Learning to Estimate the Travel Time

Zheng Wang, Kun Fu, Jieping Ye
DiDi AI Labs, Didi Chuxing
{wangzhengzwang,fukunkunfu,yejieping}@didichuxing.com

ABSTRACT

Vehicle travel time estimation or estimated time of arrival (ETA) is one of the most important location-based services (LBS). It is becoming increasingly important and has been widely used as a basic service in navigation systems and intelligent transportation systems. This paper presents a novel machine learning solution to predict the vehicle travel time based on floating-car data. First, we formulate ETA as a pure spatial-temporal regression problem based on a large set of effective features. Second, we adapt different existing machine learning models to solve the regression problem. Furthermore, we propose a Wide-Deep-Recurrent (WDR) learning model to accurately predict the travel time along a given route at a given departure time. We then jointly train wide linear models, deep neural networks and recurrent neural networks together to take full advantages of all three models. We evaluate our solution offline with millions of historical vehicle travel data. We also deploy the proposed solution on Didi Chuxing's platform, which services billions of ETA requests and benefits millions of customers per day. Our extensive evaluations show that our proposed deep learning algorithm significantly outperforms the state-of-the-art learning algorithms, as well as the solutions provided by leading industry LBS providers.

KEYWORDS

Location-based services, estimated time of arrival, wide-deep-recurrent learning

ACM Reference Format:
Zheng Wang, Kun Fu, Jieping Ye. 2018. Learning to Estimate the Travel Time. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3219900

1 INTRODUCTION

In recent years, the bloom of sharing economy is changing our life in various respects. One representative is the widely used car sharing and online ride-hailing mobile app which redefines the way people move. Several unicorn companies are growing rapidly, such as Uber, Lyft and Didi Chuxing, which help people to make efficient use of the vehicles and benefit millions of people per day. The rapid expansion of this industry field makes the location-based service (LBS) an increasingly important problem, as an efficient and precise LBS is one of the foundations for the ride-hailing platforms to provide high quality transportation services and enjoyable travel experience for their customers.

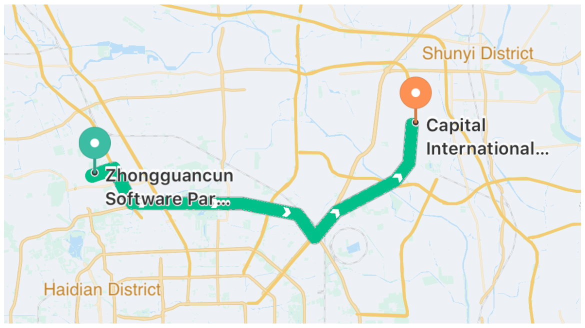

Figure 1: Estimated time of arrival (ETA) refers to the estimated travel time between a pair of origin (green pin) and destination (red pin) along the given route (green lines). It is one of the most important location-based services for digital maps and navigation systems. It is also one of the most important back end services for the ride-hailing apps.

In this paper, we consider the estimated time of arrival (ETA) as the travel time estimation between a pair of origin and destination locations, which is demonstrated in Figure 1. It is an important location-based service for digital maps and navigation systems. ETA finds wide applications on the ride-hailing platform, as travel time is one of the key concerns for drivers and riders to make a deal. Thus it is essential to make an accurate estimate of the travel time before a trip starts. An accurate ETA will boost the efficiency of the transportation system, reducing travel cost for the users, saving energy consumption and reducing motor vehicle pollution. Therefore, ETA has become a core component that influences decision-making at different stages of the online ride-hailing process, including route selection, vehicle dispatch, carpooling et al.

Travel time estimation has been widely studied in geographic information systems (GIS) [1, 2, 6, 8, 13, 15, 19, 20, 22]. And standard solutions have been established in the community. The existing solutions can be divided into two categories. The first category is the route-based solution, which represents the travel time using an intuitive physical model: the overall travel time of a given route is formulated as the summation of the travel time through each road segment[1] and the delay time at each intersection. The travel time estimate $\hat{y}$ is represented as

$$\hat{y} = \sum_i \hat{t}_i + \sum_j \hat{c}_j, \tag{1}$$

[1]A road segment or link is defined as a route in the road network which has no junctions at the middle but has junctions at its both ends. It can be viewed as a building block of the road network.

where $\hat{t}_i$ is the travel time estimated for the i-th road segment and $\hat{c}_j$ is the delay time estimate at the j-th intersection. This solution has been widely adopted in both research field and industrial applications. It divides the travel time estimation of the whole route into several subproblems, including estimation of the travel time for each road segment and the delay time at each intersection. Much attention has been paid to investigate these subproblems in the literature. Various sources of GIS related data have been introduced to estimate the travel time in road segments or subroutes [11, 15, 18]. Machine learning algorithms, such as the regression and tensor decomposition algorithms, have also been adopted to predict the travel time or traffic speed over the road network [20]. Other explorations have focused on approximating the travel time for a given route using a more general subroute, which may be an arbitrary part of the given route [15, 21]. However, the problem is still difficult to solve properly. This classical type of solutions has several inherent drawbacks: 1) Though the amount of available data and the diversity of the data sources increase dramatically, the spatial-temporal coverage of the available data is still very sparse. It is far from enough to monitor the real-time traffic patterns for the entire road network; 2) The transportation system is a dynamic system. It is difficult to physically model the future moving pattern in an explicit form. For example, it is hard to predict what the traffic condition is in a certain road segment and what the traffic light is at a specific intersection, when the vehicle arrives in a future time. Thus high estimation accuracy may not be guaranteed for each $\hat{t}_i$ and $\hat{c}_j$; 3) The model in (1) divides the overall travel time into several parts. This may lead to accumulation of estimation errors and it is harmful for the overall estimation precision; 4) Personalization is a very important factor for ETA. The travel time for a given trip may vary significantly for different drivers and riders. However, it has been neglected in the existing works.

