Intent-aware Audience Targeting for Ride-hailing Service

Yuan Xia[2]*, Jingbo Zhou[1,3]**, Jingjia Cao[2,5], Yanyan Li[1,3], Fei Gao[2], Kun Liu[2], Haishan Wu[4], Hui Xiong[1,3]

[1]Business Intelligence Lab, Baidu Research [2]Baidu Inc, Beijing, China
[3]National Engineering Laboratory of Deep Learning Technology and Application, China
[4]SenSight.ai Ltd. [5]Beijing Jiaotong University
{xiayuan,zhoujingbo,caojingjia,liyanyanliyanyan}@baidu.com
{gaofei09,liukun12,xionghui01}@baidu.com, hswu85@gmail.com

Abstract. As the market for ride-hailing service is increasing dramatically, an efficient audience targeting system (which aims to identify a group of recipients for a particular message) for ride-hailing services is demanding for marketing campaigns. In this paper, we describe the details of our deployed system for intent-aware audience targeting on Baidu Maps for ride-hailing services. The objective of the system is to predict user intent for requesting a ride and then send corresponding coupons to the user. For this purpose, we develop a hybrid model to combine the LSTM model and GBDT model together to handle sequential map query data and heterogeneous non-sequential data, which leads to a significant improvement in the performance of the intent prediction. We verify the effectiveness of our method over a large real-world dataset and conduct a large-scale online marketing campaign over Baidu Maps app. We present an in-depth analysis of the model's performance and trade-offs. Both offline experiment and online marketing campaign evaluation show that our method has a consistently good performance in predicting user intent for a ride request and can significantly increase the click-through rate (CTR) of vehicle coupon targeting compared with baseline methods.

Keywords: Audience Targeting · Location Based Service · Marketing Campaign

1 Introduction

With the rapid development of mobile internet, an increasing number of people tend to use mobile phone applications for ride-hailing services (including booking rides and paying for car driver) provided by a transportation network company (TNC) such as Uber, Lyft, Didi and GrabTaxi. Now the ride-hailing service is also an important function in Baidu Maps and Google Maps.

* Yuan Xia and Jingbo Zhou are co-first authors.
** Jingbo Zhou is corresponding author.

Given the tremendous amount of users on Baidu Maps, we aim to attract as many users as possible to use the ride-hailing service provided by Baidu Maps. To achieve this purpose, a useful method is to resort to audience targeting, which identifies a group of recipients for a particular message. Audience targeting is well known to be very useful in message propagation and marketing promotion. Therefore, in order to increase the overall flow of the ride-hailing platform and attract more customers, an accurate audience targeting system for ride-hailing services is demanding. Here, the specific problem is vehicle coupon targeting which aims to deliver coupons to potential users.

Typically, there are mainly two approaches to audience targeting. The first one is audience segments targeting which aims to identify a specific group of people, e.g., the man between the ages of 18 and 25. The second approach is audience behaviors targeting [20] which tries to push customized message according to user's online or offline behaviors. There are also mature architectures and methods for traditional audience targeting problems, like ads targeting in Google [19], Facebook [13] and Bing [8].

Our work mainly focuses on the area of audience targeting for ride-hailing services. Different from the traditional general framework for ads targeting, there are several challenges. First, the types of the coupon are diverse. For a specific service, we usually have different coupons with a short lifetime. It is computationally expensive to train different models for each kind of coupons. Second, the timeliness and context are also important. Users may not be interested in any vehicle coupon if they do not have intention for requesting a ride. The vehicle coupon targeting is also a kind of instant service, we need to push the customized message (e.g., coupon) to the targeted user before his/her movement.

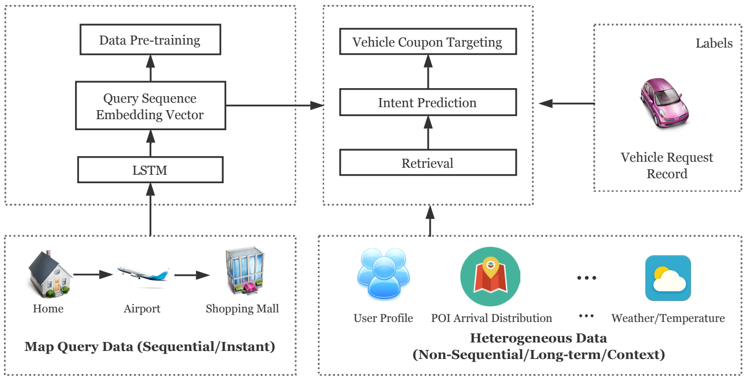

Fig. 1. Illustration of the Audience Targeting System for Ride-hailing Service

In this paper, we propose an intent-aware audience targeting system for ride-hailing services. The key idea to solve this problem is to precisely predict the user's intent for requesting a ride and then send the corresponding coupons. The overall architecture of the system is shown in Figure 1. First, our insight is that map query data on Baidu Maps reflects the instant travel intent of a user. In order to capture user's instant intent and better handle sequential map query data, we do data pre-training and generate query sequence embedding with the user's map query data based on the Long-Short Term Memory (LSTM) [15] model. Second, for the comprehensive understanding of user's intention, our method combines the LSTM and gradient boosting decision tree (GBDT) [6] to deal with multi-source heterogeneous data (sequential, non-sequential, instant, long-term and context data). Third, to achieve real-time performance, before the intent prediction model, we develop a coarse retrieval module based on the frequent pattern mining method. The main contributions of this paper are summarized as follows:

