ROD-Revenue: Seeking Strategies Analysis and Revenue Prediction in Ride-on-demand Service Using Multi-source Urban Data

Suiming Guo, Chao Chen, Jingyuan Wang, Yaxiao Liu, Ke Xu, *Senior Member, IEEE,* Zhiwen Yu, *Senior Member, IEEE,* Daqing Zhang, *Fellow, IEEE,* and Dah Ming Chiu, *Fellow, IEEE*

Abstract—Recent years witness the rapidly-growing business of ride-on-demand (RoD) services such as Uber, Lyft and Didi. Unlike taxi services, these emerging transportation services use dynamic pricing to manipulate the supply and demand, and to improve service responsiveness and quality. Despite this, on the drivers' side, dynamic pricing creates a new problem: how to seek for passengers in order to earn more under the new pricing scheme. Seeking strategies have been studied extensively in traditional taxi service, but in RoD service such studies are still rare and require the consideration of more factors such as dynamic prices, the status of other transportation services, etc. In this paper, we develop ROD-Revenue, aiming to mine the relationship between driver revenue and factors relevant to seeking strategies, and to predict driver revenue given features extracted from multi-source urban data. We extract basic features from multiple datasets, including RoD service, taxi service, POI information, and the availability of public transportation services, and then construct composite features from basic features in a product-form. The desired relationship is learned from a linear regression model with basic features and high-dimensional composite features. The linear model is chosen for its interpretability – to quantitatively explain the desired relationship. Finally we evaluate our model by predicting drivers' revenue. We hope that ROD-Revenue not only serves as an initial analysis of seeking strategies in RoD service, but also helps increasing drivers' revenue by offering useful guidance.

Index Terms—Ride-on-demand, dynamic pricing, seeking strategy, driver revenue.

1 Introduction

Recent years witness the rapidly-growing business of ride-on-demand (RoD) services such as Uber, Lyft and Didi around the world. RoD service attracts passengers by its convenience, affordable prices, and flexible service; it also attracts drivers by its driving flexibility – drivers do not have to apply for licenses or medallions to enter the service. An increasing amount of passengers now take RoD service as a regular choice in their everyday transportation.

Dynamic pricing is one of the key features making RoD service attractive to both passengers and drivers, as an effort to manipulate the supply (i.e., the number of cars on the road) and demand (i.e., the number of passenger requests). Specifically, a higher price attracts more drivers and defers those requests from passengers who are not in hurry; and a lower price does just the opposite. In most cases, the dynamic prices are represented by a price multiplier, such that the fare of a trip is the product of a *dynamic* price multiplier (dependent on the supply and demand condition) and a *fixed* normal price (based on the trip time and distance).

The adoption of dynamic pricing helps to make the service more responsive and to improve service quality, but it also gives rise to new problems to both drivers and passengers. In this study, we mainly focus on the drivers' side: *how to seek for passengers to earn more*? Instead of relying on some personal, ad-hoc experiences as was in taxi service, in RoD service the price multiplier, a more accurate description of the instantaneous supply and demand condition, becomes a new indicator for drivers to choose seeking strategies. But the effective strategies are still yet to be explored. For example, if all drivers flock to a particular region with high price multiplier, the supply in this region becomes more than enough, causing the price multiplier to drop drastically. This not only generates unstable prices, but also upsets those drivers with an intention of chasing high prices. In fact, many news stories, blogs or research papers have discussed this intuitive "surge chasing" strategy, but unfortunately they sometimes give contradictory suggestions from one to another [1], [2]. As a result, it is pressing that drivers should have some concrete guidance as to how to seek for passengers under dynamic pricing, instead of intuitive or untenable suggestions.