The second category is the data-driven solution. Recently the explosion of the data warehouse makes machine learning a powerful tool to handle the prediction problem. Besides the use of machine learning in the route-based solution to predict the traffic speed and travel time on each road segment, several explorations have been conducted to directly predict the travel time of the whole route at a future time period based on its historical travel times. Several methods model the travel time for a given route at different time periods as a time series [6, 22]. Then the problem of ETA is formulated as a multivariate time series prediction problem. The method in [19] proposes to estimate the travel time of a queried route using a weighted average of its neighboring trips, which refer to the trips with similar origin and destination locations. This approach has better scalability. However, the aforementioned second category of solutions has the following drawbacks: 1) The insufficient data coverage problem still exists. It is difficult to obtain the travel time at all historical time periods for the queried route or even a similar route. Thus this type of methods is mostly investigated on highways, which has few road segments, more stable traffic condition and better data coverage; 2) The travel time prediction is restricted to several fixed routes. It is difficult to be generalized into the unseen routes, which limits the scalability of the problem; 3) Many key information, such as traffic information and personalized information, has been neglected in these methods, which prevents them from obtaining high prediction accuracy. With the extensive use of large-scale historical data and machine learning tools in the travel time estimation problem, the boundary of the route-based method and the data-driven method is becoming vague. However, the inherent drawbacks of insufficient data coverage, weak generalization ability and insufficient information usage limit the effectiveness of the existing methods.

In this paper, we propose a systematic machine learning solution for travel time estimation that overcomes the drawbacks of the existing methods. We establish a rich and effective feature system for the location-based data, including the floating-car data, the road network data and the user behavior information. Based on the extracted features, we formulate ETA as a regression problem. The problem can be solved by popular machine learning methods, such as the gradient boosting decision tree (GBDT) [9] and factorization machines (FM) [17]. Moreover, we build a novel deep learning model to solve this problem. In this model, we jointly train wide linear models, deep neural networks and recurrent neural networks together to combine their benefits. It balances the memorization, generalization and representation abilities in one model and effectively alleviates the limitations in existing methods. The results on large-scale travel data and real-time online system demonstrate that the proposed solution outperforms state-of-the-art solutions for travel time estimation. The proposed solution has been deployed on Didi Chuxing's platform, servicing billions of ETA requests every day.

The main contributions of this paper include:

- We formulate the problem of ETA as a pure regression problem.
- We propose a novel deep learning model to solve the ETA learning problem. The proposed method outperforms existing methods.
- We construct a well-designed feature system for location-based data.
- We evaluate our algorithm using large-scale historical data and real-time online queries from Didi Chuxing. All results show that the proposed solution significantly outperforms state-of-the-art solutions.

The rest of this paper is as follows: Section 2 reformulates the ETA problem into an end-to-end machine learning problem and introduces the proposed solutions and Section 3 presents the related works, Section 4 provides large-scale offline experiments and rigorous online A/B tests to verify the superior performance of the proposed solution and we conclude this paper in Section 5.

2 APPROACH

In this paper, we define the travel time estimation problem as follows:

Definition 2.1 (ETA learning). Suppose we have a database of trips, $\mathbf{D} = \{p_i, s_i, e_i\}_{i=1}^N$, where p_i is the trajectory path for the i-th trip, s_i is the departure time and e_i is the arrival time. The actual travel time is given by $e_i - s_i$. Given a query $q = (o_q, d_q, s_q)$, our goal is to estimate the travel time t_q with a given origin o_q, destination d_q, and departure time s_q.

2.1 Feature Extraction

The wide usage of the real-time navigation apps and the online ride-hailing apps produce a large amount of floating-car data per day. Such huge amount of data makes accurate travel time prediction possible by using machine learning methods. However, in conventional data-driven methods, one of the barriers for achieving good performance is the limited number of features used in model training. To ensure the effectiveness of machine learning algorithms, we first systematically build rich features for the location-based data and establish a high dimensional feature mapping for them. We summarize the features into several types: spatial information, temporal information, traffic information, personalized information and augmented information.

Spatial Information: The travel time is highly correlated with the route that the vehicle travels through and the geographical area where the trip happens. Thus we extract the set of features according to geospatial information. We first map the vehicle trajectory into the base road network and obtain the sequence of road segments and intersections. Then we extract the characteristics of all the building blocks that compose the route, such as the road segment, intersection and traffic light information. For example, we will extract all characteristics that we can obtain for the road segment, including the length, width and grade of the road segment, the number of lanes in the segment, the index number of the segment in the road network et al. We also extract the POIs information within the areas where the route goes through.

Temporal Information: The temporal information is another key factor that affects the vehicle travel time; for example the travel time for a specific route usually will be much longer in the rush hours than the off-peak hours. Thus we indicate the departure time of the trip with different features, including the time period in a year, a month and a day, the holiday indicator and rush hour indicator, et al.

Traffic Information: The traffic condition in the traffic network has a direct impact on the travel time. We build a traffic monitor and prediction system that provides us the real-time traffic speed estimation in each road segment in the traffic network in every two minutes. Then we use several types of travel speeds for each road segment in our features, such as the real-time estimated speed, the average speed and free-flow speed, et al.

Personalized Information: The travel time is person specific, as different persons may have very different driving preference. Thus we introduce personalized information in the features, including driver profile, rider profile and vehicle profile, et al.

Augmented Information: All other available information is used as augmented features, including the weather information and traffic restriction, et al.

