Mobile Networks and Applications
https://doi.org/10.1007/s11036-019-01308-5

Fine-grained Dynamic Price Prediction in Ride-on-demand Services: Models and Evaluations

Suiming Guo[1] · Chao Chen[2] · Jingyuan Wang[3] · Yaxiao Liu[4] · Ke Xu[4,5] · Dah Ming Chiu[6]

Abstract

Ride-on-demand (RoD) services use dynamic prices to balance the supply and demand to benefit both drivers and passengers, as an effort to improve service efficiency. However, dynamic prices also create concerns for passengers: the "unpredictable" prices sometimes prevent them from making quick decisions at ease. It is thus necessary to give passengers more information to tackle this concern, and predicting dynamic prices is a possible solution. We focus on fine-grained dynamic price prediction – predicting the price for every single passenger request. 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 performed by learning the relationship between dynamic prices and features extracted from multi-source urban data. There are linear or non-linear models as candidates for learning, and using different models leads to varying implications on accuracy, interpretability, model training procedures, etc. We train one linear and one non-linear model as representatives, and evaluate their performance from different perspectives based on real service data. In addition, we interpret feature contribution, at different levels, based on both models and figure out what features or datasets contribute the most to dynamic prices. Finally, based on evaluation results, we provide discussions on model selection under different circumstances, and propose a way to combine the two models. Our hope is that the study not only serves as an accurate prediction for passengers, but also provides concrete guidance on how to choose between models to improve the prediction.

Keywords Dynamic pricing · Urban transportation · Prediction · Ride-on-demand service

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

✉ Suiming Guo
guosuiming@email.jnu.edu.cn

✉ Chao Chen
ivanchao.chen@gmail.com

1 College of Information Science and Technology, Jinan University, Guangzhou, China

2 College of Computer Science, Chongqing University, Chongqing, China

3 Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Computer Science and Engineering, Beihang University, Beijing, China

4 Department of Computer Science and Technology, Tsinghua University, Beijing, China

5 BNRist, Beijing, China

6 Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

Published online: 26 June 2019

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 helps, and price prediction – predicting the prices in the next time slot or in the neighboring locations – is one of the most straightforward solutions, and passengers could rely on the results to make quick decisions.

Price prediction can be coarse- or fine-grained, depending on who benefits from such results. For example, a coarse-grained prediction – predicting the hourly average price multipliers for certain regions – is already enough for government or policy makers to understand or regulate RoD services. Predictors of this sort have already been discussed in [13, 14]. In this study we focus on fine-grained prediction – predicting the price multipliers for every single passenger request – and it helps passengers to make decisions. Results can be used by either an individual passenger or any third party to answer questions like "*what's the price multiplier for one particular request? or if one can wait for a while, or if one can walk away for hundreds of meters?*".

Dynamic price prediction has not received much attention in RoD services. In general, there are two ways to predict prices. One way is to predict the supply & demand and then guess the relationship between dynamic prices and the supply & demand, as in [8]. 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. 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", but is easier to generalize: the prediction is achievable as long as one can collect historical data, regardless of the service provider-specific algorithms. [13] have discussed coarse-grained prediction using methods in this way, and here we discuss fine-grained prediction in a similar manner.

It depends on the goal of prediction to choose appropriate models in learning the relationship between price multipliers and relevant features. For example, sometimes we focus on the prediction accuracy, and want to have a simpler feature extraction procedure, and in such case non-linear models such as neural network or deep learning models are more suitable. In another case, contrarily, if the emphasis is on interpretability and inspecting feature contribution (i.e., "what features contribute the most to dynamic prices") with a certain level of accuracy, then linear models such as the linear regression model fit better. Additionally, different models may have varying levels of prediction accuracy and applicability in regions or cities with different characteristics. It is thus necessary to evaluate different categories of models and identify how to choose between them or make use of one or more models to improve prediction accuracy.

In this paper, we address the fine-grained dynamic price prediction and its model selection problem by training representative linear and non-linear models on features extracted from multi-source urban data. The rationale behind using multi-source urban data and model selection is:

Multi-source urban data We collect urban data from multiple sources, including the RoD and taxi service, public transportation, weather and the map of a city. This helps to:

- extract more features from data and improves prediction accuracy;
- takes into account the impact on dynamic prices from perspectives other than the RoD service itself, e.g., the status of other means of transportation, weather and location information.

Model selection The main trade-off in model selection is on *accuracy* and *interpretability*. In general, there are two categories of predictive models: (a) complicated non-linear models with a small dimension of features [12, 23] and (b) simple linear models with a large dimension of features [22, 33, 39]. For non-linear models, the non-linearity helps to describe the non-linear correlation between features and they turn out to be more accurate with only a small dimension of features, but in the meanwhile it is hard to determine feature contribution; on the other hand, linear models have a reduced accuracy due to the absence of non-linear terms, but feature contribution can be clearly determined based on corresponding weights. To improve prediction accuracy of linear models, composite features are constructed by multiplying features from different data sources in product-form terms. Specifically, we choose a neural network model and a linear regression model as the representatives of these two categories.

In addition, for these two models, we evaluate their effectiveness in price prediction from different perspectives, and provide a detailed discussion on the level of interpretability of these models. Based on the effectiveness evaluation, we give some concrete suggestions in how to make use of these models to improve prediction accuracy under different circumstances.

Contributions Our contributions are three-fold:

- This paper is in the series of our study on RoD services, and trains two different models for fine-grained dynamic

price prediction. Based on real service data, we conduct extensive evaluation of these two models.
- We introduce multi-source urban data in price prediction. This not only improves prediction accuracy, but enables us to consider the impacts from other perspectives as well, such as the status of other means of transportation, weather and location information. Unlike traditional studies on taxi services, RoD service has a complicated relationship with existing transportation services such as taxi, bus or metro, and this makes it necessary to use multi-source urban data.
- To the best of our knowledge, this study is the first to compare the performance of different predictive models under different circumstances and identify the way to combine them to improve prediction results. Relevant discussions can serve as a useful guidance for passengers, drivers or any other third party.