– We propose an intent-aware audience targeting framework for ride-hailing services which first captures the user's instant intent and then delivers coupons to particular users.
– We develop a method to capture the user's instant intent by extracting implicit intention from map query sequential data. By combining the LSTM model and GBDT model, we can deal with both map query sequential data and other multi-source heterogeneous non-sequential data. We demonstrate that the combined model can significantly improve the overall prediction performance.
– We conduct extensive experiments to evaluate the effectiveness of our framework. Both offline experiments and online marketing campaign evaluation show that our model has a consistently good performance for predicting the user's intent for requesting a ride. Our method can significantly improve the CTR of vehicle coupon targeting: compared to simple baseline our system can improve the CTR by more than 7 times; and compared to the almost optimal competitor our system can still improve the CTR by 26%.

2 Related work

In recent decades, audience targeting has been a popular research topic with many industrial applications. The audience targeting system can deliver personalized ads, while the traditional forms of advertising media are passive in nature. With the development of smartphone handsets, modern audience targeting systems now have the capability to deliver personalized advertising messages to consumers at the proper time and place. In the light of this new mobile capability, how to effectively and efficiently deliver a customized message (like an ad or a coupon) to meet consumers' needs has become a critical issue [4, 17].

There are two types of targeting methods which are prevalent in industrial applications: user segment and behavioral targeting (BT). BT provides an approach to learning the targeting function from past user behavior, especially using implicit feedback (i.e., ad clicks) to match the best ads to users [5]. Therefore, BT is generally used for improving the influence of the online advertising by targeting the most relevant user for the ads being displayed and vice versa [28].

In [20], the authors investigated the effect of historical user activity for targeting. As an effective method, BT has attracted much research attention from different perspectives, such as the scalability [1,18], privacy [7], user identification [12] and ad display [25]. The human mobility prediction methods [33,34] can also be used to improve the performance of the BT. Though our system can be considered as a special case of behavior targeting, we focus on the instant intent prediction to identify the audience with potential demand for picking a ride.

3 Preliminaries

3.1 Problem Definition

We define the problem as follows. For each user $u \in \mathcal{U}$, we want to predict the user's intent for requesting a ride in the next T hours, and then we push the vehicle coupon $c \in \mathcal{C}$ through Baidu Maps app. The main purpose of our targeting system is to improve the CTR of coupon targeting and attract more customers to use Baidu Maps app platform for ride-hailing service.

In this problem, we mainly have two kinds of data: sequential data $\boldsymbol{D_{sq}}$ and non-sequential data $\boldsymbol{D_{non-sq}}$. The $\boldsymbol{D_{sq}}$ is mainly user's map query sequence data $\{q_1, q_2, \cdots, q_t, \cdots\}$, and the non-sequential data includes user's profile data, vehicle historical request data, user's Points of Interests (POI) arrival distribution data, temperature and weather data, etc. Based on all these data, we want to build models to deal with the problem of vehicle coupon targeting. Specifically, the vehicle request record $y \in \{0, 1\}$ represents the fact that whether a user has actually requested a ride in T hours. Given the sequential data x_{sq}, non-sequential data x_{non-sq} and the vehicle request record $y \in \{0, 1\}$, we can model a multivariable non-linear function $\boldsymbol{\Psi}(\cdot)$ to calculate the probability $p_u \in [0, 1]$ which represents the user's intent for requesting a ride. After obtaining the probability, we can push the corresponding vehicle coupon according to a predefined threshold τ. The coupon type is determined by the marketing team.

3.2 Data Specification

To deal with the audience targeting for the ride-hailing service on Baidu Maps, our work mainly focuses on the following data sources:

- **User Map Query Data.** The data records the user's query search behavior on Baidu Maps app. When a user is searching for a place at Baidu Maps, it will record query word, destination POI information, user id, user's current coordinates, query destination coordinates and the current timestamp.
- **User Profile Data.** The data makes speculations on user's profile, including the user's age, gender, consumption level, job, education level, life stage, interest, etc.
- **User Vehicle Request Data.** The data records the user's online vehicle request history. It includes request id, start time, end time, duration, source, destination, source coordinates, destination coordinates, distance, price and discount.