The extracted features include both continuous and categorical features in the forms of real value, discrete value and high dimensional one-hot features. After sophisticated feature engineering, we obtain a set of features in hundreds of categories and millions of dimensions.

2.2 Learning to Estimate the Travel Time

Based on the well-designed features, we rewrite the ETA learning problem in standard machine learning form. Let us denote $\mathbf{y} = [y_1, y_2 \cdots, y_N]$ as the ground-truth label for each sample, where $y_i = e_i - s_i \in \mathbb{R}_+$ is known as the travel time for each trip that is computed as the time gap between the arrival time e_i and the departure time s_i. Let us denote the samples by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, where each sample $\mathbf{x}_i \in \mathbb{R}^d$ decodes the trajectory along the route path p_i in the road network as a d dimensional vector. Our target is to train a model, which can accurately predict the travel time for a future unseen data $\mathbf{x}_q \notin \mathbf{X}$.

In travel time estimation, the user tolerance of the estimation gap varies according the total travel time. Thus the mean absolute percentage error (MAPE) is a more reasonable metric for us in this problem. And our target is to directly minimize MAPE as

$$\min_f \sum_{i=1}^{N} \frac{|y_i - f(\mathbf{x}_i)|}{y_i}, \tag{2}$$

where $f(\mathbf{x}_i)$ is the ETA of the route $\mathbf{x}_i$, and the function f is a regression model. To guarantee good prediction performance on the unseen data, we need to constrain the complexity of the model by introducing extra regularization terms to control over-fitting. Thus the general optimization objective for the ETA learning problem becomes:

$$\min_f \sum_{i=1}^{N} \frac{|y_i - f(\mathbf{x}_i)|}{y_i} + \Omega(f), \tag{3}$$

where $\Omega(f)$ is the regularization term that controls the complexity of the model f. It can be learned by using proper machine learning models. For example, we could use gradient boosting decision tree (GBDT) [4], which is one of the most popularly used learning algorithms in real-world problems. Another efficient and practical model is the factorization machines (FM) [17].

In GBDT, the prediction model can be formulated as an additive tree model which is

$$f(\mathbf{x}) = \sum_{t=1}^{T} f_t(\mathbf{x}). \tag{4}$$

T is the number of trees. $f_t(\mathbf{x}) = w_{tu(\mathbf{x})}$ is a base tree model, where $\mathbf{w}_t$ is the vector of scores on leaves, and u is a function assigning each data point to the corresponding leaf. Thus given an input sample $\mathbf{x}$, the tree model returns the score of the leaf that $\mathbf{x}$ belongs to. The complexity of the GBDT model f is control by the structure of the trees. The complexity of a tree is defined as

$$\Omega(f_t) = \gamma L + \frac{1}{2}\sum_{i=1}^{L} w_{ti}^2, \tag{5}$$

where γ is a tunable parameter and L is the number of leaves [4]. By introducing specific GBDT model and regularization terms in the objective, the optimization problem becomes:

$$\min_f \sum_{i=1}^{N} \frac{|y_i - f(x_i)|}{y_i} + \sum_t (\gamma L_t + \frac{1}{2}\|\mathbf{w}_t\|_2^2). \tag{6}$$

This is a convex optimization problem. However, the objective is non-smooth as the MAPE function is non-differentiable. We could use the Huber loss to approximate the MAPE function or adopt the subgradient method to solve the optimization problem (3).

FM combines the high-prediction accuracy of factorization models with the flexibility of feature engineering. It has been widely applied in recommender systems and online advertising systems [5, 17], which are similar to our problem, as both problems aim to find the most proper user preference for future prediction. FM models all nested interactions up to order k between the d input variables in $\mathbf{x}$ using factorized interaction parameters. The model of order $k = 2$ is defined as,

$$f(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j. \tag{7}$$

$\mathbf{v}_j \in \mathbb{R}^m$ is an embedding transformation for the interactions with $m < d$. By substituting the prediction function and necessary model complexity control terms, the optimization objective becomes:

$$\min_{\mathbf{w},\mathbf{V}} \sum_{i=1}^{N} \frac{|y_i - f(x_i)|}{y_i} + \lambda_1 \|\mathbf{w}\|_2 + \lambda_2 |\mathbf{w}| + \lambda_3 \|\mathbf{V}\|_F. \tag{8}$$

The problem is solved by gradient methods. We follow one of the online optimization algorithms for MF by updating $\mathbf{V}$ using adaptive subgradient method (AdaGrad) [7] and updating $\mathbf{w}$ using follow-the-regularized-leader (FTRL) [16].

2.3 Wide-Deep-Recurrent Learning

In the real application, we find that GBDT and FM are not the best choices to solve the ETA learning problem. GBDT is difficult to adapt to large feature sets. And the performance of FM highly depends on the representation of the features and the model representation ability is limited. On the other hand, we have a huge number of historical data with complex data distributions. This enables us to use more complex models to solve the problem. Thus we adapt the deep learning technique into this problem.

Wide & Deep Learning [5] is proposed for recommender systems, which jointly trains wide linear models and deep neural networks to combine the benefits of memorization and generalization abilities of the learning system. The model has a two-block structure, summarized in Figure 2.

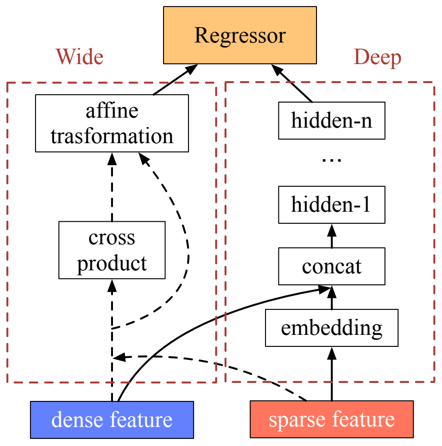

Figure 2: Model Structure of Wide & Deep Learning: the wide linear model is on the left and the deep neural networks are on the right.

