Dynamic Price Prediction in Ride-on-demand Service with Multi-source Urban Data

Suiming Guo
Jinan University
Guangzhou, China

Chao Chen
Chongqing University
Chongqing, China

Jingyuan Wang
Beihang University
Beijing, China

Yaxiao Liu
Tsinghua University
Beijing, China

Ke Xu
Tsinghua University
Beijing, China

Dah Ming Chiu
The Chinese University of Hong Kong
Hong Kong, China

ABSTRACT

Ride-on-demand (RoD) services such as Uber and Didi (in China) are becoming increasingly popular, and in these services dynamic price plays an important role in balancing the supply (i.e., the number of cars) and demand (i.e., the number of passenger requests) to benefit both drivers and passengers. However, the dynamic price also creates concerns for passengers: the "unpredictable" prices sometimes prevent them from making quick decisions at ease. One may wonder if it is possible to get a lower price if s/he chooses to wait a while. Giving passengers more information helps to tackle this concern, and predicting the prices is a possible solution.

In this paper we perform dynamic price prediction based on multi-source urban data. Price prediction helps passengers understand whether they could get a lower price in neighboring locations or within a short time, thus alleviating their concerns. The prediction is based on urban data from multiple sources, including the RoD service itself, taxi service, public transportation, weather, the map of a city, etc. The rationale behind using multi-source urban data is that the dynamic price in RoD may be influenced by different factors found in different data sources. We train a neural network to perform the prediction, and evaluate the prediction accuracy of using different combinations of multi-source urban data. Our results show that using multi-source urban data indeed helps improve the prediction accuracy, and different datasets may have varying influences on the dynamic prices.

CCS CONCEPTS

• **Information systems** → **Location based services**; • **Human-centered computing** → **Empirical studies in ubiquitous and mobile computing**; • **Computing methodologies** → *Machine learning approaches*;

KEYWORDS

Price prediction, ride-on-demand service, dynamic pricing, multi-source urban data

ACM Reference Format:
Suiming Guo, Chao Chen, Jingyuan Wang, Yaxiao Liu, Ke Xu, and Dah Ming Chiu. 2018. Dynamic Price Prediction in Ride-on-demand Service with Multi-source Urban Data. In *EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3286978.3286992

MobiQuitous '18, November 5–7, 2018, New York, NY, USA

ACM ISBN 978-1-4503-6093-7/18/11... $15.00
https://doi.org/10.1145/3286978.3286992

1 INTRODUCTION

Emerging Ride-on-demand (RoD) services such as Uber and Didi are becoming increasingly popular in recent years. They attract passengers by their convenience, as well as flexible and affordable prices; and attract drivers who want to drive more flexibly with their own cars.

Dynamic pricing is the core and distinctive feature in RoD service, and it reflects the effort in balancing the supply (the number of cars on the road) and demand (the number of passengers' requests): a higher price reduces demand and increases supply in a busy area, and a lower price does the opposite in a non-busy area. This makes the service more responsive for both drivers and passengers. Specifically, dynamic pricing is always represented by a "price multiplier": the price of a trip is the product of the price multiplier (based on the supply & demand condition nearby) and a fixed normal price (based on the estimated distance & time of the trip). The fixed normal price is similar to the price of a taxi trip, so we only focus on the price multiplier in this paper.

Despite the convenience and flexibility, dynamic pricing exerts mental burden on passengers and makes them less satisfied. In traditional taxi service with fixed pricing, passengers can estimate the trip fare based on personal experience. In emerging RoD service, however, they have an extra task before making decisions: guessing the price multipliers based on their estimate of the supply & demand condition nearby. For an individual passenger without relevant information, the estimate is invariably inaccurate and usually prevents them from making decisions at ease. Giving more information to passengers help ease the burden, including, for example, explaining why the current price is high or low, giving a recent history of prices to passengers, predicting the prices in the next time slot in the neighboring locations, etc. Among them, the most direct one is price prediction, and passenger could rely on the results to make quick decisions.

Dynamic price prediction has not received much attention in RoD services. In general, there are two ways of predicting prices. One way is to predict the supply & demand and then guess the relationship between dynamic prices and the supply & demand, as seen in [7]. Because most RoD services keep their dynamic pricing algorithms as secrets, guessing this relationship from data is not accurate enough to generate a good prediction, and is hard to generalize because different service providers may emphasize various factors in their algorithms. Furthermore, the prediction of supply & demand itself also brings some inaccuracies. Another way omits the details in between, and predicts the price multiplier directly based on historical data, including the price multipliers and features relevant to supply & demand. This way does not try to unveil the "secret algorithms", and is easier to generalize: the prediction is achievable as long as one can collect historical data, regardless of the service provider-specific algorithms. Our past work [11] uses this methodology to predict dynamic prices, but only at the granularity of a city cell or functional area. We also do the prediction in this manner here.

We extract relevant features and predict the dynamic prices based on multi-source urban data, including data from RoD service itself, taxi service, public transportation, weather, the map of a city, etc. The reason of using multi-source data is that the dynamic prices in RoD service may be influenced by many features from different data sources other than those in RoD service. For example, the number of taxis (or available taxis) around may influence the dynamic price a passenger obtains. As some other examples, weather condition (e.g., temperature, visibility, wind speed, whether there is rain), location information (e.g., POIs around), and the availability and distribution of public transportation services (e.g., whether there are buses or metro around) all may have some impacts on the dynamic prices in RoD service. We thus believe that using multi-source urban data can achieve higher prediction accuracy than only using data from RoD service.

In this paper, we address the dynamic price prediction problem by training a neural network based on multi-source urban data. We collect urban data from multiple sources: the RoD service itself, taxi service, public transportation, weather, the map of a city, etc. We then evaluate the prediction accuracy of using different combinations of data sources. Comparing the prediction accuracy under different scenarios, i.e., "one combination of data sources brings a certain amount of improvement over another combination", can give us a rough picture about what features have the highest influence on the dynamic prices in RoD service.