– **POI Arrival Data.** The data records the user's POI arrival information in the recent period of time. Different from the map query data which represents the user's current intent, the POI arrival information records the user's POI arrival distribution. For example, one record can be as follows: Tom visited "Shopping Mall" 3 times, visited "Restaurant" 4 times, and visited "Hotel" 2 times on Oct, 2016.
– **Weather and Temperature Data.** The data keeps track of the weather and temperature information in the whole nation. For each item in the database, it records the city name, area name, area code, weather and temperature information. Weather data includes nearly 40 weather types, and temperature data is measured by Celsius format.

4 The Intent Prediction Model

To fully exploit the sequential and non-sequential data to achieve a good performance on audience targeting, we propose an intent-aware ride-hailing service targeting system which takes the advantages of Sequential Pattern Mining, Long Short-Term Memory (LSTM) and Gradient Boosting Decision Tree (GBDT). As for our problem, we do have heterogeneous data from different sources. For travel intent prediction, the user's sequential map query data is essential. The following subsections demonstrate the methodology of our system.

4.1 Sequential Pattern Mining

User's historical map query sequential behaviors on Baidu Maps can implicitly reveal user's intention. There are several well-known sequential mining methods available, such as *PrefixSpan* [11], *FreeSpan* [10] and *GSP Algorithm* [22]. These methods can find out sequential frequent patterns in a user map query database. However, these methods have their own limitations. First, they can only list the frequent patterns for different scenes (the scene here is vehicle coupon targeting), and they cannot give probabilistic output. Second, these methods cannot deal with uncertain data, i.e, if a pattern is unseen before, then these methods will meet problems.

While the sequential pattern mining method is not a good choice for making final predictions, we can use frequent patterns as coarse filters to our system. Given the map query sequence database S and the minimum support threshold *min_sup*, we first adopt the *PrefixSpan Algorithm* for sequential pattern mining to find frequent patterns $p \in \mathcal{P}$ in S. Then, we extract the ride request oriented patterns from the user's vehicle request record. Finally, by eliminating some of the query patterns which have little correlation with requesting a ride, we can boost the speed of the targeting system and get the real-time performance.

4.2 Map query based LSTM Method

Recurrent Neural Network (RNN) [21, 26] has been broadly used to deal with sequential data. The standard recurrent neural network can map a sequence of map queries of variable length to a fixed-length vector. By recursively transforming current map query vector q_t with the previous step vector h_{t-1}, we can get the current step vector h_t. The transition function is typically a linear layer followed by a non-linearity layer such as $tanh$. Given the user's map query sequence $\{q_1, q_2, \cdots, q_t, \cdots\}$, the standard RNN computes the output sequence of q as follows:

$$h_t = tanh(W_{hq} \cdot q_t + W_{hh} \cdot h_{t-1} + b_h) \tag{1}$$

where $W_{hq} \in R^{m \times n}$, $W_{hh} \in R^{m \times m}$ and $b_h \in R^m$. m and n are dimensions of the hidden vector and the query embedding vector, respectively. Unfortunately, standard RNN suffers the problem of gradient vanishing or exploding [3, 15], where gradients may grow or decay exponentially over long sequences. This makes it difficult to model long-distance correlations in a sequence.

The advent of LSTM is to deal with the weakness of standard RNN. There is one memory cell g surrounded by three gates controlling whether to input new data (the input gate i), whether to forget history (the forget gate f), and whether to produce current value (the output gate o) at each time step t. The memory cell in LSTM summarizes the information at each time step of what information has been observed up to now. Such structures are more capable to learn a complex composition of query vectors than standard RNN. The definition of the gates and cell update and output are as follows:

$$i_t = \sigma(W_{ix} \cdot q_t + W_{ih} \cdot h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_{fx} \cdot q_t + W_{fh} \cdot h_{t-1} + b_f) \tag{3}$$

$$o_t = \sigma(W_{ox} \cdot q_t + W_{oh} \cdot h_{t-1} + b_o) \tag{4}$$

$$\widetilde{g}_t = \phi(W_{gx} \cdot q_t + W_{gh} \cdot h_{t-1} + b_g) \tag{5}$$

$$g_t = f_t \odot g_{t-1} + i_t \odot \widetilde{g}_t \tag{6}$$

$$h_t = o_t \odot g_t \tag{7}$$

where $\odot$ represents the product of with a gate value. σ and ϕ are non-linear activation functions. Here, $\sigma(\cdot)$ is sigmoid and $\phi(\cdot)$ is hyperbolic tangent. The W and b are the parameter matrices and bias vectors learned during the training.