The wide model first projects the input features into a high dimensional feature space. This could be achieved by computing the cross-product of the input features, which is similar to the feature interactions in FM. An affine transformation $y = \mathbf{w} \cdot \mathbf{x} + \mathbf{b}$ may be applied after the cross-product transformation. The wide model can be explained as a generalized linear model, which is similar to the traditional recommendation algorithms like logistic regression. Compared to the deep model, the wide model may be considered as a shallow model.

The deep model first converts the sparse input features into dense features using the feature embedding layers. This embedding is to represent each category of high dimensional sparse feature by a compact feature vector (size=20 in out setting). The dense input features are then concatenated with the embedded features and feed into a feed-forward neural network (FNN), also known as multiple layer perceptron (MLP). A regressor at the top layer combines the output from both wide and deep models to provide the final prediction. Usually, the sparse features are used to represent the categorical information, and the dense features are used to represent the numerical information. For example, the driver-id is a sparse feature, while the route length is a dense feature.

All aforementioned models require that the features in each sample should be aligned. We can only extract the statistical information of the trip under this requirement, as the number of road segments in different trips are usually different. The wide and deep models are able to capture the global statistical information among the route and geospatial areas where the vehicle travel through. However, they are not good at capturing the local traffic information in each road segment. Thus we need to introduce extra network structure to capture the local information of the road segment. In the map data, the road network has been divided into different road segments and these segments can be viewed as building blocks of routes. And the road segments along each route have clear sequential structure. This is similar to the situation in natural language processing, where words are the building blocks of sentences. This inspires us to introduce recurrent neural network.

Long-Short Term Memory (LSTM) [10] is a specific recurrent neural network (RNN). It has achieved great success in several learning tasks for sequential data, such as neural machine translation [3]. LSTM can capture both the local information of each segment and the long-term dependency along the sequence. Thus we introduce it to model the road segment sequence in our problem. A standard LSTM structure is shown in Figure 4. LSTM alleviates the gradient vanishing and exploding problem of RNN by using an additive memory cell to provide a highway for the gradient to pass through, and using several gates to further control the information flow. In each inference step of LSTM, the input gate, forget gate, output gate and modulated input are updated as:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i[\mathbf{x}_t; \mathbf{h}_{t-1}] + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{x}_t; \mathbf{h}_{t-1}] + \mathbf{b}_f),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{x}_t; \mathbf{h}_{t-1}] + \mathbf{b}_o),\\
\mathbf{g}_t &= \tanh(\mathbf{W}_g[\mathbf{x}_t; \mathbf{h}_{t-1}] + \mathbf{b}_g),
\end{aligned} \tag{9}$$

where $\sigma(\cdot)$ is the Sigmoid function $\sigma(u) = 1/(1+e^{-u})$. Each equation consists of an affine transformation and a non-linear activation.

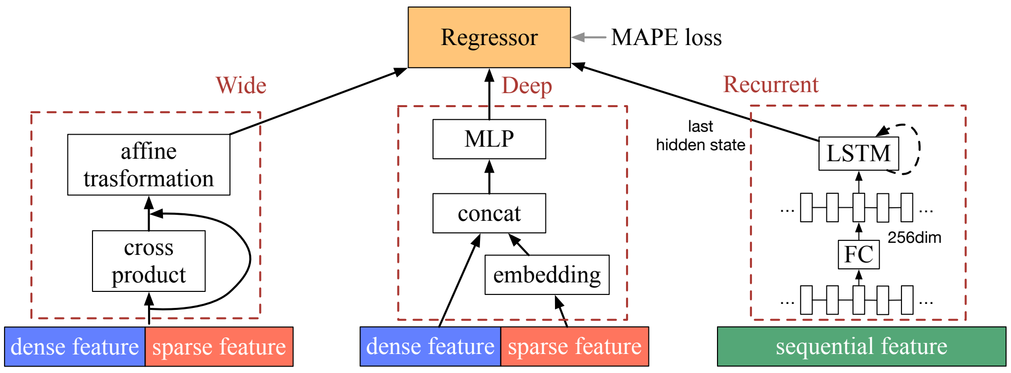

Figure 3: The Proposed Wide-Deep-Recurrent Network Structure.

Then the memory cell and hidden state are updated as:

$$\begin{aligned} \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \end{aligned} \tag{10}$$

where $\odot$ represents element-wise multiplication.

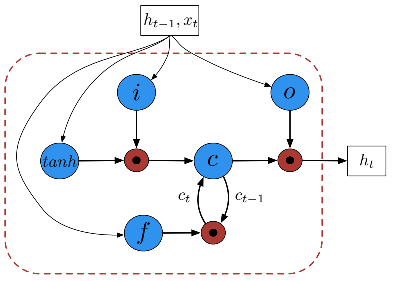

Figure 4: Structure of Long-Short Term Memory.

Wide-Deep-Recurrent Learning: We combine wide, deep and recurrent models all together to build our deep learning model for learning to estimate the travel time. The model structure is described in Figure 3. This model inherits the advantage of its ancestors, and makes effective use of the dense feature, the high dimensional sparse features and the local features along the road segment sequence. It is able to properly use all available information in the ETA learning problem.