To our knowledge, seeking strategies receive little attention in RoD service. In the traditional taxi service, seeking strategies have been studied from many different perspec-

- *S. Guo is with the College of Information Science and Technology, Jinan University, Guangzhou, 510632, China.*
  E-mail: guosuiming@email.jnu.edu.cn
- *C. Chen is with Chongqing University, Chongqing, China.*
  E-mail: cschaochen@cqu.edu.cn
- *J. Wang is with Beihang University, Beijing, China.*
  E-mail: jywang@buaa.edu.cn
- *Y. Liu and K. Xu are with Tsinghua University, Beijing, China.*
  E-mail: rootliu@gmail.com, xuke@mail.tsinghua.edu.cn
- *Z. Yu is with Northwestern Polytechnical University, Xi'an, China.*
  E-mail: zhiwenyu@nwpu.edu.cn
- *D. Zhang is with Institut Mines-Telecom/Telecom SudPais, Evry Cedex, France.*
  E-mail: daqing.zhang@telecom-sudparis.eu
- *D. M. Chiu is with the Chinese University of Hong Kong, Hong Kong, China.*
  E-mail: dmchiu@ie.cuhk.edu.hk

Manuscript received November 17, 2018; revised April 14, 2019.

tives, e.g., mining patterns of strategies from taxi GPS trajectories, building models such as the Markov Decision Process (MDP) model to evaluate certain strategies, etc. In RoD service, on the other hand, most suggestions are from news stories or blogs that are not rigorous enough, and some few existing researches are mostly based on theoretical models that require a lot of assumptions and approximation. In fact, the lack of real data in RoD service hinders relevant studies based on the data-analytical methodology.

Studying seeking strategies in RoD service requires us to take into account more factors than in taxi service. In our study, we involve factors from two perspectives:

- **Dynamic Prices**: As the core and distinctive feature of RoD service, the dynamic prices should have impacts on seeking strategies. For example, the effects of the intuitive "surge chasing" strategy, and "what to chase" correspondingly, will be discussed. As another example, the adoption of dynamic pricing may change the demand patterns of passengers, so the hours-of-day in which seeking for passengers is the most profitable is also a problem to explore.
- **Status of Other Transportation Services**: There are concerns that the emerging RoD services are competing, to some extent, with traditional transportation services such as taxi, bus or metro. In some cases, these services are also complementary to each other – for example, one may choose to seek in a region with more metro or bus stations to provide connecting services. Hence, the relationship between seeking strategies and the status of other transportation services is also among our targets to study.

In this paper, our goal is to understand the relationship between driver revenue and seeking strategies, i.e., "*what seeking strategies prove to be more profitable*? ". Based on real data, we develop ROD-Revenue, a system that learns an interpretable relationship between drivers' hourly average revenue and seeking strategies from the data, and predicts driver revenue given features relevant to seeking strategies. For the datasets, our study is based on multi-source urban datasets including the data of RoD service, taxi service, points-of-interest (POI), and public transportation services; as to the model used to learn the desired relationship, we resort to a linear regression model with high-dimensional features. The consideration of choosing multi-source urban data and linear regression model is discussed briefly below.

Multi-source Urban Data. We learn the desired relationship from real data instead of theoretical models, and we choose multi-source urban data for multiple reasons. Firstly, the use of multiple datasets helps us to describe the status of different transportation services, before we can learn the impacts of this status on drivers' revenue. Secondly, with more datasets, we can extract more features, making our model accurate enough to learn the desired relationship.

Linear Regression Model with High-dimensional Features. In addition to describing the above relationship and predicting driver revenue, we want to interpret quantitatively the learned relationship, e.g., "*how, and to what extent, one particular feature influences drivers' revenue*?" or "*which feature is the most important? by how much*?". Hence, the model should also be interpretable. Complex non-linear models such as neural network or deep learning models are generally not interpretable, albeit with high accuracy. Some simpler models such as decision tree models are, by its nature, interpretable, but the interpretability is diminished when training multiple trees at high complexity. A linear regression model is one of the simplest models with interpretability – the weight of each feature quantifies its importance – but it is hard for a linear model to characterize clearly the non-linear correlation between features. In our study, we adopt a linear regression model, and compensate for the lack of non-linear terms by adding product-form terms of a combination of features (i.e., composite features). The multiplication of two or more features and using the corresponding result as a new feature in model training help to describe the non-linear correlation between features. To validate the effectiveness of our model, we also implement a neural network model and compare their evaluation results.