As to our contribution, this is the first effort to involve multi-source urban data to predict dynamic prices in RoD service. Specifically, by using multi-source urban data, we not only try to achieve better prediction accuracy, but also paint a rough picture on the levels of influence different features have on the dynamic prices in RoD service. We hope our effort could help the industry or policy makers to understand more about the dynamic prices in emerging RoD services, and about how the RoD service interacts with other public transportation services.

The remainder of the paper is organized as follows. In §2 we describe the multi-source urban datasets used in the study, and §3 introduces the features we use. In §4 we present the neural network model, evaluate our model with different combinations of datasets and discuss the influence these datasets have on the dynamic prices in RoD service. Finally, §5 presents related work and §6 concludes the paper.

Table 1: A summary of datasets and fields.

Dataset	Fields
RoD	event_time, event_location, estimated_fare, price_multiplier, passenger_device_IMEI.
Taxi	upload_time, latitude, longitude, heading, speed, full_flag, car_plate.
Bus & metro	# of bus stations, # of bus lines, # of metro stations, # of metro lines.
POI	# of POIs of 14 categories (*car service*, *restaurant*, *shopping*, *sports* & *entertainment*, *hospital*, *hotel*, *scenic spot*, *residence* & *apartment*, *government*, *education* & *culture*, *transportation facility*, *finance* & *insurance*, *business* and *everyday life*).
Weather	temperature, wind speed, humidity, pressure, visibility, weather condition.

2 MULTI-SOURCE URBAN DATASETS

We present the multi-source urban datasets used in predicting price multiplier, including the event-log data from a RoD service, the GPS trajectory data from taxi service, the bus & metro distribution data, the POI data and the weather data. Tab. 1 summarizes our datasets and their fields.

2.1 RoD Service Event-log Data

Our data of the RoD service is collected from Shenzhou UCar (http://www.10101111.com/), one of the major RoD service providers in China. By the end of 2015, Shenzhou UCar's service covers more than 50 cities in China, with a fleet of more than 30,000 cars, offering more than 300,000 trips per day [26].

We first explain the user interface of the mobile app, as shown in Fig. 1, to illustrate the work-flow of a typical RoD service. A user usually opens the app and types the boarding location A and arriving location B. S/he could also choose "when to ride (now or several minutes later)" and "using coupon". After the user has specified the locations and chosen all available options, the app sends the relevant information back to the service provider and obtains (a) the estimated trip fare and (b) the current dynamic price multiplier, which are displayed to the user. Note that the service provider often sets a lower and upper bound on the price multiplier in the service policy. The user then chooses either to accept the current price (by pressing "Ride a Car!" button) or give up the current fare estimation if s/he considers the price multiplier too high.

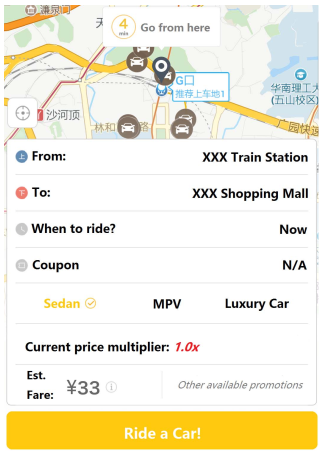

Figure 1: The user interface of a typical RoD service.

Each time when the mobile app sends all the information to the service provider and returns the current price multiplier and the estimated trip fare, an ***EstimateFee*** event is generated, and this is the source of our event-log dataset. Our dataset contains the complete record of ***EstimateFee*** events in the complete 4 months from Aug to Nov, 2016 in Beijing. Each entry corresponds to a single event, and includes fields such as *event_time*, *event_location* (longitude and latitude), *estimated_fare*, *price_multiplier*, *passenger_device_IMEI* (i.e., an unique identifies of a passenger), etc. The dataset contains 14,587,353 entries, and all are properly anonymized.

In the dataset, we find out that the service provider sets a lower and upper bound for the price multiplier. The lower bound is $m = 1.0$ and the upper bound is $U = 1.6$. So all possible multipliers are 1.0, 1.1, 1.2, 1.3, 1.4, 1.5 and 1.6.

In this subsection, we present some results that show how different price multipliers could be in different locations or during different time periods. We first show the variation of hourly average price multiplier at the level of city functional areas. We select some typical business (i.e., the place for working), residential (i.e., the place for living) and transportation (e.g., airport terminals and railway stations) areas in Beijing, and show the variation of hourly average price multipliers in Fig. 2. The criteria of selecting these typical functional areas could be found in our previous work [12] and is not discussed here.

We then divide the map of Beijing into 2500 ($= 50 * 50$) rectangular cells of the same size, and investigate the average price multiplier of all events taking place in each cell during

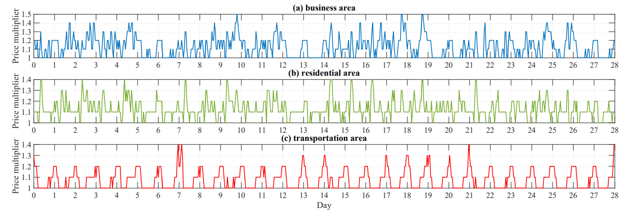

Figure 2: The variation of hourly average price multipliers in different functional areas.

some particular time periods. Fig. 3 to 5 show the average price multiplier during morning rush hour (i.e., 8am to 9am), a non-busy hour (i.e., at noon) and evening rush hour (i.e., 6pm to 7pm) on weekdays. By comparison, the average price multiplier during [8am, 9am] on weekends is shown in Fig. 6. We also show the number of ***EstimateFee*** events taking place in each cell during morning (Fig. 7) and evening rush hour (Fig. 8) on weekdays. Note that the number of events is an approximate of the total demand, i.e., the sum of met and unmet demand from passengers. In Fig. 3 to 8, the deeper the color of a cell, the larger the corresponding metric (i.e., the average price multiplier or the number of events).