For building the LSTM model with the map query, we first conduct a semantic representation transformation for the data. There are usually millions of POIs in a city. The number of the POI in a city is too large to be handled by the LSTM model. Therefore, we introduce the semantic representation transformation to reduce the vocabulary of the input space of the LSTM. For each user $u \in \mathcal{U}$, we collect his/her recently map query records in the past period of time and get the raw query sequence $\{q_r^1,\ q_r^2, \cdots,\ q_r^t, \cdots\}$, which is denoted by q_r. Then, according to the POI information, we map the raw query word into a semantic space with limited dimensions.

$$q_c^t = P(q_r^t), \quad t \in \{1, 2, \cdots, l\} \tag{8}$$

where $P(\cdot)$ projects the raw query q_r^t into the semantic space by categorization methods.For the semantic space mapping, we use the Baidu Maps' API to map

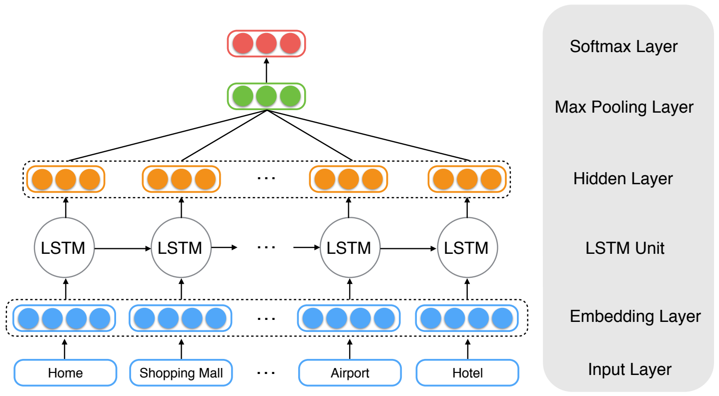

Fig. 2. Architecture of Map Query Based LSTM Model

the POI into its category space. For example, when someone is searching for "JW Marriott Hotel", it will be projected to the semantic space with the tag "Hotel". After projecting the user's raw query sequence into the semantic space, we get $\{q_c^1,\ q_c^2, \cdots,\ q_c^t, \cdots\}$, which is denoted by q_c. For instance, a user first queried for "Beijing Capital International Airport" and then "JW Marriott Hotel", and later queried for "Bird's Nest" and "Quanjude Roast Duck". Then, after mapping the POI name into the semantic space, we get a user semantic query sequence $\{Airport \rightarrow Hotel \rightarrow Landmark \rightarrow Restaurant\}$.

Then, we put user's map query sequence data into LSTM model. The architecture of our LSTM model is illustrated in Figure 2. We use an embedding layer to map the query category id to a fixed length vector.

$$c_t = T(q_c^t), \quad t \in \{1, 2, \cdots, l\} \tag{9}$$

where $T(\cdot)$ transforms the one-hot semantic representation of query q_c^t into the latent vector c_t using a learned embedding matrix W_c.

Then through LSTM, each input of the sequence will generate a hidden output, and at the end of the sequence, we get a final hidden output. Generally, people use the last hidden output as the representation of the whole sequence [2,23].

$$h_t = LSTM(h_{t-1},\ c_t) \tag{10}$$

We believe that for a map query sequence, each query word needs to be considered in order to get a better permanence. In [24], the author added an average pooling layer before the softmax layer to their network structure for merging the sequential hidden state. By adding an extra pooling layer on top of each hidden output, we can get a final output utilizing each hidden query vector. We test two methods for best utilizing the query sequence, including average pooling and max pooling. In the experiment section, we prove that max pooling outperforms the average pooling in this problem.

$$h_o = Merge(h_1, \cdots, h_l) \tag{11}$$

Finally, through the softmax layer, the LSTM outputs the p_u, which indicates the probability of the user's intent on requesting a ride in the next T hours.

$$p_u = softmax(W_d \cdot h_o + b_d) \tag{12}$$

where W_d and b_d are parameters of the dense output layer.

4.3 Combination of LSTM and GBDT

We further propose a hybrid model to combine the LSTM and Gradient Boosting Decision Tree (GBDT) for intent prediction. It is not good enough to predict the intent of a user to order a car for travelling only by map query sequential data. We need more data to help us make predictions. In Baidu, we have all sorts of data from different sources. Long Short-Term Memory (LSTM) is proved to be a good solution for long-term dependencies sequential data problem [9]. However, recurrent neural network based algorithms cannot handle the multi-source heterogeneous data well, and they require significantly more tuning compared with decision tree based methods. However, decision tree based method, for instance, Gradient Boosting Decision Tree (GBDT), is popular when it comes to solving the problem of learning non-linear functions from data, for that it has an important advantage of dealing with heterogeneous data when different features come from different sources. In this section, we explain our hybrid model combing the LSTM and GBDT for intent prediction.