The remainder of the paper is organized as follows. Section 2 reviews related work, and Section 3 presents the multi-source urban data used in this paper. Section 4 discusses feature extraction, and the two models are presented in Section 5. In Section 6 we carry out extensive evaluations on these models, based on which discussions on model selection are given in Section 7. Finally, Section 8 concludes the paper.

2 Related work

RoD service Most studies on RoD services are centered on dynamic pricing. [8] 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. The authors in [15, 18, 19] study and analyze the demand, the effect of dynamic pricing and passengers' reaction to prices in RoD services. In [13] the authors present a preliminary study on coarse-grained dynamic price prediction. Specifically, the authors in [13] 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. Their work is a reflection of the varying price multipliers in different time period 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 effects of dynamic pricing [20], the supply elasticity [9], consumer surplus [10], etc.

Taxi and other transportation services Our work on price multiplier prediction is inspired by previous work on taxi demand prediction. Li et al. [27] uses neural network to forecast the taxi demand from historical data; [26] uses SVM to determine the most related feature of taxi demand; [7] uses taxi GPS trajectories to detect anomalous trips; etc. The availability of public taxi dataset leads to a number of related studies. For example, [52] uses taxi trajectory data to detect flawed urban planning, and [48] recommends driving directions based on patterns mined from historical taxi trajectory data. Besides, data from taxi and other transportation services have also been used in traffic speed prediction [42], event detection [3], city structure discovery [41], human mobility [6, 34–36], safe driving [47], crowd management [11, 49], taxi ride-sharing [21, 30], trajectory clustering [31], mobile crowd sensing [45], route planning [28] and other urban computing topics [40, 43, 44].

Dynamic pricing and concerns 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 [5], inventory management [4], hotel booking [25] and airline pricing [32]. For the RoD service, the mental burden created by dynamic prices have been discussed previously. Guo et al. [19] 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. Guo et al. [18] 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, 24, 29].

3 Multi-source urban data

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. Table 1 summarizes our datasets and their fields.

3.1 RoD service event-log data

Our data of the RoD service is collected from Shenzhou UCar, one of the major RoD service providers in China. By the end of 2015, Shenzhou UCar's service covers more than

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.

50 cities in China, with a fleet of more than 30,000 cars, offering more than 300,000 trips per day [37].

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.

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.

Figure 2 illustrates how different price multipliers could be in different locations or during different time periods by showing 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. The criteria of selecting these typical functional areas could be found in [15] and is not discussed here.

There are some basic observations:

- 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. We don't

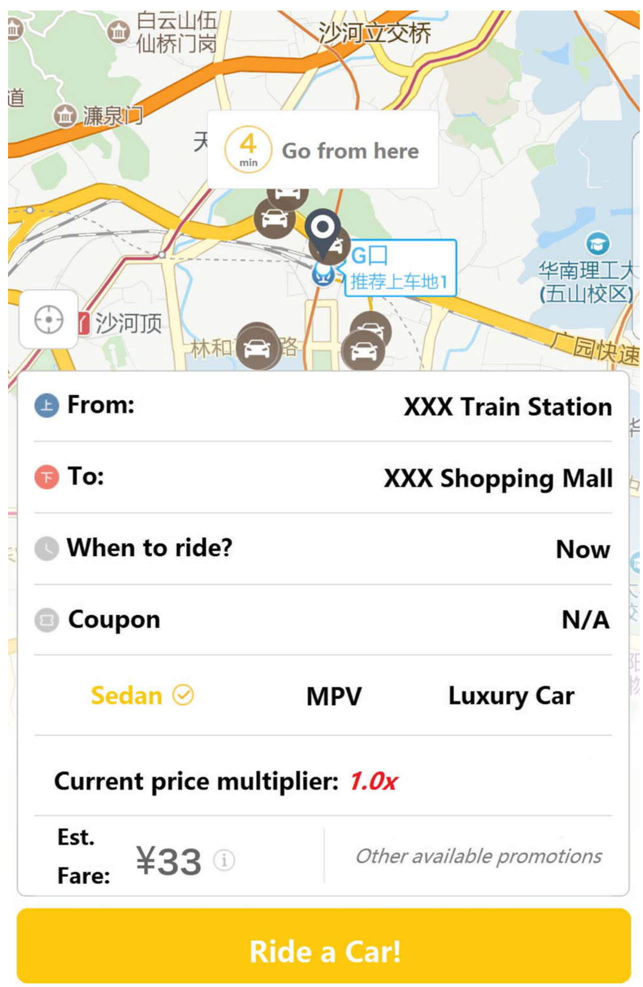

Fig. 1 The user interface of a typical RoD service

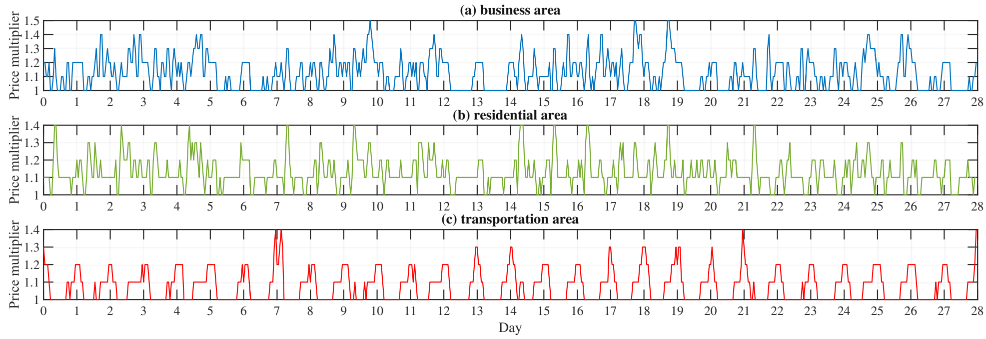

Fig. 2 The variation of hourly average price multipliers in different functional areas

illustrate this observation here because of limited space, and relevant figures can be found in [16].

3.2 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.

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.

3.3 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.

3.4 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 mentioned in Section 3.1, either the price multiplier or the number of events is closely related to the location in which 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. 3, 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. Figure 4 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.

