people to make efficient use of the vehicles and assists int making travel plans per day. At the same time, the growing population in large cities causing the ever high demands of transport has been one of the major contributive factors of traffic bottleneck problems over the years. In addition, location-based service (LBS) has been an increasingly important problem. Accurate and stable LBS can greatly improve the reliability of app, and so, help to give users a comfortable experience.

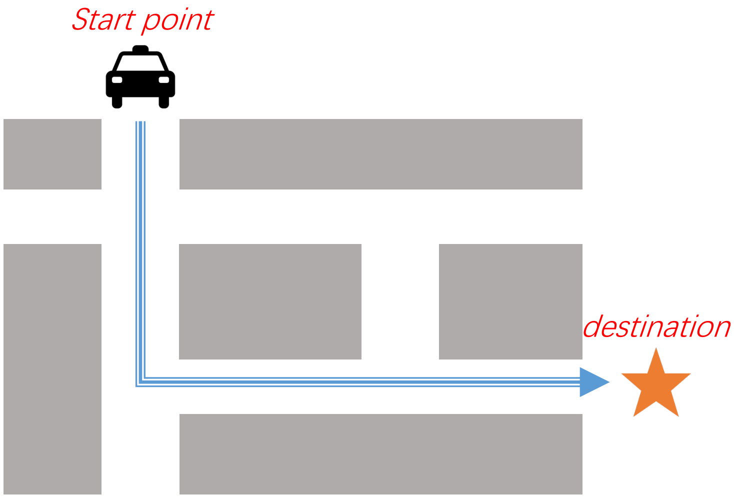

Figure 1. The Estimated Time of Arrival (ETA) Problem

1.1 Background

Accurate Estimated Time of Arrival (ETA) is core functionality in LBS, which is shown in Figure 1. It is the key part to make safer, more stable, and more efficient use of LBS. Thus, it is essential to make an accurate estimate of the travel time before a plan starts.

However, ETA has been a challenging problem for a long time because of high technical difficulty. A lot of factors can undermine the accuracy of ETA[1–3], which includes:

1) Pronounced differences in ETA between weekdays and weekends, as well as between peak and off-peak periods.

SIGSPATIAL '21, November 02–05, 2021, SIGSPATIAL, Beijing, China

ACM ISBN 978-1-4503-8664-7/19/11...$15.00
https://doi.org/10.1145/3474717.3488238

2) Low accuracy in the sum of average time due to cumulative error.
3) Differences in driver behaviours.
4) Unstable average through time in the areas of traffic lights.

1.2 Problem Definition

Given the departure time, route information, real-time traffic and weather features as input, the purpose of the problem is to predict the travel time accurately, which is a typical regression problem. Given a road network, a training dataset which includes trip information from August 1st to August 31st, Shenzhen, China, organized by day and a test dataset including trip information of Sep 1st, Shenzhen, China. The objective task is to minimize the mean absolute percentage error between the estimated and the actual time of arrival.

1.3 Technical Limitation

Most of the regression models such as gradient boosting decision tree (GBDT), which is one of the most popularly used models in ETA problems, is receiving an input vector of a fixed length. Using these kinds of models, the user needs to delete link-level features which are of different feature length among different orders, and use statistical features instead. But original link-level features are very important and indispensable features in ETA, thus there will be information loss if only using statistical information. Besides, GBDT is difficult to adapt to large feature sets[4], which is a fatal flaw in solving our problem.

2 APPROACH

We propose a Wide-Deep-Double-Recurrent model with Auxiliary loss (WDDRA), and its detailed structure is illustrated in Fig. 2. First, to take great advantage of the huge number of historical data with complex data distributions from this contest's dataset, we adapt a deep neural network called WDR[4] into this problem. In addition, by adding an auxiliary loss and adapting ensemble strategy to our model, it will be more stable in the training process and greatly improve its generalization performance, thus, overcome the overfitting issue. In the rest of this chapter, we will introduce the detailed modules of our model.

2.1 Wide-Deep-Double-Recurrent model

The Wide-Deep-Recurrent (WDR) model was aimed to solve the ETA problem given input features of different lengths and data types. Here, we briefly introduce the structure of the WDR network. Interested readers may refer to [4] for more details.

WDR is a deep-learning-based model for learning to estimate the travel time, which combines wide, deep and recurrent models altogether. This type of model inherits advantages of both Factorization Machines (FM)[5] and Deep Neural Network, and it is able to properly use all available information in this contest. WDR contains three main blocks:

(1) The wide model. This block is similar to the wide in Wide & Deep network[6], which contains cross-product and affine-transformation layers. The cross-product layer can project the input features into a high dimensional feature space, and afterwards, the affine-transformation layer $y = \vec{w} * \vec{x} + \vec{b}$ can make feature screening by giving different combined features a different weight according to its feature-importance. In our model, we use a second order cross-product transformation followed by an affine transformation to get a 256 dimensional output.