First, we extract the user's map query sequential features, user's profile features, user's query context features, user's POI arrival statistical features, weather and temperature features from different kinds of sources, and then we encode the features into an id-value format. (Please recall the example illustrated in Figure 1). For a user's map query sequence, we get sequential features $\boldsymbol{F_{sq}}$, and for the rest of the features, we get non-sequential features $\boldsymbol{F_{non-sq}}$. Then, we train the LSTM model with map query sequential data. The detail has been illustrated in Section 4.2.

Next, the problem is how to combine the LSTM learned sequential features into the GBDT model. Here, we evaluate two kinds of methods. One is stacking [27] which involves training a learning algorithm to combine the predictions of several other learning algorithms. The other is feature embedding which is a kind of popular method in recent years. In computer vision, people use Convolutional Neural Network (CNN) [16] for image feature embedding. In the recent work [29, 31, 32], the author uses CNN for spatial-temporal data embedding. There are also some works [30] using knowledge base embedding for recommendation system.

Our work applies LSTM for map query sequence embedding. In the experiment section, we show that the feature embedding method outperforms the stacking method. Similar to feature extraction in image classification problem, we keep the output vector before the final softmax layer, and use this feature vector as our LSTM embedding vector $\boldsymbol{E_{sq}}$.

$$\boldsymbol{E_{sq}} = LSTM^e(\boldsymbol{F_{sq}}) \tag{13}$$

Then we put the embedding output of LSTM into the GBDT model. The reason we use GDBT rather than other deep learning based method (like CN-N,DNN) is that GBDT is better when dealing with heterogeneous data. Finally, the GBDT model makes predictions based on two kinds of features, i.e., sequential data and non-sequential data. $\boldsymbol{F_{sq}}$ represents the user's instant features, and $\boldsymbol{F_{non-sq}}$ involves the context features, long-term features and part of instant features.

$$p_u^{'} = GBDT(\boldsymbol{E_{sq}},\ \boldsymbol{F_{non-sq}}) \tag{14}$$

Here, $p_u^{'}$ is different from the p_u in the previous section, since $p_u^{'}$ not only considers instant and sequential characteristics of map query but also utilizes the heterogeneous data from different sources.

5 Experiment

In this section, we evaluate the performance of our model in a real-world dataset from Baidu Maps and compare it with other models. For offline evaluation, we mainly consider the area under the curve (AUC) as our metrics. In next section, we consider the coupon click-through rate (CTR) as a measurement of our online vehicle coupon targeting system.

5.1 Data Description

Using the user id as the key value, we combine the mentioned different sources of data and get our offline training dataset for user vehicle coupon targeting. It consists of 5,986,928 map queries, and covering 1,089,571 users. Each user's map query sequence is appended with the user's non-sequential data. Then we split the dataset as the training set (4,791,755 queries), the validation set (597,417 queries) and the test set (597,756 queries) with ratio 8:1:1. Note that, the dataset is sampled from the whole map query database. We mainly focus on queries which are connected to the user's ride historical request record, because the primary thing we want to do is vehicle coupon targeting. After connecting with the user's ride request history, we get 628,215 map queries which are correlated with ride request history. In other words, the user requests a ride in the next T hours after these query behaviors in Baidu Maps. Here, T is a parameter. We evaluate the T with grid search and find that the model performance is relative stable if T is large enough (like more than one day). In the following experiment, the T is set to 36 hours. These queries are labeled as positive samples. Simply, we can throw all other queries into the training dataset and labeled as negative. However, similar to user purchase behavior in the e-commerce scene, people's behavior on requesting a ride is also quite sparse. In our data set, the negative samples are randomly sampled from the map queries which are not connected to the ride request behavior. After the sampling process, we finally get 5,358,713 negative samples, which means the pos/neg rate is near 1/10.

In the training process, we try different parameters to evaluate our model performance with validation data, and then we use the model which has the best result to evaluate the predictive performance on test data.

5.2 Model Comparison

For the problem of intent prediction, there are several candidate models available. In the next subsection, we demonstrate the reason why we choose to integrate the LSTM and GBDT model for such intent prediction problem and then prove that our model is the best choice for such problem. The following candidate models will be evaluated in the next subsection:

– **Pattern Match Prediction.** Pattern match model mainly utilizes the frequent patterns we discovered to make predictions. If a user's search queries match the ride request oriented patterns we extracted before, then we predict the user will requests a ride. How to mine the patterns is demonstrated in Section 4.1.
– **GBDT Model.** Gradient boosting decision tree model is a prediction model in the form of an ensemble of decision trees generated by gradient boosting.
– **DNN Model.** Deep Neural Network (DNN) [14] uses a cascade of many layers of nonlinear processing units for feature extraction and transformation.
– **LSTM Model.** LSTM is one of the variant of RNN, and its gated memory structure can deal with the weakness of standard RNN.