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.

3.5 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*.

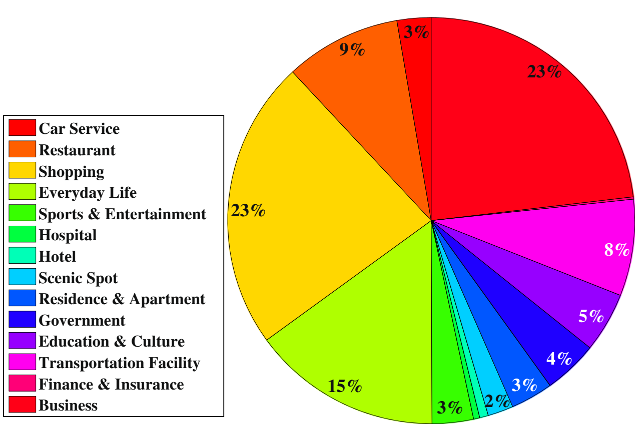

Fig. 3 The distribution of POI categories

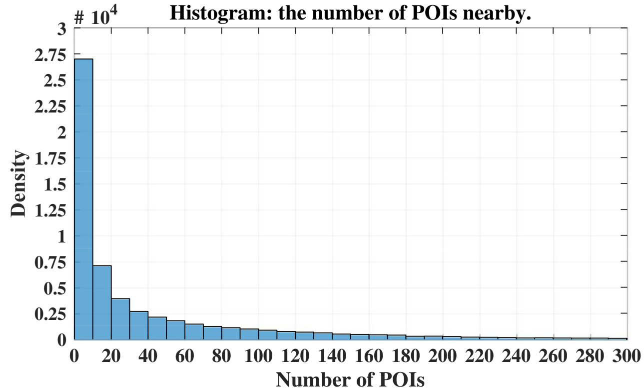

Fig. 4 The histogram: the total number of POIs around

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.

4 Feature extraction

Based on the multi-source urban data, we extract and construct features and these features will later be used to train linear and non-linear models to perform fine-grained price prediction. For a non-linear model, features are directly extracted from our datasets, and we call them as *basic features*; for a linear model, due to the lack of non-linear terms, we need to integrate basic features extracted from different data sources to form new, high-dimensional *composite features*, acting as the substitute for non-linear terms. In this section, we elaborate on how to extract and construct basic features. Using composite features with linear model has been discussed in [17], and relevant key points will be presented later in Section 5.2. Table 2 summarizes all basic features.

Basic features are extracted from each of our datasets. Features of the RoD service and taxi dataset are processed and calculated from their corresponding data fields, whereas features of other datasets are simply the corresponding data fields previously shown in Table 1.

Table 2 Feature extraction: basic features from multi-source urban data

Basic features		
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

4.1 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 Section 3.1, 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".

4.2 Taxi service features

In Section 3.2, 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 (and off) taxis around the location (i.e., specified by 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 the above 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 features can not only reveal information about the taxi service, but also provide clues to a number of useful facts such as the traffic condition around the location.

5 Models for prediction

We have already discussed the multi-source urban data as well as the construction of basic features based on such datasets. In this section we present two representative models for dynamic price prediction – a linear regression model and a (non-linear) neural network model.

The rationale behind choosing different models is about the trade-off between prediction accuracy and interpretability. In some cases, we desire model interpretability in addition to prediction accuracy. For example, when we want to quantify feature contribution and understand what factors contribute the most to the dynamic prices, we need an interpretable model, and in this case interpretability is emphasized over prediction accuracy. Specifically, non-linear models such as neural network or deep learning models are hardly interpretable, and they are always viewed as black-boxes; linear models such as the linear regression model or simple decision tree model, on the other hand, are interpretable by nature, and it is very easy to do that by simply inspecting weights or following decision tree directions.

Our choice of representative models (i.e., the neural network model for non-linear models, and the linear regression model for linear models) is based on the consideration that they are already accurate enough in price prediction, and that they are enough to illustrate the trade-off between linear and non-linear models. For linear models, some complicated linear models such as ensemble models or multiple decision trees have a reduced interpretability, making it harder to judge feature contribution quantitatively. The linear regression model with composite features, on the other hand, is the simplest interpretable model and can achieve a satisfactory accuracy [17]. Secondly, when choosing the representative non-linear model, the goal is to justify that a non-linear model can have a higher prediction accuracy with only limited interpretability. The neural network model is a simple non-linear model, and if it can achieve the above goal, it is then unnecessary to involve those more complicated non-linear models.

In the following we will first present the details of these models, and then a thorough evaluation in the next section.

5.1 The non-linear model

A neural network model is chosen as the representative for non-linear models. The prediction target (i.e., the output of the model) is the dynamic price multiplier for any passenger request in any location in the city of Beijing. The input features are those basic features explained in Section 4 and Table 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, describing both the status of taxi services and traffic condition around;
- the distribution of buses and metro around the location, describing the availability of public transportation services around;
- the POIs around the location, describing the location characteristics around;
- the weather condition around the location,

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

Our neural network model uses a four-layer structure. There are three hidden layers with ReLU activation function between the input and output layer, and each layer contains 15 neurons. 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). Figure 5 illustrates the structure.

Neural network model can also offer certain level of interpretability. With multi-source urban datasets, we choose to identify the importance of each dataset – whether 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 of each dataset based on the corresponding accuracy measure.

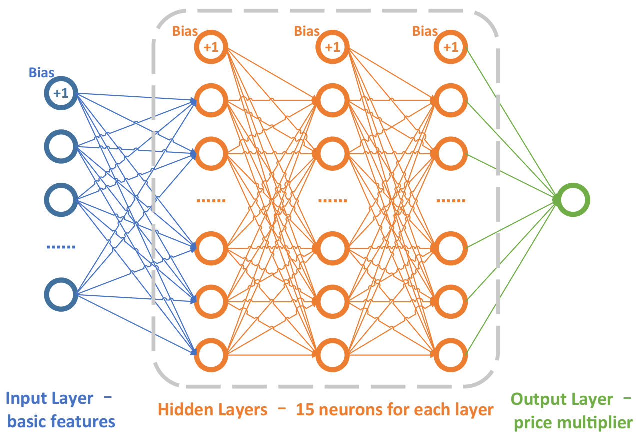

Fig. 5 The structure of the neural network model

5.2 The linear model

The construction of the linear regression model in dynamic price prediction has already been discussed in [17], and in this section we list some key points so that readers can follow the main idea in the following sections about evaluation results and model selection.