There are some basic observations from these figures:

- The regularity of the variation of price multipliers is closely related to the locations of passengers. In some location (e.g., transportation area) the variation is more regular, whereas in some locations (e.g., business area) it is more random.
- The average price multiplier of a location is related to hour-of-day, day-of-week, and the location itself.
- Passengers' potential demand (i.e., the number of ***EstimateFee*** events) also varies significantly in different locations, hour-of-day, and day-of-week.

2.2 Taxi Service GPS Trajectory Data

Taxi is a major competitor of RoD service, and we also collect GPS trajectory data from the taxi service in Beijing. The taxi data helps us to (a) capture the operating status of taxi service in the city and (b) characterize the general traffic condition of different locations. Examples include "*whether a region is busy during a particular time period*" or "*the number of available taxis around a location*".

Our dataset covers the GPS trajectory data of about 30,000 taxis in Beijing in November, 2016. Each taxi uploads one GPS data entry every 30 seconds during operation. For each day, the volume of dataset ranges from 45 to 50 million entries. Each entry contains the following fields:

- *upload_time*: the timestamp of this entry;
- *latitude* & *longitude*: the location of the taxi;
- *heading* & *speed*: the heading and driving speed of the taxi;
- *full_flag*: whether the taxi is full or available;
- *car_plate*: the MD5-encrypted string of the taxi's plate number.

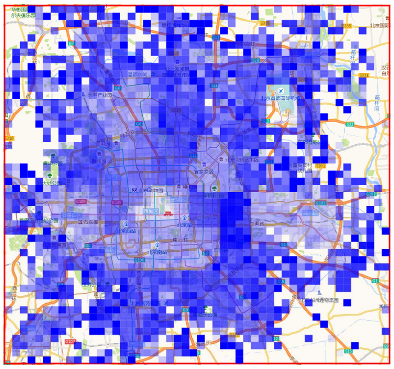

Figure 3: The average price multiplier during [8am, 9am] on weekdays.

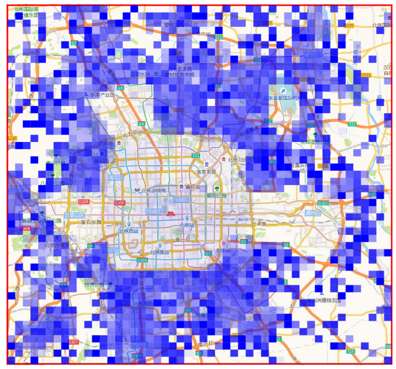

Figure 4: The average price multiplier during [12pm, 1pm] on weekdays.

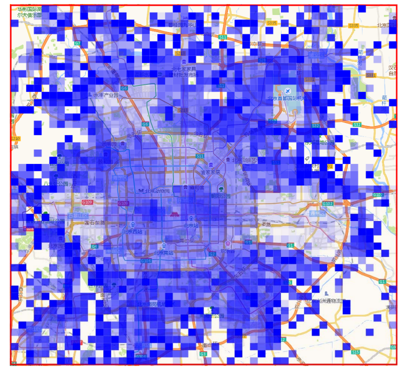

Figure 5: The average price multiplier during [6pm,7pm] on weekdays.

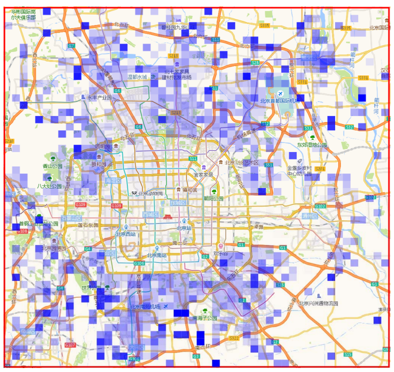

Figure 6: The average price multiplier during [8am, 9am] on weekends.

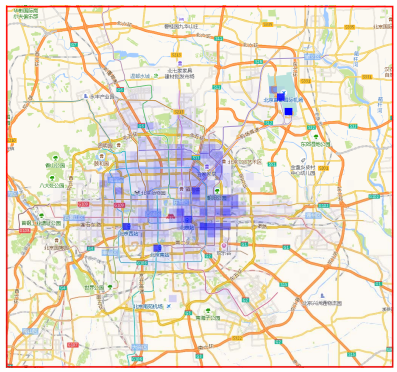

Figure 7: The number of *Estimate-Fee* events during [8am, 9am] on weekdays.

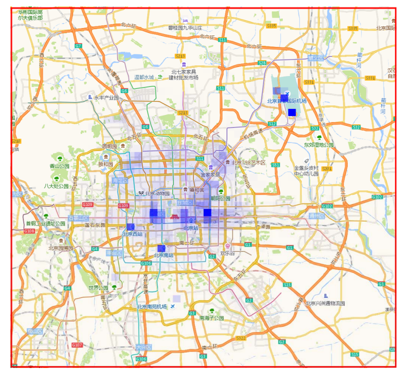

Figure 8: The number of *Estimate-Fee* events during [6pm,7pm] on weekdays.

With GPS trajectory and especially the *full_flag* of a taxi, we can determine all the trips a particular taxi serves each day. Specifically, the *full_flag* changing from "available" to "full" indicates that a passenger is getting on a taxi; and the reverse direction indicates that a trip is finished.

2.3 Bus and Metro Distribution Data

The distribution of bus & metro helps to characterize the availability of public transportation around different locations, and this may have impacts on RoD service.