(2) The deep model. In this block, sparse features, for example, driver ID or time slice are firstly embedded into a dense high-dimensional space. And then, these embedded sparse features can be concatenated with those original dense features to form the input features of the subsequent multiple layer perceptron (MLP) model. In our model, the MLP has three hidden layers with ReLU activation to get a 256 dimension output. Compared to the wide model, this block is a deep model which has the ability to extract more complex feature information.

(3) The double recurrent model. This block is based on a variant of standard RNN which is widely used in several learning tasks for sequential data. In the ETA problem, every travel data contains a series of link and cross information, which is expected to be solved by a recurrent model. In our model, firstly, we use a fully connected layer with ReLU as the activation function to project the input feature of each link into a 256-dimensional space. The feature of each cross is a combination of the features of two links that are connected by this cross. And an additive one-dimension feature of the cross its own information will be concatenated to the cross features. The transformed feature of link and cross is then fed into a LSTM model with corresponding cell sizes, respectively. The last hidden state of two LSTM models are the representation of travel's link and cross sequence, and will be both fed into the final regression model together with the output of Wide & Deep network.

Using the back-propagation (BP) method, all parameters of three modules in the WDR model are jointly trained under MAPE loss. For automatic training, Adam [7], a stochastic gradient descent method with adaptive step size and momentum, is chosen to optimize the model.

2.2 Auxiliary Loss

In multi-Task learning, there will be several tasks trained at the same time by using a combined loss of different tasks all together[8]. Therefore, the network becomes biased towards representations that serve all tasks. In this case, the network is most likely to generalize to new tasks. If there is the main task, the rest of the tasks will be auxiliary tasks, and their loss added to the main loss are called auxiliary loss.

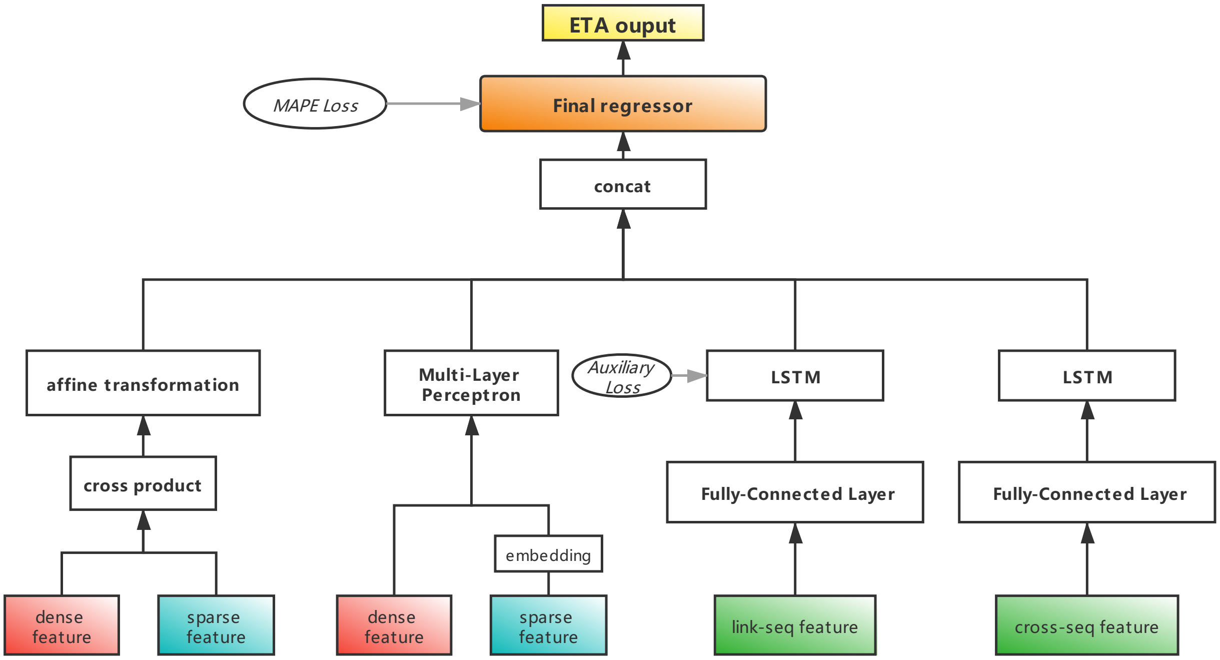

Figure 2. The Proposed Network Structure

We found that the link-current-status feature, which indicates the congestion condition of the road when the car arrived here, is of significant influence on the accuracy of predictors. Therefore, we set the task to predict the link-current-status of an unknown order as an auxiliary task for our model. In the implementation, we add the cross-entropy loss of this auxiliary task to our main MAPE loss of ETA.