Besides *basic features* used in the non-linear model, we introduce *composite features* in the linear model. In a non-linear model, the model itself will automatically discover the relationship between features; but in a linear model, the lack of non-linearity makes it necessary to construct composite features. Without non-linear terms, a linear model is unable to characterize the non-linear relationship between features, and thus has a relatively lower accuracy in fitting the data.

Composite features are the multiplicative product of multiple basic features. Addinvg product-form terms into a linear model transforms the model into a non-linear one, while the model still retains the same level of interpretability. For example, assuming we have two features x_1 and x_2 and the target variable is y, a simple form of a linear regression model can be written as

$$y = \omega_1 x_1 + \omega_2 x_2 + b. \tag{1}$$

If we multiply x_1 and x_2 and use $x_3 = x_1 x_2$ to denote the resulting feature, and then use x_1, x_2 and x_3 to build the linear regression model, the result becomes:

$$y = \omega'_1 x_1 + \omega'_2 x_2 + \omega'_3 x_1 x_2 + b'. \tag{2}$$

In Eqs. 1 and 2, ω_1, ω_2, b, ω'_1, ω'_2, ω'_3 and b' are the model parameters learned. Changing from Eqs. 1-2 with the introduction of x_3 makes the model non-linear, but we can still use $\omega'_i (i = 1, 2, 3)$ and b' to interpret the model. Hence, product-form terms are equivalent to non-linear terms. The construction procedures of composite features can be found in [17], and selected examples of composite features are shown in Table 3. The total dimension of features, including basic and composite ones, reaches about 4,000.

Similar to the non-linear model, given a passenger request (i.e., an *EstimateFee* event at a particular time and location), we extract the *basic* and *composite* features and build a feature vector containing these features. The feature vector is the input to the linear regression model, and the output of the model is a predicted dynamic price multiplier. In training the model, the target to optimize is a squared-error loss of price multiplier, and we add $L1$, $L2$ and a spatio-temporal regularization, as an effort to control sparsity and over-fitting, as well as to maintain the smoothness of price multipliers among neighbouring locations. Finally, we use the stochastic gradient descent (SGD) algorithm to minimize the objective function based on the training data, and obtain a linear regression model to predict dynamic prices.

Table 3 Selected examples of composite features

Type	Datasets	Examples of combinations
Combining basic features from the same dataset	RoD+RoD	(*hour of day*, *day of week*), (*hour of day*, *isWeekend*)...
	Taxi+Taxi	(*full taxi count*, *full taxi ratio*), (*average speed*, *up count*)...
	Bus&metro+Bus&metro	(*bus station count*, *metro station count*)...
	...	...
Combining basic features from different datasets	RoD+Taxi	(*day of week*, *average speed*), (*hour of day*, *up count*)...
	RoD+Bus&metro	(*hour of day*, *bus station count*), (*isHoliday*, *bus line count*)...
	RoD+POI	(*day of week*, *POI counts*), (*isWeekend*, *POI counts*)...
	ROD+Weather	(*day of week*, *weather condition*), (*historical price multipliers ,temperature*), (*day of week*, *visibility*)...
	Taxi+POI	(*full taxi count*, *POI counts*), (*average speed*, *POI counts*)...
	Taxi+Bus&metro	(*taxi count*, *bus station count*), (*up count*, *bus line count*)...

6 Model evaluation

We present the evaluation of both the linear and non-linear models, compare their effects and performance in different situations and provide relevant discussions.

6.1 Evaluation metrics

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 it is 1.1 or 1.2.

Instead, we use the symmetric mean absolute percentage error (sMAPE) [46], a metric based on the relative error:

$$sMAPE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \frac{|y_i - p_i|}{y_i + p_i}. \tag{3}$$

In Eq. 3, 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, with its percentage form. Furthermore, as the price multiplier is always positive, we can avoid getting undefined values in sMAPE.
- 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 [38, 39, 50, 51]. Moreover, the baseline predictors we use in our evaluation (see Section 6.2) also use the sMAPE metric. To compare our results with baselines, we also use sMAPE so that it can give a sense as to how our prediction model performs.

6.2 Baselines

We have discussed and evaluated coarse-grained dynamic price prediction in [13] – predicting the hourly average price multiplier in specific city functional areas or cells – using two predictors, namely a Markov-chain predictor and a neural network predictor. Different from the fine-grained prediction in this paper, these coarse-grained predictors only try to predict multipliers *in specific areas*, and the goal is the *hourly average* price multipliers. By comparison, the two models in our paper are used to predict the *exact* price multiplier given a set of features in any *individual* passenger request, in *any location* of the city. But the baseline predictors can still give us a sense of the sMAPE metric. Table 4 shows the sMAPE of the baseline predictors in selected business, residential and transportation areas, with "M" and "N" denoting the Markov-chain and neural network predictor respectively. Similar to Fig. 2, the criteria of selecting these typical functional areas is not discussed here.

6.3 Experiment results

6.3.1 Non-linear v.s. Linear model

In this section, we compare the accuracy of the linear and non-linear models in dynamic price prediction. We randomly choose 70% of our 14,587,353 entries as the training set, and the remaining 30% as the test set. Each model is trained based on the training set, and we perform the training process for 10 times.