WDR model has three main blocks: 1) the *wide model* is similar to the wide model in Wide & Deep network. We use a second order cross-product transformation followed by an affine transformation to get a 256 dimensional output; 2) the *deep model* embeds the sparse features into a 20 dimensional space, then processes the concatenated features by a 3-hidden-layer MLP with a ReLU [14] activation to get a 256 dimensional output. The size of all the three hidden layers in the MLP is 256; 3) the *recurrent model* is a variant of standard RNN. The feature of each road segment is first projected into a 256 dimensional space by a fully connected layer with ReLU as the activation function. The transformed feature is then fed into to a standard LSTM with cell size 256. The last hidden state $\mathbf{h}_T$ of the LSTM is input to the top regressor, where T is the length of the road sequence.

All parameters in the WDR model are jointly trained using back-propagation (BP) under the MAPE loss. Due to the combination of three modules, it is difficult to find a proper global learning rate. Therefore, we choose Adam [12], a stochastic gradient descent method with an adaptive step size and momentum, to optimize the model. We set the learning rate of Adam to 0.001, and initialize other hyper-parameters following the guideline in [12].

2.4 Overall Pipeline

We summarize our solution pipeline for learning to estimate the travel time in Figure 5. In the data aggregation module, we first match the GPS trajectory to the road network to obtain the corresponding sequence of road segments. Then a feature extractor aggregates road network information, trajectory information, order context information and augmented information to produce the input data for the training module. After the feature extraction, we start offline training based on large-scale historical data. Then we further fine-tune the model with a set of latest generated data to make sure that the model adapts to up-to-date data distribution. The model is pushed to the online server, once the fine-tune finishes.

3 RELATED WORK

In this paper, we formulate the travel time estimation problem as a spatial-temporal regression problem and propose WDR learning to solve it. Artificial neural networks, including MLP and RNN, have been adopted to solve the travel time estimation problem in the literature [6, 22]. However, these conventional methods apply standard deep learning models to solve a time series prediction problem. In those works, the main input feature for the neural network is the historical travel time for the queried routes. With such a small feature set, it is difficult to take full advantage of the deep learning models. Besides, those solutions do not work well in the multiple step forecasting situation, which is a common issue in time series prediction. As far as we know, the proposed WDR learning model is the first specially designed deep learning model to

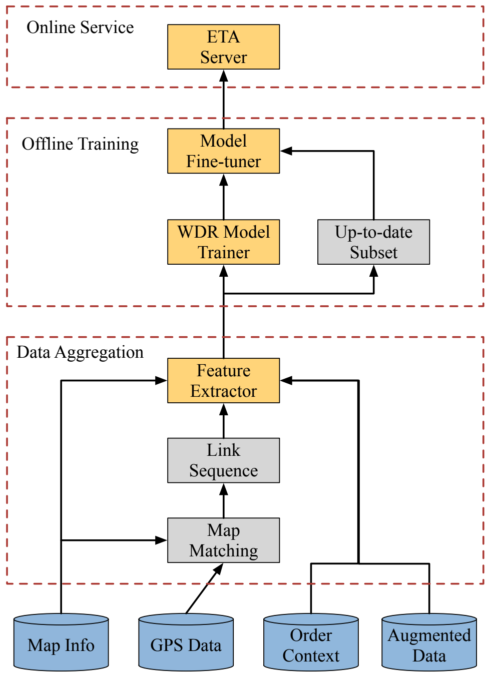

Figure 5: ETA Service Pipeline Overview.

solve the travel time estimation problem. We solve the road trajectory mining problem by analogy with natural language processing (NLP). Each route is regarded as a sentence, and each road segment with the corresponding interaction as a word. The WDR network captures the overall statistical property of the trip by wide and deep models, and captures the detailed characteristics of the sequential road segments by recurrent models. The richness and effectiveness of the well-designed features guarantee the generalization ability of the learnt deep learning model, which can be used to accurately predict the travel time for any unseen trip in the road network starting at any time.

In this paper, we also adopt GBDT and FM to solve the formulated spatial-temporal regression problem. Both of them show competitive performance. We try to explain this result by briefly comparing and analyzing the three models we have used in this paper: GBDT, FM and WDR network. GBDT introduces a nonlinear transformation into the model representation based on decision trees. It can be viewed as a simple deep model. When the available training data and the feature set are both limited, GBDT is a reasonable choice. FM can be viewed as a model combination with two parts: one is the linear model part, which is a wide linear model; the other part represents the nonlinear feature interaction, and it is equivalent to a two layer MLP, which can also be viewed as a simple deep model. Thus FM is a simplified wide & deep model. And it can be used when the training data and the feature set are in large size. In WDR learning, the deep and wide model parts have much higher complexity than FM, which has better representation ability if the model could be trained with sufficient training data. The recurrent model part introduces more capability to represent the detail information of each road segment. By making use of more information, the WDR learning obtains better prediction performance. And we believe the proposed solution is more promising with the increase of both available data and features. Besides all above benefits, the WDR model can be applied in more general sequence learning problems.

4 EXPERIMENT

We empirically evaluate the proposed solution on large-scale offline datasets. We also build a real-time ETA service based on this solution and compare it with major ETA service providers.

4.1 Datasets

We collect the floating-car data in Beijing from Jan. 1st to May 31st, 2017 on DiDi platform. The data are then divided into two different types according to the working status of the driver – pickup data and trip data. A pickup sample is collected when the driver responds to a rider's request until he/she picks up the rider. A trip sample is collected when a rider is on board until the rider arrives at the destination. Vehicle trajectories collected from real-world scenarios can be extremely complicated. For example, one route may contain urban freeways, local streets and private roads in residential community. Figure 1 presents a typical example on DiDi platform, which starts from a local street, goes through several freeways and finally drives into the internal roads of an airport. To accurately predict the travel time for such route is a challenging problem.