5.3 Experiment Results

In this subsection, first, we compare the performance of LSTM, GBDT and DNN when only dealing with sequential data, and then we compare the performance of GBDT and DNN when only dealing with non-sequential data. Note that, through feature preprocess, we can fit non-sequential data into LSTM model, but it makes no sense in the real situation. Second, we demonstrate that our proposed LSTM embedding and LSTM stacking model beat the traditional GBDT model in the real-world dataset. Third, we illustrate that utilizing the heterogeneous data from different sources, including map query data, other non-sequential long-term and context data can significantly improve the performance.

Comparison for Different Data First, we compare the performance of LSTM, GBDT and DNN when dealing with sequential data and non-sequential data, respectively. Our LSTM model architecture is shown in Figure 2. To better clarify the performance of LSTM model, we test the LSTM model with different kinds of architectures and different parameters. We prove that adding an extra pooling layer on top of the hidden layer can lead to better performance in our map query sequential data. We also test the parameters such as batch size, learning rate and embedding dimension. The best result is achieved when using LSTM with a max pooling extra layer, the batch size is 64, the embedding dimension is 256, and the LSTM hidden state dimension is 128. The model is trained with Adam optimizer at the learning rate of 1e-4. The best AUC for LSTM on sequential data is 0.763. We evaluate the sequential and non-sequential data with GBDT and DNN model, respectively. The best result for traditional GBDT is achieved when the depth is 6, the number of trees is 200, and the learning rate is 0.3. For DNN model, we build a three-layer neural network with the batch size being 128 and the hidden unit dimension being 128. We use dropout for the last connected layer, and the other optimizer parameters are the same with LSTM.

The best results of each model are shown in Table 1. It is obvious that LSTM outperforms GBDT and DNN model when handling with sequential data. When dealing with non-sequential heterogeneous data, GBDT and DNN model get similar performance, while GBDT model is slightly better than DNN. However, GBDT is much faster than DNN and requires less feature engineering and parameter tuning. Therefore, in our model, we chooses the combination of LSTM and GBDT.

Table 1. Single Model Comparison for Different Data

Method	Sequential (AUC)	Non-Sequential (AUC)
LSTM	**0.763**	-
GBDT	0.743	**0.754**
DNN	0.739	0.748

Different Model Comparison Second, we compare our proposed model with several other models. We evaluate our model with different trees number N and tree depth d by grid search. The number of boosting trees N is ranging from 100 to 1000, and the depth of a boosting tree d is chosen from $\{3, 6, 9,12\}$. We also try different learning rates η, and η is chosen from $\{0.01, 0.03, 0.1, 0.3\}$. Finally we set the number of trees as 200, the depth as 6 and η as 0.1.

There are two methods for integrating the LSTM model with GBDT model. The one is to use stacking, which considers the final score of LSTM as a feature; the other one is to use feature embedding method. Our method extracts the output of max pooling layer as our embedding vector. The LSTM training parameter is consistent with the previous best configuration. The evaluation results of traditional GBDT, LSTM stacking and embedding method with GBDT are shown in Figure 3.

The comparison results of different models are shown in Table 2. It shows that our model outperforms the Pattern Match prediction, traditional GBDT and DNN model. The LSTM embedding model is superior to the LSTM stacking model, probably because the former one preserves more information about learned map query sequence data.

Table 2. Different Model Comparison

Method	AUC Score
Pattern Match	0.652
DNN	0.850
GBDT	0.863
GBDT + LSTM Stacking	0.878
GBDT + LSTM Embedding	**0.901**

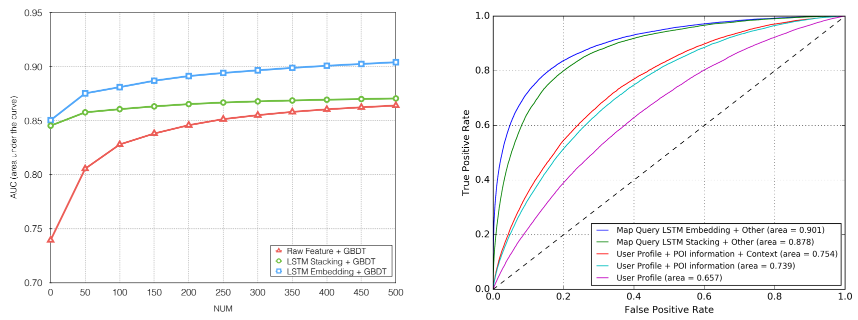

Fig. 3. Left: Receiver Operating Characteristic (ROC) Curves for Feature Integration. The top-left two curves (blue and green curves) show that the integration of map query sequential data and other non-sequential data leads to significant improvements on AUC. Right: Model Comparison (depth = 6). Red triangle, green circle and blue square represent the performance of traditional GBDT, LSTM stacking and embedding with GBDT, respectively.