Averaged across the whole city, the average sMAPE for the non-linear (neural network) model is 0.0385, and the sMAPE for the linear regression model is 0.0431. These

Table 4 sMAPE of the baseline coarse-grained predictors in different functional areas

Predictor	Business	Residential	Transportation
M	0.0548	0.0468	0.0366
N	0.0448	0.0457	0.0513

sMAPE values could be considered a much better result than our baseline predictors of coarse-grained prediction, for the following two reasons:

- The sMAPE of the linear and non-linear models in this paper is already lower than the baseline predictors, except in the case of transportation area with the Markov-chain predictor. The reason why in the transportation area our previous baselines perform better is that the demand and price multipliers in this area exhibit certain level of regularity and are much more predictable. By comparison, the models in this paper are for any location in the city, where the price multipliers are more random.
- The sMAPE of our models in this paper 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. For the neural network model, 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%. For the linear regression model, the corresponding figures are 40.28%, 40.89%, 15.32%, 2.25%, 0.86%, 0.38% and 0.02%, respectively. In other words, the neural network and the linear regression model can have a very good prediction (i.e., having a difference less than or equal to 0.1) in about 85.68% and 81.17% cases.

The differences between linear and non-linear models become more apparent when we go into smaller areas instead of calculating the abovementioned metrics across the whole city. [13] point out that the regularity of the variation of dynamic prices, and hence the predictability of price multipliers, differ between different city cells or functional areas. We take city functional areas as examples: transportation areas such as airports or railway stations generally have more regular and highly predictable dynamic prices; business areas such as CBD exhibit just the opposite characteristics and prices are much less predictable; residential areas are in between, showing more stable dynamic prices than business areas, but still with certain level of randomness. We then show the average sMAPE of the linear and non-linear models in these three representative functional areas in Table 5. Different from Table 4, in Table 5 we show the sMAPE averaged across all individual passenger requests taking place in each functional area.

Results from Table 5 gives us the following observations:

- For areas with regular and highly predictable price multipliers (e.g., transportation areas), the linear model has a better (lower) sMAPE than the non-linear model. The linear model with both basic and composite features is already enough to describe the relationship between dynamic prices and relevant features, whereas the non-linear model is more complicated than necessary so that over-fitting may happen from time to time.
- For areas with more random and less predictable price multipliers (e.g., business areas), the non-linear model outperforms the linear model. The relationship in question is more complex than that in areas with highly predictable price multipliers, so a linear model, even coupled with composite features, may be still not enough to characterize the relationship, leading to a worse prediction result.

The differences between these two models go beyond the model accuracy across the whole city or across some specific areas. Below we give some discussions on other differences:

The need for composite features The dimension of feature vector in the neural network model, being only 130, is much smaller than that in our linear regression model, as we do not need to artificially compensate for the lack of non-linear terms. This makes it easy in designing features. In the linear model, we virtually combine any two basic features to form composite features, and perform feature selection based on the corresponding weight in the trained model.

The need for hyper-parameter tuning A non-linear model such as neural network always requires careful tuning to perform well – our neural network model is tuned by trying different sets of hyper-parameters, but it is hard to determine whether our resulting set of hyper-parameters is the optimal one. The need for human experience in parameter tuning makes the model not standardized enough.

The interpretability of results Most importantly, it is easy and natural to interpret the results in a linear regression model – simply inspecting the weight of each feature or

Table 5 The average sMAPE of the linear & non-linear models across different functional areas

Predictor	Business	Residential	Transportation
Linear	0.0452	0.0406	0.0391
Non-linear	0.0373	0.0399	0.0423

feature component is enough. This allows us to judge "what factors contribute to the dynamic prices, and by how much?". A non-linear model, on the other hand, does not offer this level of interpretability with simple inspection.

6.3.2 Effects of using multi-source urban data

We evaluate the effects of using multi-source urban data in improving prediction accuracy by inspecting the sMAPE of using different combination of datasets with the neural network model (with the same model structure). Table 6 lists the average sMAPE across the whole city by using some representative dataset combinations. In Table 6, 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.
- 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.

The effects of using multi-source urban data can be viewed as an interpretation of feature contribution, but at the level of datasets. In other words, we can answer questions like "*what datasets contribute the most to dynamic price prediction?*", "*what is the ranking of datasets according to their contribution?*", etc. The contribution of individual feature is difficult to quantify with the non-linear model, and we will discuss it in more details with the linear model in Section 6.3.3.

Table 6 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

6.3.3 Interpretability in the linear model

Compared with the non-linear model, the linear model is better in terms of interpretability, e.g., feature contribution can be quantified. Feature contribution can be helpful for different parties – passengers, drivers, government agencies, etc. For example:

- Passengers and drivers can learn "*what features are more important in determining dynamic prices?*", "*under what circumstances one may come across higher prices?*", etc.
- Government agencies can learn "*what features are, quantitatively, relevant to dynamic prices?*", "*Is the service provider manipulating prices?*", etc.
- Taxi practitioners can learn "*Is there any competition between taxi service and RoD service, and to what extent?*", "*Is the market large enough for both services to run well?*", etc.

We have already mentioned in Section 6.3.2 that with the non-linear model, it is possible to quantify the importance of each dataset by inspecting the corresponding sMAPE values of training a model using different combination of datasets.

With the linear model, the number of features (basic and composite) is more than 370 and the dimension reaches about 4,000. Each dimension (of features) has a corresponding weight in our trained linear regression model, and the absolute value of the weight quantifies the contribution of this dimension/feature to the predicted dynamic price multiplier. [17] shows the top features that influence dynamic price prediction and presents quantitative interpretation, and here we give some qualitative results to illustrate that a linear model offers better interpretability:

- At the level of dataset, the RoD data, weather data and taxi data are the three datasets most relevant to dynamic price prediction. This can be shown by counting the number of features produced from each dataset among the top features. In fact, among the top-20 features, all are relevant to the RoD data, and 9 or 4 are relevant to the weather and taxi data, respectively.
- At the level of individual feature, the historical price multipliers are the most influential. This reflects the consistency of dynamic prices, and the strong correlation between historical and current price multipliers makes it possible to predict dynamic prices.
- The second most influential features are from weather condition. For example, when there is higher temperature, rain, lower air pressure, the price multiplier rises. In other words, the corresponding weights are positive value – they are large enough, but smaller than the weights corresponding to historical price multipliers.

– The competition from taxis proves to be the third most influential. Fewer taxis or available taxis lead to higher price multipliers in RoD services. Analysis based on feature weights also shows that the competition is only obvious during evening rush hour.