Our contributions are three-fold:

- Our study is one of the very few on seeking strategies in RoD service. As far as we know, existing studies mainly use theoretical models such as the MDP model with assumptions about the supply, demand and driver behavior because of the lack of real data. Instead, we are the first to mine the relationship between driver revenue and seeking strategies by a learning model from real service data. Thus, our focus is not only on the learning model itself, but also on mining and understanding new patterns and relationship about seeking for passengers in emerging RoD services and increasing the research community's understanding about such a service.
- To the best of our knowledge, we are the first to involve multi-source urban data in studying seeking strategies in RoD service. This enables us to take into account the status of other transportation services as well as the POI information, instead of considering only RoD service itself.
- Based on the linear model, ROD-Revenue quantifies the above relationship and provide concrete heuristics to drivers as to how to earn more under dynamic pricing in RoD service. Quantifying the relationship helps to understand "*what seeking strategies are more profitable, and by how much?*". The heuristics are derived from real data, and some of them are counter-intuitive and may be contradictory to intuition.

The remainder of the paper is organized as follows. §2 reviews related works. We show the system framework of ROD-Revenue in §3. §4 to §7 elaborate on the three main parts of ROD-Revenue, i.e., multi-source urban datasets, feature extraction, and model & prediction. Together with the model in §7, we also present our evaluation results. §8 provides discussions on feature contribution, seeking strategies and relevant topics. Finally §9 concludes the paper.

2 Related Work

The problem about driver revenue and seeking strategies has been studied in traditional taxi services from different perspectives, but receives very limited attention in emerging

RoD services. We first review some related work in RoD service, then discuss previous studies on seeking strategies.

RoD Services. RoD service is relatively new, and there are fewer studies compared to traditional taxi service. Quite a few compare the differences of the price, waiting time, incentives, and service quality between taxi and RoD service, from a data statistical perspective. For instance, Picchi pointed out that Uber is not always the economical choice although it can reduce the waiting time to a great deal [3]; Salnikov conducted a head to head Uber-taxi comparison study in a reasonable spatio-temporal resolution [4]. In addition, the market sharing of the taxi service and public transportation before and after the entering of Uber is also compared and discussed in [5], [6], [7]. There are also a number of studies estimating Uber's market effects such as "Is Uber a substitute or complement for public transit?" [8], "Drivers of disruptions?" [9], etc.

As a key feature of RoD service, dynamic pricing receives a great deal of attention. [10], [11], [12] examine the effectiveness of dynamic pricing in balancing and redistributing the supply and demand in different regions, increasing driver revenue, reducing passenger waiting time, etc. [13] tries to evaluate Uber's surge pricing mechanism based on the measurement treating Uber as a black-box, but their evaluation is not accurate enough because of the lack of data. [14], [15] study and analyze the demand, the effect of dynamic pricing and passengers' reaction to prices in RoD services. [16] focuses on dynamic price prediction using different data mining techniques. There are also some works on economic analysis of dynamic pricing [10], the supply elasticity [17] and consumer suplus [18].

Seeking Strategies. Seeking strategies, together with seeking route recommendation, have been studied extensively in taxi services. For example, [19], [20] study seeking strategies by mining GPS trajectories, and identify whether hunting (i.e., seeking for passengers actively) or waiting (i.e., staying in popular locations) are more profitable under different circumstances. [21] builds a Markov Decision Process model to optimize taxi driver revenue efficiency. [22] discusses the same problem, but with reinforcement learning. [23] extends the model in [21], incorporates the charging process, takes into account the battery constraint, and discusses how to earn more when driving electric taxis. Alternatively, [24] recommends routes to drivers to minimize the distance between the taxi and an anticipated customer request. However, as taxi adopts fixed pricing, price is not a possible factor that influences seeking strategies. Also, studies on taxi service generally consider the taxi service itself as an entity that influences driver revenue.