Significance of Feature Integration Third, we prove that the model performance is significantly improved by integrating heterogeneous data from different sources. Furthermore, the map query sequential data is crucial for the overall performance. To verify above points, we do a four-stage experiment for model evaluation. All the data used in this section is discussed in section 3.2. First, we only use profile data to make predictions. In the second stage, we add the POI arrival information to our model. The data records the user's POI arrival distribution information in recent one month. As the context information can affect user's decision on requesting a ride, therefore, in the third stage, we add the mentioned context data into our model. Finally, in the fourth stage, we add the most important feature, i.e, user's map query sequence data to our model.

The results are shown in Figure 3. From the result, we can see that the user's map query is indeed important, and the integration of map query sequential data and other non-sequential data gives significant improvements on AUC, from 0.754 to 0.878 (LSTM stacking) and 0.901(LSTM embedding), respectively.

6 Online Evaluation

In this section, we evaluate our vehicle coupon targeting system for the real-time online vehicle coupon targeting. The evaluation is based on the Baidu Maps online marketing campaign for vehicle coupon pushing. The campaign aims at attracting more customers to use Baidu Maps app to request a ride (such as taxi, Didi and Baidu cars).

The marketing campaign is mainly launching at the four cities in China, including Guangzhou, Wuhan, Nanjing and Chengdu, and lasts for a month. The coupon has two kinds of types. One is for the Baidu Ordinary Car discount, and the other is for the Baidu Luxury Car discount. In general, the coupon discount on Baidu Luxury Car is higher than Baidu Ordinary Car, for that Baidu Luxury Car is expensive compared to Baidu Ordinary Car. The coupon type is selected by the marketing team. Our proposed model is primarily to tell the system which user should receive vehicle coupons.

We test following two methods as baselines, and refer Click-Through Rate (CTR) as our evaluation metric:

- **Random Targeting.** The random targeting method is the baseline method, which randomly pushes a vehicle coupon to a user.
- **Airport Enclosure.** The airport enclosure method pushes vehicle coupons based on airport environment. The mechanism is that, if the system detects that there is a user whose current location is within 5 kilometers from the airport, then he/she will receive the vehicle coupon. Note that the airport environment has a very close correlation with vehicle coupon targeting, s-ince people appear within airport has strong demand for requesting a ride. Therefore, it can be treated as an almost optimal competitor for our online test.

For our method, we need to determine the predefined threshold τ for pushing coupons to users. Usually, it can be determined by the marketing team according to the actual campaign requirement. However, in this evaluation, in order to make the performance of our method be comparable with the baselines, we set the threshold τ to push coupons to the same percent of users out of the studied group users as the same as the percent of the detected users in airports out of total monthly active users on Baidu Maps app.

The reasons that we use CTR rather than AUC as online evaluation metric are as follows. First, the value of AUC is calculated by its corresponding Receiver Operating Characteristic (ROC) curves, and the ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. While our baseline methods (like Airport Enclosure) cannot output probabilistic result, the output is just binary (pushing or not pushing), which makes its ROC curve not be reasonable. Second, to some extent, the coupon pushing activity can affect the online AUC results, thus using CTR evaluation metric is better. Third, for the business of Baidu Maps, the CTR is the most meaningful metric they care about.

The coupon CTR of our system and two other marketing campaign methods is shown in Table 3. The CTR of our model is 2.67%, which is higher than all the baselines. Note that as for the scene of vehicle coupon targeting, the CTR is hard to improve, for the reason that we may have tremendous of offline data, but the online click data is rare. Our best model can get a 26.5% boost comparing to the airport enclosure baseline, which is a significant improvement.

By retrospecting the results of the campaign, we also analyze coupon CTR on different coupon types. The result is shown in Figure 4. We can see the overall coupon CTR on Baidu Luxury Car is higher than Baidu Ordinary Car (2.79% vs 2.60%). We also find the iOS customer group has a higher CTR than Android customer group (3.05% vs 2.43%, which is not shown in Figure 4).

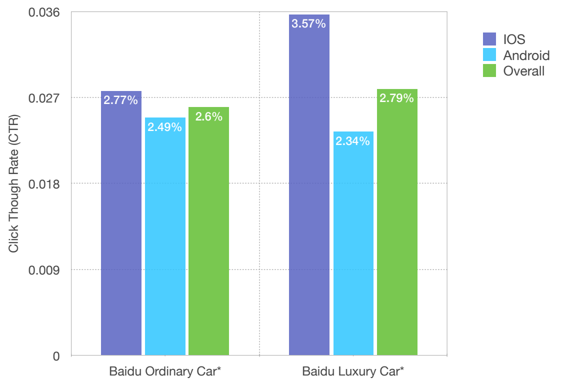

Fig. 4. Click-Trough Rate (CTR) on Different Coupon Types. The left side is Baidu Ordinary Car coupon, and the right side is Baidu Luxury Car coupon. Purple, blue and green bar stands for IOS, Android and overall average CTR, respectively.