The most accurate description of the bus and metro distribution should be like "*the number of buses around a particular location during a particular time period*", and could be obtained by, for example, examining the smart-card usage data (i.e., "*how many people wipe their smart-card on a bus*") or collecting the GPS data of bus & metro. However, bus & metro have relatively fixed time tables, and most people decide whether to take public transportation based on the availability of bus & metro lines/stations nearby, instead of the availability of bus & metro nearby. So we turn to an easier method to acquire our datasets by simply counting the number of bus & metro lines and stations nearby.

We crawl the above data from AMap service (one of the most popular digital map service providers in China) using its JavaScript API [2]. Specifically, for a location (i.e., a pair of longitude and latitude) given in an entry of the RoD service dataset, we count the number of bus & metro lines and stations within a 500-meter radius of the location. As a result, the volume of this dataset is the same as that of the RoD service dataset. For the whole city, there are more than 7,700 bus stations and about 380 metro stations, and we plot the distribution of bus & metro stations in Beijing in Fig. 9 and 10 with each point as a bus or metro station.

2.4 POI Data

This dataset mainly contains the POI (point of interest) information. The goal of using POI information is that we hope some properly selected POI features could represent the location information given a pair of longitude and latitude. As shown in Fig. 3 to 8, either the price multiplier or the number of events is closely related to the location in which

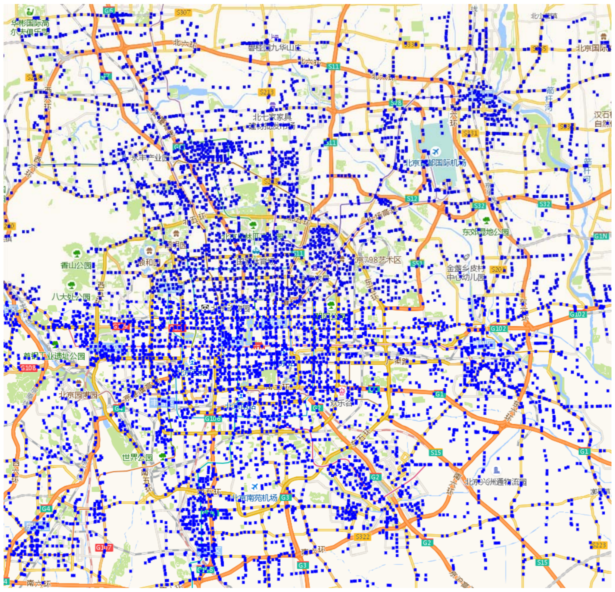

Figure 9: The distribution of bus stations in Beijing.

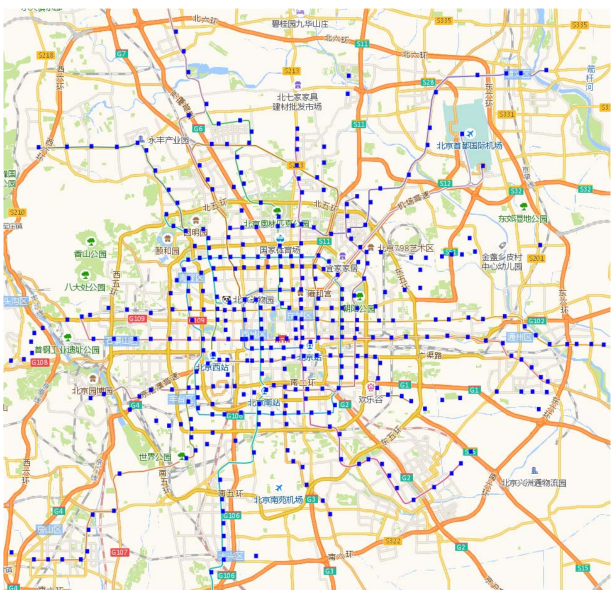

Figure 10: The distribution of metro stations in Beijing.

an ***EstimateFee*** event takes place. For example, the number of events is significantly higher around airport terminals or railway stations; the price multiplier is, on average, much higher in some business areas during evening rush hour than in other locations. We seek for some features to accurately describe this sort of location information.

Similar to the bus and metro distribution data, we also crawl POI data from AMap service. This map service provider categorizes each POI on the map into 14 coarse categories: *car service*, *restaurant*, *shopping*, *sports & entertainment*, *hospital*, *hotel*, *scenic spot*, *residence & apartment*, *government*, *education & culture*, *transportation facility*, *finance & insurance*, *business* and *everyday life*. For a location (i.e., a pair of longitude and latitude) given in an entry of the RoD service dataset, we count the number of POIs of each of these 14 categories within a 500-meter radius of the location, and use the resulting vector as our POI data. The volume of the POI dataset is the same as that of the RoD service dataset. In Fig. 11, we show, for all locations, the distribution of POI categories that have the largest count around each location – the *shopping*, *business* and *everyday life* POIs are the most prevailing. Fig. 12 focuses on the total number of POIs around each location, and the histogram shows while in most cases there are less than 20 POIs around a location, in much rarer cases there are up to 300 POIs within a 500-meter radius of some locations.

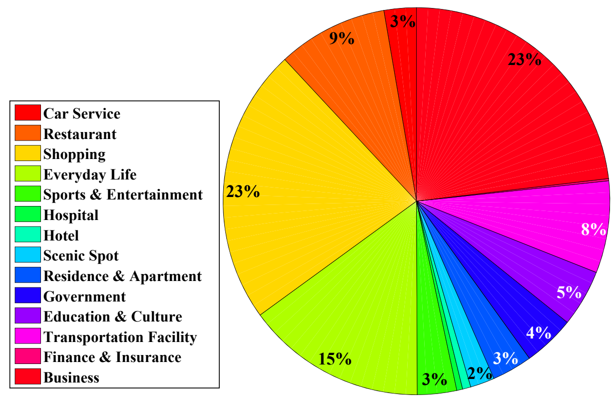

Figure 11: The distribution of POI categories.