In RoD service, there are much fewer studies on seeking strategies considering the effects of dynamic pricing. [25] studies how to optimize earning in on-demand ride-hailing (i.e., another name similar to RoD service) based on theoretical modelling. It models drivers, cities, and the service itself with a number of assumptions and approximations, and the driver strategies mentioned are idealized to a certain extent.

Different from the above works, our study on seeking strategies and driver revenue is based on real data, and tries to mine the relationship between driver revenue and seeking strategies using a learning model. We also evaluate the accuracy of such a model based on ground truth. Besides, we also offer tenable suggestions for drivers to increase revenue based on the learned model.

3 System Framework

In this section, we formalize the problem to study, and then present briefly the system framework of ROD-Revenue.

3.1 Problem Statement

The problem ROD-Revenue tries to solve is to learn the relationship between drivers' *hourly average revenue* and seeking strategies, and then predict any driver's *hourly average revenue* based on the learned model and corresponding seeking strategies.

Definition 3.1 (Timeslot)**.** Our study is on the unit of timeslots. We divide one day into 4 timeslots of equal length: timeslot-0 to -3 refers to [4am, 10am), [10am, 4pm), [4pm, 10pm) and [10pm, 4am), respectively.

Roughly speaking, for weekdays, timeslot-0 and -2 correspond to the morning and evening rush hours; timeslot-1 is the non-rush hours around noon; and timeslot-3 represents night hours. For weekends, our study and [14] suggest that human activity remains relatively high and stable during the day (about [9am, 10pm)), and is lowered during the rest of the day, so the above partition of timeslots still makes sense. In determining the length of timeslots, it cannot be too long to avoid losing useful information of time division; and it cannot be too short to have fewer than enough passenger delivery trips to be representative. Our study is based on real data from Beijing, a major Asian metropolitan city that accommodates a diversified population and hence diversified trip patterns – making the length of rush or non-rush hours longer than normal. In our model and evaluation, we also try to partition one day into timeslots of 4 hours in length, and it proves to generate lower accuracy (see §7.3.2).

Definition 3.2 (Hourly Average Revenue)**.** The hourly average revenue of a driver in one particular timeslot, is defined as the sum of trip fares of all passenger delivery trips taking place in this timeslot divided by the length of the timeslot in hours. We use K to denote the number of trips of a driver in a timeslot, and use $f_k (1 \leq k \leq K)$ to denote the trip fare of the k-th trip, then the hourly average revenue r_{avg} is:

$$r_{avg} = \frac{\sum_{k=1}^{K} f_k}{6}, \tag{1}$$

given that the length of each timeslot is 6 hours.

Choosing the *hourly average revenue* as the target of study is intuitive, as we want to learn the relationship between driver revenue and seeking strategies. Calculating the hourly average revenue over a timeslot of multiple hours helps in dealing outliers or special events.

We use y to denote the hourly average revenue for one driver during a timeslot, and $x \in \mathbb{R}^m$ the feature vector, with m being the dimension of the feature vector. From our datasets, we can extract N data entries of different drivers or during different timeslots, denoted by $X = \{x_1, x_2, \cdots, x_N\}$ and $Y = \{y_1, y_2, \cdots, y_N\}$. Data entries X and Y are then divided into a training set (X_{train} and Y_{train}) and a testing set (X_{test} and Y_{test}). We then build

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2921959, IEEE Transactions on Mobile Computing

a model based on X_{train} and Y_{train} to learn the relationship $f(x)$ between X and Y such that $y = f(x)$. The validation of the model is by predicting a $Y_{predict}$ based on X_{test}, and comparing between $Y_{predict}$ and Y_{test}.