6.3.4 Effects of using composite features

We have mentioned in Section 5.2 and [17] that composite features are constructed in a way to compensate for the lack of non-linear terms in the linear model. At a high level, similar to using multi-source urban data, using composite features lowers the sMAPE of the linear model, and we evaluate these effects by inspecting the sMAPE of using composite features. At a finer granularity, with the linear model we can quantify the contribution of each composite feature, and it has been discussed in Section 6.3.3.

We first train a linear regression model with only basic features (extracted from all datasets), and its sMAPE is 0.0672. Then we train a linear model with basic features and composite features from the same dataset, and the resulting sMAPE is 0.0576. Lastly, a linear model with basic features and composite features from different datasets gives a sMAPE of 0.0557.

Based on these results, we observe that:

– Using composite feature improves prediction accuracy. In fact, using only basic features results a sMAPE 55.92% higher than with composite features.
– Features from different datasets are related to each other in determining the dynamic prices. Without combining features from different datasets, the sMAPE will become 33.64% higher.
– Features within the same datasets also have a significant impact on dynamic prices. Similarly, without combining features from the same datasets, the sMAPE will become 29.23% higher.

6.4 Summary of results

In Section 6.3 we conduct extensive evaluation of and comparison between the linear and non-linear model proposed earlier. Below we summarize some key results:

– Our models achieve better prediction accuracy than baselines, and when averaged across the whole city, the non-linear model is a little bit better than the linear one.
– The two models exhibit different levels of prediction accuracy in locations with different characteristics. The non-linear one performs better in locations with volatile prices, and the linear one gains more power when prices are more stable. Plausible reasons include over-fitting and not-high-enough expressiveness.
– The non-linear model only offers explanation at the level of datasets, whereas in the linear model we can explain the contribution of every single feature quantitatively.
– The use of multi-source urban data and composite features proves to improve prediction accuracy.

7 Discussions on model selection

Based on the multi-source urban data, there are many possible ways to predict dynamic prices at a fine granularity, and in this paper we propose two representative linear and non-linear models. Previous sections have already explained the structure of each model, evaluate their prediction accuracy, and compare their performance in locations with different characteristics, with a brief summary in Section 6.4.

In this section, we give some discussions and suggestions about choosing between these models – *how to make full use of these models to meet specific targets*.

Goal of model: learning v.s. predicting As we have mentioned, different groups of people have various requirement on the price prediction model. Government agencies, consultants and taxi practitioners want to learn, quantitatively and qualitatively, the relationship between dynamic prices and relevant features to ease their concerns about price manipulation, new service introduction and regulation, competition, etc. Individual drivers or passengers, on the other hand, mostly need only an accurate enough prediction result. Hence, if the goal of prediction is *learning*, the linear model should be selected, and in the meantime the use of multi-source urban data and composite features guarantees a satisfactory level of prediction accuracy. If the goal is *predicting*, then other criteria will be discussed in the following.

Concerns about model training We give some examples of concerns about model training.

– *Training time* is an important measure frequently considered in model training. In our case, though both the linear regression and neural network models support batch-based training (e.g., stochastic gradient descent), the training time for a single batch is more than tripled with the neural network model than with the linear regression model. So if the training time requirement cannot be relaxed, a linear model may be a better fit. For example, when the service policy and the price prediction models or parameters are constantly updated (e.g., every several minutes or half an hour, etc.) due to rapidly-changing traffic condition, using different policies for different locations and time, etc., then there is a tight restriction on model training time.

– *Training parameters* is another problem. With a non-linear model such as neural network, when there are newly designed features or when we change the extraction technique of some features, all the hyper-parameters (e.g., number of layers, number of neurons, learning rate, etc.) need to be carefully re-tuned. Things get worse with higher complexity of the non-linear model – in a deep learning model, for example, extra convolution neurons have to be designed. In a linear model, on the other hand, such procedures are simplified to extracting new basic features, constructing new composite features by multiplication, and using SGD to train a new model – requiring little or no parameter tuning based on human experience. So if the service provider is faced with significant and frequent changes in service model or government regulation, a linear model may be a better fit.

Number of basic features combined In the linear model, the description of the non-linear correlation between basic features is represented by combining basic features into composite features. Hence, the lack of expressiveness for the linear model in areas with more unpredictable prices is, to some extent, due to the fact that in our study a composite feature is only combined from two basic features.

In our study, we try combining more than two basic features (e.g., three), and it indeed reduces the sMAPE by up to 15%. But this at the same time increases the dimension of features to a value more than tripled, leading to a notably longer training time. As a result, under such circumstances the non-linear model should be selected.

Combining the two models Our evaluation results show that these two models perform differently in locations with different characteristics. In fact, the performance of our models may be specific, to some extent, to the service provider (e.g., different service providers may have various targets to optimize in determining their dynamic prices), the choice of city (e.g., Beijing has a totally different human mobility patterns compared to some smaller cities), etc. Hence, the applicability of these models may vary in different cities or to different service providers. However, the models themselves and relevant methodologies are generic – as long as one can collect the required data, s/he can build these models to predict dynamic prices and evaluate their performance.

To improve applicability, we propose a weighted combination of these two models – the predicted price multiplier is the weighted sum of the predicted price multipliers of the two models. We denote the single feature vector containing only basic features as $\mathbf{x_b}$, and the feature vector containing both basic and composite features as $\mathbf{x_c}$. For the predicted price multipliers, we use p, p_{lin} and p_{non} to denote the predicted result of the weighted combination model, linear model and non-linear model. Also, f_{lin} and f_{non} represent the linear and non-linear models. Then we have,

$$p = \alpha p_{lin} + (1-\alpha) p_{non} = \alpha f_{lin}(\mathbf{x_c}) + (1-\alpha) f_{non}(\mathbf{x_b}) \quad (4)$$

In Eq. 4, α ($0 \leq \alpha \leq 1$) is the weight between the two models. The weighted model always produces a weighted error of the two models and, if the two models have errors in different signs, the weighted model may produce an error smaller than either of the two models. We regard α as a representation of city characteristics, and it is also acceptable to calculate the values of α at finer granularities – for example, one may want to calculate an α for each city functional area or for each day-of-week, as an effort to distinguish characteristics of dynamic prices across city functional areas or days-of-week. It should be noted that a too small granularity is not realistic either, as this may bring a prohibitively high computational complexity, and lead to over-fitting, such that α may fluctuate too much and such fluctuation may not represent any spatio-temporal characteristics. Here we only use a single α for the whole city across the time range of our dataset for illustrative purposes, and we currently do not have large enough datasets to go into a granularity such as city cells.