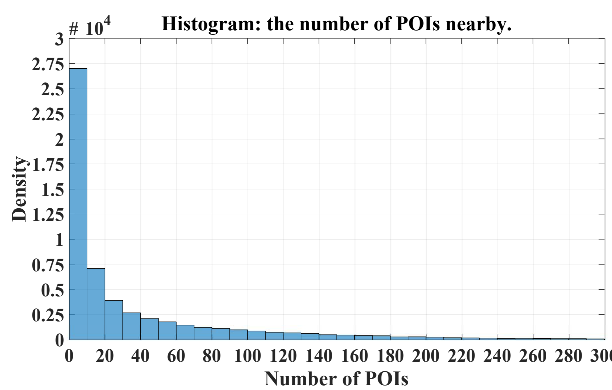

Figure 12: The histogram: the total number of POIs around.

Some previous work may associate a location with its nearest POI and use its category to describe the location. We consider this way to be not accurate enough. For example, a passenger is standing out of a big shopping mall and there are also some restaurants or lifestyle services around him. It is possible that a particular restaurant is the nearest POI, but the big shopping mall turns out to be the reason why the passenger is standing here requesting for the RoD service.

2.5 Weather Data

Weather should also be a factor that influences either the dynamic prices or the number of ***EstimateFee*** events. Intuitively, a higher demand is triggered when there is a bad weather, such as rain or extreme temperature.

We turn to *timeanddate.com* for the weather data. We crawl the weather data in every 3 hours in the complete 4 months from August to November, 2016 in Beijing, corresponding to the time range of the RoD service data. The weather data includes the following fields: *temperature*, *wind speed*, *humidity*, *pressure*, *visibility* and *weather condition*. The first five fields are self-explanatory, and *weather condition* categorizes the weather into 17 types: *ice fog*, *partly sunny*, *sprinkles*, *scattered clouds*, *heavy rain*, *dense fog*, *sunny*, *clear*, *overcast*, *light rain*, *low clouds*, *haze*, *fog*, *rain*, *passing clouds*, *light fog* and *light snow*.

It is true that the weather dataset has a coarser granularity than other four datasets. Firstly, we only have the weather of the whole city of Beijing, instead of having the weather information associated to each smaller region. Secondly, the weather information is updated every 3 hours. This is due to the availability of the weather history data, but we consider our crawled data is enough to make some sense: compared to other factors, the weather condition affects a far larger area and its effect usually lasts much longer.

3 FEATURE EXTRACTION

We describe how to extract features from each of our multi-source urban datasets. Features of the RoD service and taxi dataset are processed from their corresponding data fields, whereas features of other datasets are simply the corresponding data fields, as summarized in Tab. 2.

3.1 RoD Service Features

We extract the following features from RoD service dataset: *month*, *hour of day*, *day of week*, *day of month*, *estimated fare*, *isHoliday*, *isWeekend*, and *historical price multipliers*. The *historical price multipliers* include the average price multiplier in the last 1, 2 and 3 hours within a 500-meter radius of the location of the corresponding RoD data entry. Because the price multiplier is regular, to different extents, in different locations as shown in Fig. 2, we expect the historical price multipliers will influence the current price multiplier. The *estimated fare* is an indication of the travelling distance, and when averaged, the average estimated fare in a small region describes the travelling habit of people in this region from a particular perspective. Other features are temporal features, and as we can see from Fig. 2 to 8, these features play an important role in predicting the price multiplier. The location of the ***EstimateFee*** event (i.e., a pair of longitude and latitude values) is not used as a RoD service feature; alternatively, we use the location of the event to extract features from the other 4 datasets. In the following, "around the location" means "within a 500-meter radius of the location".

3.2 Taxi Service Features

In §2.2, we mention that we have the taxi GPS trajectory data and also extract the trip information of taxis. From the trip information, we extract 2 features: *up count* and *down count* – the number of passengers getting on (off) taxis around the location (i.e., the pair of longitude and latitude values in a RoD data entry). From the taxi GPS trajectory, we extract 5 features:

- *aveage speed*: the average speed of full taxis (i.e., with passengers on-board) around the location.
- *speed variance*: the variance of speed among full taxis around the location.
- *taxi count*: the number of taxis appearing around the location.
- *full taxi count*: the number of full taxis appearing around the location.
- *full taxi ratio*: the ratio of full taxis to all taxis around the location.

The *up/down count* features are an indication of passengers' demand for taxis and the popularity of the location. The *average speed* and *speed variance* reflect the traffic condition around the location. The other 3 features describe the availability of taxis as well as the popularity of the location.

For these 7 features, we calculate each of them based on the taxi GPS entries that fall in the same hour-of-day (called "*hourly taxi features*") and in the same hour-of-day and day-of-week (called "*daily taxi features*"). Additionally, we extract 2 other daily taxi features: *variance of taxi count* and *variance of full taxi count* to characterize the variance of the availability of taxis and of the location's popularity.

The above taxi service features can not only reveal information about the taxi service, but also provide clues to a number of useful facts about the location.

4 MODEL AND EVALUATION

4.1 The Neural Network Model

In this subsection, we present our neural network model to predict dynamic prices in RoD service. The prediction target is the dynamic price multiplier for any passenger request in any location in the city of Beijing. The input features are those explained in §3 and Tab. 2, having a dimension of about 130. In other words, given the following information:

- the temporal features and historical price multipliers around the location (i.e., the location one requests for a ride);
- the taxi features around the location;
- the distribution of buses and metro around the location;
- the POIs around the location;
- the weather condition around the location,

the neural network model tries to predict the price multiplier one will encounter in his/her request.

Table 2: Feature extraction from different datasets.