3.2 System Framework of ROD-Revenue

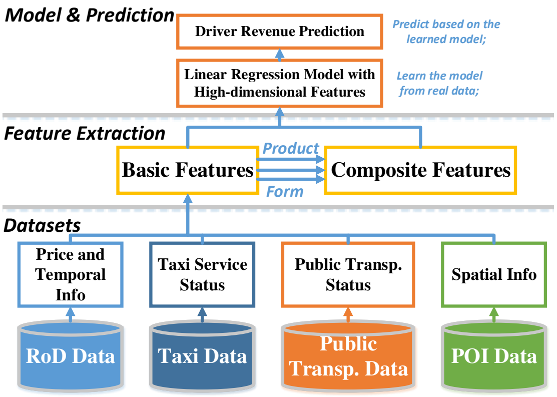


Fig. 1. The system framework of RoD-Revenue.

As illustrated in Fig. 1, ROD-Revenue consists of three main layers: datasets, feature extraction, model & prediction. We discuss briefly each of them below.

Datasets. We use multi-source urban datasets as the fundamental component of ROD-Revenue. Datasets are obtained from RoD service, taxi service, POI information, and public transportation services. These datasets are used to describe the spatio-temporal information as well as the status of other transportation services from different perspectives.

Feature Extraction. For each driver during each timeslot, we extract corresponding feature set. We have two categories of features, namely basic features and composite features. Basic features are those extracted from each single dataset. Features from RoD service give the temporal and price information; features from taxi service and other public transportation services describe the status of these services around the seeking locations; features from POI information characterize the function and category of these locations. Composite features are those combined from basic features in a product-form. We use composite features to compensate for the lack of non-linearity in our linear model.

Model & Prediction. Based on the basic and composite features, we build a linear regression model to learn the relationship between driver revenue and seeking strategies based on the training dataset, and use the learned model to predict driver revenue based on the test dataset. We also measure the difference between the predicted revenue and ground truth in test dataset to evaluate model accuracy.

In the following sections, we elaborate on each layer with more details: §4 for datasets, §5 for the extraction of basic features, §6 for the extraction of composite features, and §7 for the model and prediction.

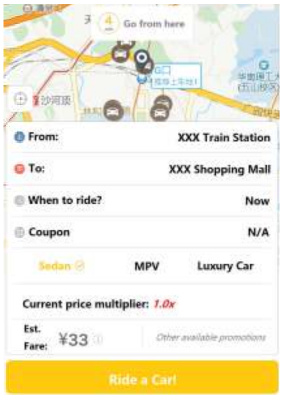
4 Multi-source Urban Data

Multi-source urban datasets are the fundamental component of ROD-Revenue. In this section, we explain the RoD service data, taxi service data, bus & metro distribution data, and POI data. Tab. 1 summarizes the datasets and fields.

4.1 RoD Service Data

The use of mobile apps for both passengers and drivers to access RoD services is a key enabler of our research. In traditional taxi service, most cars are now equipped with GPS devices that upload GPS trajectories, and in recent years there are an emerging usage of mobile apps to assist the matching between drivers and passengers. But in a RoD service, all communication messages between passengers, drivers, and the service provider are carried out through mobile apps, and there is not any other way of matching between drivers and passengers such as street-hailing. Hence, in addition to the car GPS trajectories data typically used in taxi studies, now we have more data to rely on.

Our data is from Shenzhou UCar (https://bit.ly/2MG47xz), a major RoD service provider in China. Fig. 2 shows the user interface of its app, and we use it to explain the work-flow of a typical RoD service. One types the boarding location A and arriving location B and could also choose “when to ride” and “using coupon”. After these steps, the app sends 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 (i.e., “Ride a Car!”) or give up the current fare estimation if s/he considers the price multiplier too high.



From: XXX Train Station
To: XXX Shopping Mall
When to ride? Now
Coupon N/A
Sedan ✓ MPV Luxury Car
Current price multiplier: 1.0x
Est. Fare: ¥33
Other available promotions

Fig. 2. The user interface of a typical RoD service.