Determining the optimal α requires a trial-and-error process. Basically, the optimal α depends on city characteristics, such as the distribution of functional areas, the regularity of human mobility patterns, the service provider's target and policy in the city, etc. For example, a metropolitan city such as Beijing has human mobility patterns and functional area distribution much more complex than a smaller city or a tourism city. This may, to some extent, require a larger weight for the non-linear model (i.e., a smaller α). However, such relationship between α and city characteristics may not have a closed-form, and hence trail-and-error is needed.

In our case, we find out that $\alpha = 0.23$ gives the lowest sMAPE 0.0372 (averaged across the whole city). In our future work, we will try to collect multi-source urban data in cities with different characteristics, and find the corresponding αs for these cities, as an effort to quantify the relationship between α and city characteristics.

The weighted combination of the two models is generic and improves model applicability – as long as one have the required data, s/he can train models and find out the optimal α to combine them. This procedure is not specific to service provider or the choice of city. Furthermore, the weighted combination model helps to improve prediction accuracy, and is suitable when prediction accuracy is emphasized over the running time of the model.

8 Conclusion

We focus on the fine-grained dynamic price prediction problem in 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 and linear regression model, as representatives for non-linear and linear models, based on features extracted from multi-source urban data to perform prediction. For the linear regression model, to boost model expressiveness, composite features are constructed by combining basic features in a product form.

We conduct extensive evaluation of the two models, including the comparison between the linear and non-linear model, the effects of using multi-source urban data and composite features, and the interpretability in the linear model. Results show that, when averaged across the whole city, the non-linear model performs a little bit better than the linear one, but they behave differently in locations with volatile or stable price variation – in general the non-linear model is better with volatile prices, and the linear one becomes better with stable prices. Based on these results, discussions regarding how to make full use of these models for prediction accuracy are presented, and we also propose using a weighted combination of the two models to improve prediction accuracy as well as model applicability.

Acknowledgements The work was supported by the National Natural Science Foundation of China (No. 61872050, 61602067, 61572059, 61825204), the National Key Research and Development Program of China (No. 2017YFC0820405), the Fundamental Research Funds for the Central Universities (No. 21619310, 2018cdqyjsj0024), the Chongqing Basic and Frontier Research Program (No. cstc2018jcyjAX0551), the Science and Technology Project of Beijing (No. Z181100003518001), the Open Foundation of TUCSU (No. TUCSU-K-17002-01), and Beijing Outstanding Young Scientist Project. Chao Chen is the corresponding author.

References

1. Aimee Picchi (2016) Uber vs. Taxi: Which Is Cheaper?. http://bit.ly/2DMgrMc
2. AMap (2017) API of AMap Service. http://bit.ly/2n8YRbZ
3. Aoki S, Sezaki K, Yuan NJ, Xie X (2017) An early event detection technique with bus gps data. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL '17. ACM, pp 49:1–49:4
4. Besbes O, Zeevi A (2009) Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. Oper Res 57(6):1407–1420
5. Cao Y, Gruca TS, Klemz BR (2003) Internet pricing, price satisfaction, and customer satisfaction. Int J Electron Commer 8(2):31–50
6. Chen C, Jiao S, Zhang S, Liu W, Feng L, Wang Y (2018) Tripimputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data. IEEE Trans Intell Transp Syst 19(10):3292–3304
7. Chen C, Zhang D, Castro P, Li N, Sun L, Li S (2011) Real-time detection of anomalous taxi trajectories from GPS traces. In: 8Th international conference on mobile and ubiquitous systems: computing, Networking and Services (MobiQuitous 2011). Springer, pp 63–74
8. Chen L, Mislove A, Wilson C (2015) Peeking beneath the hood of Uber. In: Proceedings of the 2015 ACM Conference on Internet Measurement Conference, IMC '15. ACM, pp 495–508
9. Chen MK (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, pp 455–455
10. Cohen P, Hahn R, Hall J, Levitt S, Metcalfe R (2016) Using big data to estimate consumer surplus: The case of uber. http://bit.ly/2pqXiWo
11. Franke T, Lukowicz P, Blanke U (2015) Smart crowds in smart cities: real life, city scale deployments of a smartphone based participatory crowd management platform. J Internet Serv Appl 6(1):27
12. Friedman JH (2001) Greedy function approximation: A gradient boosting machine. Ann Stat 29(5):1189–1232. http://www.jstor.org/stable/2699986
13. Guo S, Chen C, Liu Y, Xu K, Chiu DM (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, pp 146–155
14. Guo S, Chen C, Liu Y, Xu K, Chiu DM (2018) How to pay less: a location-specific approach to predict dynamic prices in ride-on-demand services. IET Intell Transp Syst 12(7):610–618
15. Guo S, Chen C, Liu Y, Xu K, Chiu DM (2018) Modelling passengers' reaction to dynamic prices in ride-on-demand services: A search for the best fare. Proc. ACM Interact. Mob Wear Ubiq Technol 1(4):136:1–136:23
16. Guo S, Chen C, Wang J, Liu Y, Xu K, Chiu DM (2018) Dynamic price prediction in ride-on-demand service with multi-source urban data. In: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous '18. ACM, pp 412–421
17. Guo S, Chen C, Wang J, Liu Y, Xu K, Zhang D, Chiu DM (2018) A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data. Proc. ACM Interact. Mob Wear Ubiq Technol 2(3):112:1–112:24
18. Guo S, Liu Y, Xu K, Chiu DM (2017) Understanding passenger reaction to dynamic prices in ride-on-demand service. In: 2017 IEEE international conference on Pervasive computing and communication workshops (percom workshops). IEEE, pp 42–45
19. Guo S, Liu Y, Xu K, Chiu DM (2017) Understanding ride-on-demand service: Demand and dynamic pricing. In: 2017 IEEE international conference on Pervasive computing and communication workshops (percom workshops). IEEE, pp 509–514
20. Hall J, Kendrick C, Nosko C (2015) The effects of Uber's surge pricing: a case study. http://bit.ly/2kayk9O
21. He W, Hwang K, Li D (2014) Intelligent carpool routing for urban ridesharing by mining GPS trajectories. IEEE Trans Intell Transp Syst 15(5):2286–2296