2.3 K-fold Bagging method

In order to further improve the accuracy and robustness of our model, we adopt a K-fold Bagging strategy. The bagging method is one of the major kinds of ensemble methods, the basic idea of this method is to train many week learners, which perform not so well by themselves, independently from each other in parallel and combine them following some kind of deterministic averaging process[9]. And bagging method is good at getting an ensemble model with less variance, while other ensemble methods will mainly try to produce strong models less biased.

In this contest, we found that the results of different training parameters are very unstable. In some cases, we found our model is great at predicting travel time with a shorter distance, while in another case, its performance in long-distance orders is better. In other words, WDDRA is a high-variance model. So we choose the bagging strategy to ensemble several WDDRA models together to get a robust and accurate result.

3 EXPERIMENT

3.1 Experimental Settings

Dataset. We use the travel data of Shenzhen in china from didi platform (*https://sigspatial2021.sigspatial.org/sigspatial-cup/*) that includes the departure time, route info, real-time traffic conditions and weather features. And we select data from August 1st to August 24th as the training dataset, data from 25th to August 31st (the last week) as the validation dataset, and data on September 1st as the testing dataset.

Model structure.

- The wide module contains Simple ETA, Distance, Link and Cross number, weekday and slice id as input features to output a 256-dimensional vector which will be fed into the final regression model.
- The deep module will firstly embed Weekday, slice and driver id into a dense high-dimensional space, and then concatenate them with the same numerical features as the wide model to feed into a three layers MLP.
- The recurrent module has the link's time, ratio and status as the link features. And the cross-time is the feature of the cross.

Bagging strategy. For the implementation of our bagging method, we randomly shuffle travel orders of whole training dataset from August 1st to August 31st, and then split those orders into K parts of the same size. Then, we train our model for K times. In each training process, one part of orders is selected as the validation set, and the rest of orders is the training set with which the model is trained. After that, K models trained on different dataset are combined to get a K-fold ensemble model.

Hyper-Parameters. The default hyper-parameters in the training process are as followed: we use SGD with a batch size of 512. And use Adam as an optimizer, the learning rate is set to be 10^{-4} initially, and without weight decay. The models are trained up to 25 iterations without the early-stop strategy.

Evaluation Metrics. Metrics of this problem is the Mean Absolute Percentage Error (MAPE) between the estimate and real travel time, which can evaluate the accuracy of our prediction model. The mathematical definition of MAPE is as followed:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|eta - ata|}{ata} \tag{1}$$

n is the number of travel orders, eta and ata are estimated time of arrival and actual time of arrival, respectively.

3.2 Performance Evaluation

The performance comparison of different models is shown in Table 1. It is obvious that the WDDR model has better performances compared to the classical WDR model, which proves the effectiveness of our approach. Besides, the additional auxiliary loss can improve the accuracy of models by 0.048% approximately, either for WDR or WDDR model. This is because the side task of predicting the link-current-state contributes greatly to the main ETA task.

Table 1. Performance Comparison

Model	MAPE
simple ETA	12.578%
WDR	12.103%
WDR + auxiliary loss	12.045%
WDDR	12.037%
WDDR + auxiliary loss	**12.000%**

In order to further improve the performance of our model, we adapt the ensemble learning method to it, and the result of the K-fold Bagging method based on the WDDR with auxiliary loss is shown in Table 2. As we can see, the model fusion of the 5-fold and 10-fold bagging strategy outperforms the single-fold original model by almost 3% in terms of MAPE metrics.

In addition, we observe that there is a consecutive improvement to model performances as the number of fold increases. This means that our K-fold Bagging method can steadily improve model performances until a critical point, beyond which no more improvement will be achieved. Unfortunately, limited by the time of this contest, we didn't find this optimal fold number yet.

Table 2. Improvement of K-fold Bagging

Bagging strategy	MAPE
single fold	12.378%
5 fold	12.161%
10 fold	12.138%
model fusion	**12.086%**

4 CONCLUSION

In this paper, we propose a Wide-Deep-Double-Recurrent model with Auxiliary loss named WDDRA. In this model, we adopt a deep neural network WDR to extract multi-scale information from the complex given data, and an auxiliary loss is added to solve the overfitting issue. The empirical results of WDDRA in the SIGSPATIAL 2021 GISCUP competition dataset, which wins second place, show that our model has excellent performances in the ETA problem.

Acknowledgments

References

[1] Qiang Wang, Chen Xu, Wenqi Zhang, and Jingjing Li. GraphTTE: Travel time estimation based on attention-spatiotemporal graphs. *IEEE Signal Processing Letters*, 28:239–243, 2021.

[2] Kun Fu, Fanlin Meng, Jieping Ye, and Zheng Wang. CompactETA: A Fast Inference System for Travel Time Prediction. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1:3337–3345, 2020.

[3] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. HetETA. pages 2444–2454, 2020.

[4] Zheng Wang, Kun Fu, and Jieping Ye. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 858–866, 2018.

[5] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000, 2010.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[9] Marina Skurichina and Robert PW Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002.