Dataset	Feature	Description
RoD	*month*	the month the RoD event takes place
	hour of day	the hour of day the event takes place
	day of week	the day of week the event takes place
	day of month	the day of month the event takes place
	estimated fare	the estimated trip fare for the event
	isHoliday	whether the event takes place in a holiday
	isWeekend	whether the event takes place in weekends
	historical price multipliers	the average price multiplier in the last 1, 2, 3 hours
Taxi	*up count*	# of passengers getting on taxis around the location
	down count	# of passengers getting on taxis around the location
	average speed	average speed of full taxis around the location
	speed variance	variance of speed among full taxis around the location
	taxi count	# of taxis appearing around the location
	full taxi count	# of full taxis appearing around the location
	full taxi ratio	the ratio of full taxis to all taxis around the location
	variance of taxi count	variance of taxi count daily
	variance of full taxi count	variance of full taxi count daily
Bus & metro	*bus station count*	# of bus stations around the location
	bus line count	# of bus lines around the location
	metro station count	# of metro stations around the location
	metro line count	# of metro lines around the location
POI	*POI counts*	# of POIs of 14 categories around the location
Weather	*temperature*	the temperature of the city at the time of the event
	wind speed	the wind speed at the time of the event
	humidity	the humidity at the time of the event
	pressure	the atmosphere pressure at the time of the event
	visibility	the visibility at the time of the event
	weather condition	the type of weather at the time of the event

In general, for predictive models, there are generally two paradigms: (a) complicated non-linear models with a small number of features [10, 18] and (b) simple linear models with a large number of features [17, 25]. For linear models, the ability to explain the results regarding feature contribution is a plus, but the loss of the inherent non-linearity in the model prevents us from generating highly accurate prediction without using high-dimensional composite features to express the correlation between features. Non-linear models, on the contrary, make it harder to explain the results quantitatively, but facilitate the feature engineering process as they only require using a smaller number of features. For our prediction task, we aim to achieve an accurate prediction, and explaining feature contribution is not as important as an accurate prediction result. In our study, we choose the neural network model to perform dynamic price prediction.

It is also possible to offer certain level of explanation with our neural network model. With multi-source urban datasets, we choose to identify the importance of each dataset – a particular dataset can improve the prediction accuracy more than another. With a neural network model, we train it on different combinations of datasets, and identify the importance based on the corresponding accuracy measure.

Our neural network model uses a four-layer structure. There are three hidden layers with ReLU activation function between the input and output layer. The data fed to the input layer is a tuple of about 130 dimensions, and the output is a continuous value between 1.0 and 1.6 (i.e., the lower and upper bound of price multipliers in our data).

4.2 Evaluation Setup

4.2.1 Evaluation Metric. The usual way to evaluate the performance of a prediction algorithm is based on the "absolute" accuracy measure, i.e., how many of the predicted items, p_i, are equal to the ground-truth y_i. In predicting price multipliers, on the other hand, we don't care that much about the "absolute" accuracy. In some cases, even though there are a slight difference between the predicted multiplier and the ground truth, it is not a problem for passengers. For example, a passenger getting a price multiplier 1.3 only wants to know if it would be possible to get a lower multiplier nearby or within a short distance, but doesn't care that much whether the lower multiplier is 1.1 or 1.2.

Instead, we use the symmetric mean absolute percentage error (sMAPE) [29], a metric based on the relative error:

$$sMAPE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \frac{|y_i - p_i|}{y_i + p_i}. \qquad (1)$$

In (1), N_{test} is the size of the testing set. A higher sMAPE means lower prediction accuracy. There are multiple considerations on choosing sMAPE as our evaluation metric:

- The sMAPE metric has the advantage of being scale-independent and easily interpretable, as it is in a percentage form. Furthermore, as the price multiplier is always greater than 0, we can avoid getting undefined values in using sMAPE (i.e., the sMAPE metric may take an undefined value when the predicted or the actual quantity is 0)
- More importantly, because of the special properties of price multipliers, other metrics such as MAE, MSE or RMSE can be directly represented from the prediction accuracy (i.e., in what percentage we have a absolute difference being 0, 0.1, 0.2,...,0.6 between the predicted price multiplier and the ground truth), while sMAPE cannot. This is because the price multiplier only takes discrete values such as 1.0, 1.1, ..., 1.6 (and we round the predicted price multiplier to these values), and so the difference between the predicted price multiplier and the ground truth also takes discrete values. As a result, metrics such as MAE/MSE/RMSE can be calculated from the prediction accuracy directly.
- Using sMAPE to measure the relative prediction error is a common practice in evaluating forecast accuracy on, for example, human mobility pattern, taxi demand prediction and so on [27, 28, 30, 31]. Moreover, the baseline predictors we use in our evaluation (see §4.2.2) also uses the sMAPE metric. To compare our results in this paper with the baseline predictors, we also use sMAPE so that it can give a sense as to how our prediction model performs.

4.2.2 Baseline. We use a simple predictor in our previous work [11] as the baseline. It tries to predict the hourly average price multiplier in specific city functional areas such as business, residential and transportation areas, based on the history of hourly average price multipliers in a one-month time range. This baseline is a Markov-chain predictor, predicting the price multiplier by training a 3-order Markov predictor based on the current and past price multipliers. It is applied in a coarser granularity: first, it only tries to predict multipliers in specific functional areas; second, the goal of prediction is the hourly average price multiplier. By comparison, the neural network model in our paper is used to predict the *exact* price multiplier given a set of features in each *individual* passenger request, in *any location* of the city. But the baseline predictor can still give us a sense of the sMAPE metric. Tab. 3 shows the sMAPE of the baseline predictor in selected business, residential and transportation areas. Similar to Fig. 2, the criteria of selecting these typical functional areas could be found in our previous work [11, 12] and is not discussed here.

Table 3: sMAPE of the baseline predictor in different functional areas.