We obtain three different datasets from RoD service:

The Order Data. Each entry represents a single order from a passenger, containing the order’s boarding/arriving time and location, the unique ID of the user/driver/car/order, the type of order, etc. We use the data in Beijing, as it is one of the most representative metropolitan cities and also the biggest market of the service provider. The dataset lasts for 4 months, from Nov. 2015 to Mar. 2016, and contains about 2.7 million entries for Beijing alone. All entries are properly anonymized.

GPS Trajectories. This dataset covers the GPS trajectories of every single car under the service provider. Each car uploads its location to the service provider every two minutes, and the upload period becomes longer (varying from five to ten minutes) when the car is out of service (e.g., the driver is taking a rest, having lunch, etc.). Each entry represents one particular data upload with information such

TABLE 1
A summary of datasets and fields.

Dataset	Fields
RoD	**Order**: boarding location, boarding time, arriving location, arriving time, user ID, driver ID, car ID, order ID/type. **GPS Trajectories**: car location, upload time, car ID, car number plate. **Event-log**: event time, event location, estimated fare, price multiplier, user ID.
Taxi	**GPS Trajectories**: taxi location, upload time, speed, full flag, car number plate.
Bus & metro	the number of bus stations, bus lines, metro stations, metro lines.
POI	the number of POIs of 14 categories mentioned in §4.4.

as the location of the car (i.e., the longitude and latitude), data upload time, the unique ID and the number plate of the car, etc. The time range of the dataset is the same to the order data, and on each day the number of cars on the road is about 3,500 for the service provider.

The Event-log Data. 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. Our event-log data contains the record of this event in the same time range. Each entry corresponds to a single event, and includes fields such as event time, event location (i.e., the longitude and latitude when the user triggers the event), estimated fare, price multiplier, the unique user ID, etc. The dataset contains 14,832,418 entries.

The event-log data gives clues about dynamic prices: it covers more information than the orders created by passengers, as those fare estimations that do not lead to order creations are also recorded. For a particular time period and a region, we can calculate the average price multipliers of all events in this region during this period, and it tells "how different price multipliers could be in different locations or during different time periods?". In [16] the authors show the average price multiplier during different time periods around Beijing, and a very simple observation is that the dynamic prices are related to temporal and spatial factors, so are the drivers' revenue.

These three datasets help us to obtain information about both the passenger delivery trip and driver seeking trip. The feature extraction process in details will be discussed later.

Passenger Delivery Trip. We can extract the following information about a delivery trip: the boarding/arriving time and locations are directly from the order data; and the trip distance and order revenue can also be estimated. Details will be discussed in §5.1.1.

Driver Seeking Trip. A seeking trip is defined as the trip from the arriving location of one order to the boarding location of the next order. The starting and ending points (time and locations) of a seeking trip are just the arriving time/location of one order, and the boarding time/location of the next order.

4.2 Taxi Service GPS Trajectory Data

The motivation of using taxi GPS trajectory data along with RoD service data is two-fold. Firstly, we envision that the operation status of RoD service is relevant to taxi service, as RoD service is similar to taxi service in many ways. Thus, the profitability of a particular seeking strategy maybe related to the status of taxi service along the seeking routes. Secondly, the taxi service GPS trajectories help to characterize the general traffic condition of different locations. For examples, "*whether a region is busy during a particular time period*", "*the number of available taxis around a region*", etc.

Our dataset covers the GPS trajectories of about 30,000 taxis in Beijing in November, 2015. Similar to RoD service, each taxi uploads one GPS entry every 30 seconds during operation. For each day, the volume of dataset ranges from 45 to 50 million entries. Each entry contains information such as the location of the car, upload time, speed, full flag (i.e., whether the taxi is available), the number plate, etc.

4.3 Bus & Metro Distribution Data

We use this dataset to describe the availability of public transportation around different locations, as the profitability of a seeking strategy may be related to the status of public transportation services around.