Table 1: Statistics of offline dataset

	date	pickup	trip
training set	1.1 - 5.10 (2017)	48M	51M
validation set	5.11 - 5.17 (2017)	3M	4M
test set	5.18 - 5.31 (2017)	6M	7M
unique links	–	0.5M	0.5M
unique drivers	–	0.4M	0.4M

After removing the abnormal cases with very short travel time (< 60s) or extremely high travel speed (>120km/h), we obtained about 57 million samples in the pickup dataset and 62 million samples in the trip dataset. Table 1 lists the statistics of these two datasets. Note that the trajectories are distributed among the entire road network of Beijing, they cover 0.5 million unique road segments in both pickup and trip datasets. And the data are collected from 0.4 million drivers for each dataset.

As most ETA models suffer from the problem of data sparsity, we analyze the statistics of the data coverage on the given datasets. We plot the average link coverage through a whole day in Figure 6. The results show that the trip trajectories have better coverage than the pickup trajectories in general. The statistics on the trip data show that DiDi drivers' trajectories cover about 45% of the road links at daytime. The coverage drops to its minimum before dawn.

4.2 Competing Methods

On the offline datasets, we compare our solution with several competitors, including a representative route-based solution **route-ETA** and two state-of-the-art methods in the literature – **TEMP**$_{\text{rel}}$

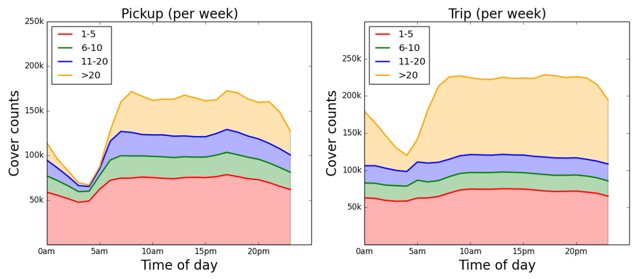

Figure 6: Visualization of road coverage. The number associated with each curve represents the times that each road has been traveled through repeatedly by DiDi drivers.

[19] and **PTTE** [20]. We also evaluate our ETA machine learning solution by using different models, such as GBDT, FM and a combination of WD network with MLP. During the online evaluation, we compare our real-time service based on WDR model with three leading industry LBS providers by rigorous A/B test.

Route-ETA is a widely used solution in real-time map services and navigation systems. In this solution, ETA of a given route is a weighted sum of the travel time in each road segment and the delay time at each interaction. Route-ETA estimates the travel time in each road segment by dividing its length by the real-time traffic speed on the segment. The traffic speed in each road segment and delay time at each intersection are provided by a real-time traffic monitoring service.

TEMP$_{\text{rel}}$ [19] is a route-free method that estimates the travel time of a queried trip based on its neighboring trips. Two trips are considered as neighbors if the distance between their origins and between their destinations is less than a certain threshold. The ETA of a queried trip is obtained by averaging the travel time of its neighbors. The result of TEMP$_{\text{rel}}$ is restricted to a subset of the test data, as TEMP$_{\text{rel}}$ requires that the test sample has at least one neighbor in the training set. The subsets contain about 68% samples for pickup dataset and 61% samples for trip dataset[2].

PTTE [20] models the ETA building block as the element in a 3D tensor, which is the travel time for each road segment during each time slot by each driver. It estimates the missing element in this tensor by using low rank tensor completion algorithm. Then dynamic programming is used to find the minimum travel time between two given locations as the result of ETA.

To verify the importance of sequential dependency among the road segments in estimating the travel time, we compare different machine learning models under our framework, including GBDT, FM and a variant of WDR network named WD-MLP. In WD-MLP, an MLP is used to replace the recurrent module. The MLP is applied to each link and the output vectors are then averaged through all links along the route. We use the same output size of the recurrent module for this MLP.

4.3 Evaluation Metrics

We use multiple evaluation metrics in our experiments. For the offline experiment, we adopt three classical metrics, including Mean Absolute Percentage Error (MAPE), Mean Average Error (MAE) and Mean Square Error (MSE), to evaluate the competing methods.

For the online comparison, we use four evaluation metrics, including MAPE, APE20, bad case rate and underestimate rate. The last three metrics are popularly used in industry to evaluate the performance of the real-time ETA service. Their detailed definitions are listed below:

- APE20: the percentage of predictions with absolute percentage error (APE) less than 20% (higher the better).
- Bad case rate: the percentage of predictions with APE greater than 50% or absolute error (AE) greater than 180 seconds, measuring the percentage of extreme bad cases (lower the better).
- Underestimate rate: the percentage of underestimated predictions (lower the better).

4.4 Offline Results

We implement the WDR model in Python using TensorFlow toolbox, and parallelize the training on 4 NVIDIA Tesla P100 GPU cards. A typical training process takes 10 hours on the pickup dataset and 45 hours on the trip dataset. Table 2 and Table 3 summarize the competing results, which show that our WDR learning solution outperforms all competitors on both pickup and trip datasets.

We observe that the pure machine learning methods (WDR, GBDT, FM and WD-MLP) under our framework outperform the route-based method (route-ETA). This confirms that we can accurately predict the travel time by using machine learning models based on large-scale historical data. As a route-free method, TEMP$_{\text{rel}}$ does not make full use of available information in the historical data. Thus its performance is not satisfactory. The results also show that WDR outperforms PTTE. This may be due to the data sparsity, as the performances of tensor completion are heavily influenced by data sparsity. In the original work of PTTE [20], they collected data from Beijing taxicabs which have 0.4% non-zero values in the historical tensor. However, we have only 0.012% non-zero values when constructing the tensor using our floating-car data.