Business	Residential	Transportation
0.0548	0.0468	0.0366

4.3 Evaluation Results

We randomly choose 70% of our 14,587,353 entries as the training set, and the remaining 30% as the test set. The neural network model is trained based on the training set, and we perform this process for 10 times. The resulting average sMAPE is 0.0385. This sMAPE could be considered a much better result than our baseline predictor for the following two reasons:

- The sMAPE of our neural network model is already lower than the baseline predictor, except in the case of transportation area. The specific city functional areas chosen in our previous work already exhibit, to some extent, certain regularity in either the demand or the price multiplier, but in our paper the neural network model is for any location in the city, where the price multipliers are more random.
- The sMAPE of our neural network model is averaged among every single ***EstimateFee*** event, instead of from the predicted hourly average price multiplier.

In addition to the sMAPE, we also calculate the absolute difference between the predicted price multiplier and the ground truth based on the test set. The percentage of predicting exactly the ground truth multiplier is 53.21%, and the percentage of having a difference of 0.1 is 32.47%. The percentage of having a different of 0.2 to 0.6 are 10.12%, 2.98%, 0.73%, 0.42% and 0.07%. In other words, our model can have a very good prediction (i.e., having a different smaller than or equal to 0.1) in about 85.68% cases.

We then investigate the importance of each dataset. To do this, we train a neural network model (with the same number of hidden layers and hidden neurons) for each combination of datasets, and list the average sMAPE of some representative combinations in Tab. 4. In Tab. 4, we also copy the sMAPE of using all datasets in the first row. We have the following observations:

- Using multi-source urban data indeed improves the prediction accuracy significantly. Particularly, using all 5 datasets decreases the sMAPE by about 15.01%, compared with only using the RoD dataset.
- Features extracted from the taxi and weather dataset are more important than other features, as using these two datasets can improve the sMAPE more significantly. This indicates that the availability of taxis and weather condition have a stronger influence on the dynamic prices in RoD service.

Table 4: sMAPE of using different combinations of datasets.

Datasets	sMAPE
RoD+Taxi+Bus&Metro+POI+Weather	0.0385
RoD+Taxi+Weather	0.0399
RoD+Taxi	0.0423
RoD+Bus&Metro	0.0441
RoD+POI	0.0435
RoD+Weather	0.0421
RoD	0.0453

- Public transportation has a smaller influence on the dynamic prices in RoD service. This can be verified by the fact that using datasets from RoD and Bus & Metro services does not bring a big enough improvement on sMAPE.

5 RELATED WORK

Ride-on-demand Service. Most studies on emerging RoD services are centered on dynamic pricing. [7] tries to evaluate Uber's surge pricing mechanism based on the measurement treating Uber as a black-box, and predicts future prices based on a guessed relationship between price multiplier and supply & demand. The prediction is not accurate enough, due to the lack of real service data and the inaccuracies in guessing the relationship. Our previous work [12–15] study and analyze the demand, the effect of dynamic pricing and passengers' reaction to prices in RoD services. In [11] we present a preliminary study on predicting price multipliers, but at a coarser granularity spatio-temporally. Specifically, [11] only uses the RoD service data and weather data to predict the *hourly average price multiplier* in city cells or specific city functional areas. The authors define a metric to characterize the variation pattern and the predictability of price multipliers in different regions in the city, and use different predictors such as Markov-chain predictor or neural network predictor in different regions based on the defined metric. It is a reflection of the varying price multipliers in different time and locations, and can tell passengers "when and where you may get a lower hourly average price multiplier". Other works focus on economic analysis of the effect and impact of dynamic pricing [16], the supply elasticity [8], consumer surplus [9], etc.

Taxi and Other Transportation Services. Our work on price multiplier prediction is inspired by previous work on taxi demand prediction. The availability of public taxi dataset leads to a number of related studies. For example, [5] infers the trip purposes by taking taxis in a real time manner; [6] proposes to hitchhike citywide pakages by leveraging the under-used taxi capacity. [22] uses neural network to forecast the taxi demand from historical data; and [21] uses SVM to determine the most related feature of taxi demand.

Concerns on Dynamic Pricing. Dynamic pricing is not an invention in RoD service, and it has been used in lots of services and scenarios to either improve service efficiency or manipulate supply and demand in different forms. For examples, it has been used in Internet retail [4], inventory management [3], hotel booking [20] and airline pricing [24]. For the RoD service, the mental burden created by dynamic prices have been discussed previously. [15] shows that during morning rush hours, the probability of finding a lower price multiplier within 1km is about 75.99%. The probability is 76.10% and 34.21% for evening rush hours and non-rush hours. [14] shows that only in 39.77% cases a passenger accepts the price multiplier after only one fare estimation. Concerns about the relationship between RoD service and taxi service or public transportation could also be found in news reports such as [1, 19, 23].

6 CONCLUSION

We focus on the prediction of dynamic prices in emerging RoD services such as Uber and Didi. Predicting the dynamic prices can help passengers to obtain more information and make decisions (i.e., whether to take a ride) at ease. We use a neural network based on features extracted from multi-source urban data to perform the prediction for any passenger request in any location in the city of Beijing.

We train our neural network model based on features of more than 130 dimensions. The model can achieve a very good prediction in about 85.68% cases, with a sMAPE of 0.0385, much better than our baseline predictor. The sMAPE metric also provide some clues regarding the feature contribution and the relationship between different transportation services. Results show that the availability of taxis and the weather condition have the strongest influence on the RoD service, whereas the distribution of public transportation services like bus and metro has a more negligible impact.

ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Project of China (No. 2017YFB1002000), the National Natural Science Foundation of China (No. 61602067, 61872050, 61572059, 61202426), the Fundamental Research Funds for the Central Universities (No. 2018cdqyjsj0024), the Chongqing Basic and Frontier Research Program (No. cstc2018jcyjAX0551), the Science and Technology Project of Beijing (No. Z181100003518001), and the Open Foundation of TUCSU (No. TUCSU-K-17002-01). Chao Chen is the corresponding author.