22. He X, Pan J, Jin O, Xu T, Liu B, Xu T, Shi Y, Atallah A, Herbrich R, Bowers S, Candela JQN (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, pp 5:1–5:9
23. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
24. Jacob Davidson (2014) Uber Has Pretty Much Destroyed Regular Taxis in San Francisco. http://ti.me/1vegHWv
25. Kasavana ML, Singh AJ (2001) Online auctions. J Hospital Leisure Market 9(3-4):127–140
26. Li B, Zhang D, Chen C, Li S, Qi G, Yang Q (2011) Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: 2011 IEEE international conference on Pervasive computing and communication workshops (percom workshops). IEEE, pp 63–68
27. Li X, Pan G, Wu Z et al (2012) Prediction of urban human mobility using large-scale taxi traces and its applications. Front Comput Sci 6(1):111–121
28. Liu H, Jin C, Zhou A (2018) Popular route planning with travel cost estimation from trajectories. Front Comput Sci pp(99):1–17
29. Lloyd Alter (2017) Is Uber Killing Transit?. http://bit.ly/2DMf1S3
30. Ma S, Zheng Y, Wolfson O (2013) T-share: a large-scale dynamic taxi ridesharing service. In: 2013 IEEE 29Th international conference on data engineering, ICDE '13, pp 410–421
31. Mao J, Song Q, Jin C, Zhang Z, Zhou A (2018) Online clustering of streaming trajectories. Front Comput Sci 12(2):245–263
32. McAfee RP, Te Velde V (2006) Dynamic pricing in the airline industry. Handbook on Economics and Information Systems. http://bit.ly/2ChavdL
33. McMahan HB, Holt G, Sculley D, Young M, Ebner D, Grady J, Nie L, Phillips T, Davydov E, Golovin D, Chikkerur S, Liu D, Wattenberg M, Hrafnkelsson AM, Boulos T, Kubica J (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, pp 1222–1230
34. Miyazawa S, Song X, Xia T, Shibasaki R, Kaneda H (2018) Integrating gps trajectory and topics from twitter stream for human mobility estimation. Front Comput Sci pp(99):1–11
35. Noulas A, Scellato S, Lambiotte R, Pontil M, Mascolo C (2012) A tale of many cities: Universal patterns in human urban mobility. PLOS One 7(5):1–10
36. Scellato S, Musolesi M, Mascolo C, Latora V, Campbell AT (2011) Nextplace: a spatio-temporal prediction framework for pervasive systems. In: Proceedings of the 9th International Conference on Pervasive Computing, Pervasive '11. Springer, pp 152–169
37. Shenzhou UCar (2015) Annual results for the year ended 31 Dec 2015. http://bit.ly/2cFdL6U
38. Song X, Kanasugi H, Shibasaki R (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, pp 2618–2624
39. Tong Y, Chen Y, Zhou Z, Chen L, Wang J, Yang Q, Ye J (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, pp 1653–1662
40. Wang J, Chen C, Wu J, Xiong Z (2017) No longer sleeping with a bomb: a duet system for protecting urban safety from dangerous goods. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '17). ACM, pp 1673–1681
41. Wang J, Wu J, Wang Z, Gao F, Xiong Z (2019) Understanding urban dynamics via context-aware tensor factorization with neighboring regularization. IEEE Trans Knowl Data Eng pp(99):1–1. https://doi.org/10.1109/TKDE.2019.2915231
42. Wang J, Gu Q, Wu J, Liu G, Xiong Z (2016) Traffic speed prediction and congestion source exploration: a deep learning method. In: Proceedings of the 16th IEEE International Conference on Data Mining (ICDM). IEEE, pp 499–508
43. Wang J, He X, Wang Z, Wu J, Yuan NJ, Xie X, Xiong Z (2018) Cd-cnn: a partially supervised cross-domain deep learning model for urban resident recognition. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI '18). AAAI
44. Wang J, Wang X, Wu J (2018) Inferring metapopulation propagation network for intra-city epidemic control and prevention. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18). ACM, pp 830–838
45. Wang L, Yu Z, Guo B, Yi F, Xiong F (2018) Mobile crowd sensing task optimal allocation: a mobility pattern matching perspective. Front Comput Sci 12(2):231–244
46. Wikipedia (2017) Symmetric mean absolute percentage error. http://bit.ly/2umuNKT
47. You CW, Lane ND, Chen F, Wang R, Chen Z, Bao TJ, Montes-de Oca M, Cheng Y, Lin M, Torresani L, Campbell AT (2013) Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones. In: Proceeding of the 11th annual international conference on mobile systems, applications, and services, mobisys '13. ACM, pp 13–26
48. Yuan J, Zheng Y, Zhang C, Xie W, Xie X, Sun G, Huang Y (2010) T-drive: Driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10. ACM, pp 99–108
49. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI '17. AAAI, pp 1655–1661
50. Zhao K, Khryashchev D, Freire J, Silva C, Vo H (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, pp 833–842
51. Zhao K, Tarkoma S, Liu S, Vo H (2016) Urban human mobility data mining: an overview. In: Proceedings of the 2016 IEEE International Conference on Big Data, Big Data '16, pp 1911–1920
52. Zheng Y, Liu Y, Yuan J, Xie X (2011) Urban computing with taxicabs. In: Proceedings of Ubicomp '11, pp 89–98