Table 3. Online Evaluation Comparison

Methods	*Online CTR*	*Δ CTR*
Random Push	0.32%	-
Airport Enclosure	2.11%	0% -
Our Model	**2.67%**	**26.5%** ↑

7 Conclusion

In this work, we described our deployed intent-aware audience targeting system on Baidu Maps. We demonstrated that the user's sequential map query data implicitly represents the user's intent and is significantly important for such intent prediction problem. In order to better exploit the map query sequential data and other multi-source heterogeneous data, we develop a method to combine the advantages of LSTM model and GBDT model to handle sequential and non-sequential data, and we showed that the combination can lead to significant improvements in the overall prediction performance. We evaluated the performance of our method over a large real world dataset and conducted a large-scale online marketing campaign over Baidu Maps app client to verify the effectiveness of our audience targeting system. The experiment results of the offline evaluation and online marketing campaign demonstrates the effectiveness of our system for predicting the user's intent on requesting a ride, and also exhibited significant improvements in the click-through rate of vehicle coupon targeting.

References

1. Ahmed, A., Low, Y., Aly, M., Josifovski, V., Smola, A.J.: Scalable distributed inference of dynamic user interests for behavioral targeting. In: KDD. pp. 114–122. ACM (2011)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
3. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks **5**(2), 157–166 (1994)
4. Chen, P.T., Hsieh, H.P.: Personalized mobile advertising: Its key attributes, trends, and social impact. Technological Forecasting and Social Change **79**(3), 543–557 (2012)
5. Chen, Y., Pavlov, D., Canny, J.F.: Large-scale behavioral targeting. In: KDD. pp. 209–218. ACM (2009)
6. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
7. Goldfarb, A., Tucker, C.E.: Online advertising, behavioral targeting, and privacy. Communications of the ACM **54**(5), 25–27 (2011)
8. Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In: ICML. pp. 13–20 (2010)
9. Ha, J.W., Pyo, H., Kim, J.: Large-scale item categorization in e-commerce using multiple recurrent neural networks. In: KDD. pp. 107–115. ACM (2016)
10. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.C.: Freespan: frequent pattern-projected sequential pattern mining. In: KDD. pp. 355–359. ACM (2000)
11. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: ICDE. pp. 215–224 (2001)
12. Hao, T., Zhou, J., Cheng, Y., Huang, L., Wu, H.: User identification in cyber-physical space: a case study on mobile query logs and trajectories. In: SIGSPATIAL. p. 71. ACM (2016)

13. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al.: Practical lessons from predicting clicks on ads at facebook. In: ADKDD. pp. 1–9. ACM (2014)
14. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation **18**(7), 1527–1554 (2006)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
17. Li, K., Du, T.C.: Building a targeted mobile advertising system for location-based services. Decision Support Systems **54**(1), 1–8 (2012)
18. Liu, K., Tang, L.: Large-scale behavioral targeting with a social twist. In: CIKM. pp. 1815–1824. ACM (2011)
19. McMahan, H.B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al.: Ad click prediction: a view from the trenches. In: KDD. pp. 1222–1230. ACM (2013)
20. Pandey, S., Aly, M., Bagherjeiran, A., Hatch, A., Ciccolo, P., Ratnaparkhi, A., Zinkevich, M.: Learning to target: what works for behavioral targeting. In: CIKM. pp. 1805–1814. ACM (2011)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Cognitive modeling **5**(3), 1 (1988)
22. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: EDBT. pp. 1–17. Springer (1996)
23. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS. pp. 3104–3112 (2014)
24. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: EMNLP. pp. 1422–1432 (2015)
25. Tang, J., Liu, N., Yan, J., Shen, Y., Guo, S., Gao, B., Yan, S., Zhang, M.: Learning to rank audience for behavioral targeting in display ads. In: CIKM. pp. 605–610. ACM (2011)
26. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proceedings of the IEEE **78**(10), 1550–1560 (1990)
27. Wolpert, D.H.: Stacked generalization. Neural networks **5**(2), 241–259 (1992)
28. Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., Chen, Z.: How much can behavioral targeting help online advertising? In: WWW. pp. 261–270. ACM (2009)
29. Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z.: Deep multi-view spatial-temporal network for taxi demand prediction. In: AAAI (2018)
30. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: KDD. pp. 353–362. ACM (2016)
31. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: AAAI (2017)
32. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dnn-based prediction model for spatio-temporal data. In: SIGSPATIAL. pp. 92:1–92:4. ACM (2016)
33. Zhou, J., Pei, H., Wu, H.: Early warning of human crowds based on query data from baidu maps: Analysis based on shanghai stampede. In: Big Data Support of Urban Planning and Management, pp. 19–41. Springer (2018)
34. Zhou, J., Tung, A.K., Wu, W., Ng, W.S.: A semi-lazy approach to probabilistic path prediction in dynamic environments. In: KDD. pp. 748–756. ACM (2013)