REFERENCES

[1] Aimee Picchi. 2016. Uber vs. Taxi: Which Is Cheaper? Retrieved Jan 25, 2018 from http://bit.ly/2DMgrMc
[2] AMap. 2017. API of AMap Service. http://bit.ly/2n8YRbZ
[3] Omar Besbes and Assaf Zeevi. 2009. Dynamic Pricing Without Knowing the Demand Function: Risk Bounds and Near-Optimal Algorithms. *Operations Research* 57, 6 (2009), 1407–1420.
[4] Yong Cao, Thomas S. Gruca, and Bruce R. Klemz. 2003. Internet Pricing, Price Satisfaction, and Customer Satisfaction. *International Journal of Electronic Commerce* 8, 2 (2003), 31–50.
[5] Chao Chen, Shuhai Jiao, Shu Zhang, Weichen Liu, Liang Feng, and Yasha Wang. 2018. TripImputor: Real-time Imputing Taxi Trip

Purpose Leveraging Multi-sourced Urban Data. *IEEE Transactions on Intelligent Transportation Systems* 19, 10 (2018), 3292–3304.

[6] Chao Chen, Daqing Zhang, Xiaojuan Ma, Bin Guo, Leye Wang, Yasha Wang, and Edwin Sha. 2017. CrowdDeliver: Planning City-wide Package Delivery Paths Leveraging the Crowds of Taxis. *IEEE Transactions on Intelligent Transportation Systems* 18, 6 (2017), 1478–1496.

[7] Le Chen, Alan Mislove, and Christo Wilson. 2015. Peeking Beneath the Hood of Uber. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference (IMC '15)*. ACM, New York, NY, USA, 495–508.

[8] M. Keith Chen. 2016. Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work on the Uber Platform. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC '16)*. ACM, New York, NY, USA, 455–455.

[9] Peter Cohen, Robert Hahn, Jonathan Hall, Steven Levitt, and Robert Metcalfe. 2016. Using Big Data to Estimate Consumer Surplus: The Case of Uber. Retrieved Jan 25, 2018 from http://bit.ly/2pqXiWo

[10] Jerome H. Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29, 5 (2001), 1189–1232. http://www.jstor.org/stable/2699986

[11] Suiming Guo, Chao Chen, Yaxiao Liu, Ke Xu, and Dah Ming Chiu. 2017. It Can be Cheaper: Using Price Prediction to Obtain Better Prices from Dynamic Pricing in Ride-on-demand Services. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '17)*. ACM, 1–10.

[12] Suiming Guo, Chao Chen, Yaxiao Liu, Ke Xu, and Dah Ming Chiu. 2017. Modelling Passengers' Reaction to Dynamic Prices in Ride-on-demand Services: A Search for the Best Fare. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (2017), 136:1–136:23.

[13] Suiming Guo, Chao Chen, Jingyuan Wang, Yaxiao Liu, Ke Xu, Daqing Zhang, and Dah Ming Chiu. 2018. A Simple but Quantifiable Approach to Dynamic Price Prediction in Ride-on-demand Services Leveraging Multi-source Urban Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3 (2018), 112:1–112:24.

[14] Suiming Guo, Yaxiao Liu, Ke Xu, and Dah Ming Chiu. 2017. Understanding Passenger Reaction to Dynamic Prices in Ride-on-demand Service. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 42–45.

[15] Suiming Guo, Yaxiao Liu, Ke Xu, and Dah Ming Chiu. 2017. Understanding Ride-on-demand Service: Demand and Dynamic Pricing. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 509–514.

[16] Jonathan Hall, Cory Kendrick, and Chris Nosko. 2015. The effects of Uber's surge pricing: a case study. Retrieved Jan 25, 2018 from http://bit.ly/2kayk9O

[17] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD'14)*. ACM, Article 5, 5:1–5:9 pages.

[18] G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507.

[19] Jacob Davidson. 2014. Uber Has Pretty Much Destroyed Regular Taxis in San Francisco. Retrieved Jan 25, 2018 from http://ti.me/1vegHWv

[20] Michael L. Kasavana and A. J. Singh. 2001. Online Auctions. *Journal of Hospitality & Leisure Marketing* 9, 3-4 (2001), 127–140.

[21] Bin Li, Daqing Zhang, Chao Chen, Shijian Li, Guande Qi, and Qiang Yang. 2011. Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2011 IEEE International Conference on*. IEEE, 63–68.

[22] Xiaolong Li, Gang Pan, Zhaohui Wu, et al. 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* 6, 1 (2012), 111–121.

[23] Lloyd Alter. 2017. Is Uber Killing Transit? Retrieved Jan 25, 2018 from http://bit.ly/2DMf1S3

[24] R Preston McAfee and Vera Te Velde. 2006. Dynamic pricing in the airline industry. *forthcoming in Handbook on Economics and Information Systems, Ed: TJ Hendershott, Elsevier* (2006). http://bit.ly/2ChavdL

[25] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. 2013. Ad Click Prediction: A View from the Trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, 1222–1230.

[26] Shenzhou UCar. 2015. Annual results for the year ended 31 Dec 2015. http://bit.ly/2cFdL6U

[27] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI '16)*. 2618–2624.

[28] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, and Jieping Ye. 2017. The Simpler The Better: A Unified Approach to Predicting Original Taxi Demands on Large-Scale Online Platforms. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, 1653–1662.

[29] Wikipedia. 2017. Symmetric mean absolute percentage error. http://bit.ly/2umuNKT

[30] Kai Zhao, Denis Khryashchev, Juliana Freire, Claudio Silva, and Huy Vo. 2016. Predicting Taxi Demand at High Spatial Resolution: Approaching the Limit of Predictability. In *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data '16)*. 833–842.

[31] Kai Zhao, Sasu Tarkoma, Siyuan Liu, and Huy Vo. 2016. Urban Human Mobility Data Mining: An Overview. In *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data '16)*. 1911–1920.