We evaluate the influence of modeling the sequential information along road segments by comparing the WDR and WD-MLP models. These two models share the same set of features and the same WD network structure. The only difference is that WDR models the road segment sequence by using a shared RNN structure, but WD-MLP models each segment separately by using an MLP structure. The WDR model outperforms the WD-MLP model by an MAPE gap of 0.75% on pickup dataset and 1.77% on trip dataset, which confirms the benefit of introducing the recurrent neural network structure. This evidence also shows that the benefit on the trip dataset is larger than the pickup dataset. A main reason is that the trip route is often longer than the pickup route, and the sequential dependency information helps more when there are more segments in the route. We also find that the fine-tuning of our model introduces 0.5% gain in terms of MAPE.

In Figure 7 and Figure 8, we plot the MAPE curves over different departure time periods during a day. The pattern of MAPE in different hours provide us a more detailed demonstration of the model performance. These figures show that our model has a better performance during off-peak hours compared with rush hours, which is

[2]The results obtained on these subsets are marked by * in the experiments.

Table 2: The results on the pickup dataset. The results marked by * are obtained on a subset required by TEMP$_{rel}$.

	MAPE	MAE	MSE
route-ETA	31.27%	88.0	16555.8
PTTE [20]	29.35%	83.3	13153.5
GBDT	23.64%	68.3	11674.2
FM	21.41%	63.6	9664.2
WD-MLP	21.58%	64.1	9816.3
WDR	**20.83%**	**59.9**	**9078.2**
*TEMP$_{rel}$ [19]	35.30%	79.7	15480.7
*WDR	**21.68%**	**55.6**	**7962.0**

Table 3: The results on the trip dataset. The results marked by * are obtained on a subset required by TEMP$_{rel}$.

	MAPE	MAE	MSE
route-ETA	15.01%	153.2	66789.8
PTTE [20]	14.78%	150.1	66141.2
GBDT	14.01%	133.2	52006.6
FM	12.87%	122.5	47457,3
WD-MLP	13.43%	127.5	50012.8
WDR	**11.66%**	**112.4**	**37482.7**
*TEMP$_{rel}$ [19]	23.53%	166.8	88767.1
*WDR	**12.27%**	**87.2**	**20929.3**

consistent with common sense that traffic condition in rush hours are unpredictable. Usually there is no morning rush hours at the weekends or during holidays. However, a morning MAPE peak is observed on trip data during the weekends in our experiment. This is caused by the irregular increase of traffic volume in the morning in labor day holidays during the period of data collection, which happens to be in a weekend this year. The result in Figure 7 shows that the performance pattern for the pickup data is more regular.

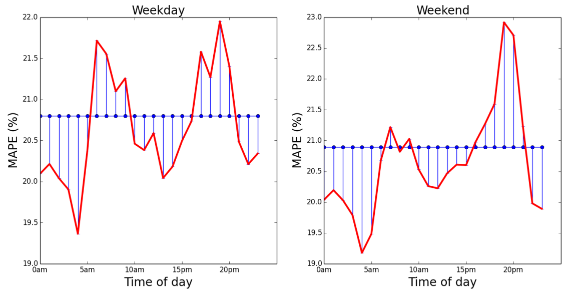

Figure 7: The trend of MAPE over time during a day (pickup). The blue baselines represent MAPE of 20.80% and 20.89% for the weekday and weekend, respectively.

Besides the temporal pattern, we further analyze the influence of the trajectory length. Figure 9 presents the curve of MAPE over the segment number in the route. In most cases, a route containing

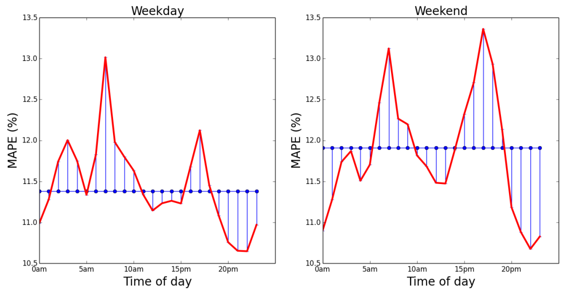

Figure 8: The trend of MAPE over time during a day (trip). The blue baselines represent MAPE of 11.38% and 11.91% for the weekday and weekend, respectively.

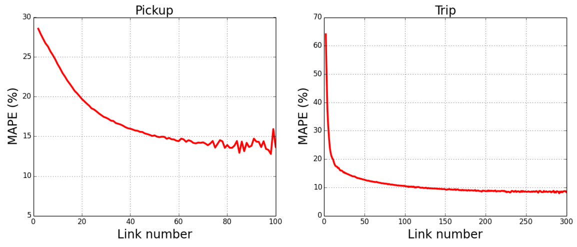

Figure 9: The trend of MAPE over the segment number. The MAPE decreases as the segment number grows.

more road segments is longer. For our WDR model, the MAPE decreases as the segment number grows.

Route deviation: It is known that the actual travel time may vary if the driver chooses a different route for the same origin and destination pair. However, in real applications we cannot know the exact route before the trip starts. Though this is beyond the scope of this paper, we adapt our solution to improve the ETA prediction accuracy by considering the influence of route planning. Instead of extracting the features along the historical GPS trajectory, we extract the features along a pseudo route which is provided by a route planner for each origin and destination pair. We denote the model trained on historical GPS trajectories as WDR-GPS, and denote the model trained on pseudo routes as WDR-RP. The results are obtained on the test dataset with the features extracted along the pseudo routes given by the route planner. The comparison results in Table 4 show that WDR-RP obtains much better results than WDR-GPS.

Table 4: Comparison Results of WDR-GPS vs. WDR-RP.

	pickup MAPE	trip MAPE
WDR-GPS	30.01%	19.17%
WDR-RP	22.69%	14.86%

4.5 Online Results

Besides the offline experiments, we also evaluate the performance of the real-time service that we build based on the proposed WDR solution on DiDi platform. We compare our solution with three leading industry LBS providers on our test platform. We use the names of com1, com2 and com3 for the three competitors. In the experiment, we randomly sample the real time order requests and query the ETA from the competing services. The prediction accuracy is then obtained by comparing with the recorded ground-truth travel time after all the orders are finished.

We plot the value of MAPE, APE20, bad case rate and underestimate rate for all competing services during 7 days in Figure 10. The results show that our solution provides significantly lower MAPE and higher APE20. In real-world scenarios, user experience will be badly hurt by extreme cases. Thus, we compare the bad case rate. The results show that our system obtains state-of-the-art performance in this metric. The bad cases in our system are mainly caused by very short routes, which usually have higher MAPE in Figure 9. The competing results also show that our system obtains a low level of underestimate rate, sharing the leading position with one of the competitors.

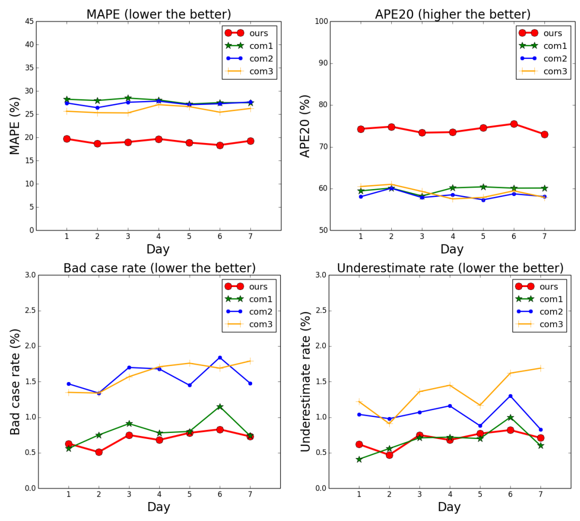

Figure 10: Online competing results in terms of MAPE, APE20, bad case rate and underestimate rate.

5 CONCLUSION

In this paper, we propose a systematic machine learning solution for travel time estimation. To overcome the drawbacks in the existing methods, we establish a rich and effective feature system for the location-based data and represent each route as a high dimensional feature vector. Based on the this well-designed feature representation, we formulate ETA as a pure regression problem. We first adopt several existing regression algorithms to solve the problem, then we build a novel deep learning model to solve this problem. The model includes wide, deep and recurrent components, which are designed to handle the high dimensional sparse features, the real value features and the road segment features. We evaluate our solution offline with large-scale vehicle travel data. We also evaluate the proposed solution by building a real-time service system on DiDi platform. Both results show that the proposed model is more powerful than the existing deep learning models in solving the ETA learning problem. We believe that the accuracy of ETA learning will be further increased by enriching the feature system and introducing more powerful learning model. Though the novel deep learning model is proposed for ETA learning, this model could be applied for general regression problems. We plan to adopt this model to solve other important location-based problems in the future.

REFERENCES

[1] Pouria Amirian, Anahid Basiri, and Jeremy Morley. 2016. Predictive analytics for enhancing travel time estimation in navigation apps of Apple, Google, and Microsoft. In *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science.*

[2] Mohammad Asghari, Tobias Emrich, Ugur Demiryurek, and Cyrus Shahabi. 2015. Probabilistic Estimation of Link Travel Times in Dynamic Road Networks. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representation (ICLR '15).*

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16).*

[5] Heng-Tze Cheng, Levent Koc, et al. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '16).*

[6] Yanjie Duan, Yisheng Lv, and Fei-Yue Wang. 2016. Travel time prediction with LSTM neural network. In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC '16).*

[7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12 (jul 2011), 2121–2159.

[8] C. de Fabritiis, R. Ragona, and G. Valenti. 2008. Traffic Estimation And Prediction Based On Real Time Floating Car Data. In *2008 11th International IEEE Conference on Intelligent Transportation Systems.*

[9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Erik Jenelius and Haris N Koutsopoulos. 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological* 53 (2013), 64–81.

[12] Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR '15).*

[13] M. Kormáksson, L. Barbosa, M. R. Vieira, and B. Zadrozny. 2014. Bus Travel Time Predictions Using Additive Models. In *IEEE International Conference on Data Mining (ICDM '14).*

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS '12).*

[15] Wang-Chien Lee, Weiping Si, Ling-Jyh Chen, and Meng Chang Chen. 2012. HTTP: A New Framework for Bus Travel Time Prediction Based on Historical Trajectories. In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

[16] H. B. Mcmahan. 2016. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS '11).*

[17] Steffen Rendle. 2010. Factorization machines. In *IEEE 10th International Conference on Data Mining (ICDM '10).* IEEE, 995–1000.

[18] Raffi Sevlian and Ram Rajagopal. 2010. Travel Time Estimation Using Floating Car Data. *arXiv preprint arXiv:1012.4249* (2010).

[19] Hongjian Wang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. 2016. A simple baseline for travel time estimation using large-scale trip data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

[20] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel Time Estimation of a Path Using Sparse Trajectories. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14).*

[21] Jiancheng Weng, Chang Wang, Hainan Huang, Yueyue Wang, and Ledian Zhang. 2016. Real-time bus travel speed estimation model based on bus GPS data. *Advances in Mechanical Engineering* 8, 11 (2016), 1687814016678162.

[22] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee. 2004. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems* 5, 4 (Dec 2004), 276–281.