We count the number of bus & metro lines and stations within a 500-meter radius of a given location. It is true that the most accurate description should be the availability of bus & metro around, but as bus & metro have relatively fixed time tables, most people decide whether to take public transportation based on the availability of bus & metro lines or stations nearby, instead of the exact number of buses or metro trains. The dataset is crawled from AMap service [26] (one of the largest digital map service providers in China). For the whole city, there are more than 7,700 bus stations and about 380 metro stations.

4.4 POI Data

The goal of using POI (point of interest) information is that we want to extract some POI features to characterize a particular location. For example, the average price multiplier is much higher in some part of the city (e.g., some business areas) during evening rush hour than in other locations. We want to find out some features to accurately describe the differences between locations.

We also crawl POI data from AMap service. It categorizes each POI into 14 coarse categories: *car service*, *restaurant*, *shopping*, *sports & entertainment*, *hospital*, *hotel*, *scenic spot*, *business & residential building*, *government*, *education & culture*, *transportation facility*, *finance & insurance*, *business* and *lifestyle*. For a location given, we count the number of POIs of each category within a 500-meter radius of the location, and use the resulting vector as our POI data.

Essentially, the *POI-counts* data we collect describes a particular location with the number of POIs of different categories that appear around this location. Some previous work used the nearest POI and its category to describe a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2921959, IEEE Transactions on Mobile Computing

location, and we do not adopt this idea, as we consider it not an accurate characterization of a location. For example, a passenger standing out of a shopping mall may have the nearest POI as a restaurant, but the reason of waiting here turns out to be the shopping mall instead of the restaurant.

5 Feature Extraction: Basic Features

Our study is on a weekly basis. For each driver driving on a particular day-of-week, we gather his/her passenger orders and seeking trips during each timeslot, and calculate the corresponding features based on our multi-source datasets.

Basic features are those extracted from each dataset, and in the following we elaborate on them in more details.

5.1 Features from RoD Service

RoD service features are the most fundamental in our study. We extract features about both passenger delivery trips and driver seeking trips. All features are extracted in the unit of timeslot for each driver on a particular day-of-week. The common features for delivery and seeking trips are the temporal features: *day-of-week* and *timeslot-of-day*. Below we elaborate on features related to passenger delivery trips and driver seeking trips separately.

5.1.1 Features about Passenger Delivery Trips

For passenger delivery trips, the goal is to calculate the *average delivery speed* and the *hourly average revenue* of a driver in a timeslot.

Average Delivery Speed. We first calculate the trip distance of each passenger delivery trip of a driver in a timeslot. The order dataset provides the boarding and arriving location of an order, but the straight line distance between these two locations is only a rough estimate of the trip distance. The RoD service GPS trajectories are used to approximately calculate the distance. Specifically, the GPS trajectories of a single car in one day consist of a series of points $(t_i, lon_i, lat_i)(1 \leq i \leq n)$. n is the total number of data points, and t_i, lon_i, lat_i are the data upload time, longitude and latitude of the i-th point, respectively. For a single order, we use T_{board} and T_{arrive} to denote the boarding and arriving time, and find the $board\_L$, $board\_R$, $arrive\_L$, $arrive\_R$-th data points on the GPS trajectories such that $t_{board\_L} \leq T_{board} \leq t_{board\_R}$ and $t_{arrive\_L} \leq T_{arrive} \leq t_{arrive\_R}$. We then use two trajectories with a slight difference to approach the real distance: one from t_{board_L} to t_{arrive_L}, and another from t_{board_R} to t_{arrive_R}. For each trajectory, the trip distance is approximated by the sum of straight line distances between adjacent points. The trip distance of this order is the average of distances of the two trajectories.

The *average delivery speed* v_{avg} of a driver in a timeslot can then be calculated. Assuming that this driver serves K orders during this timeslot, with boarding time $T_{board,k}$, arriving time $T_{arrive,k}$ and trip distance d_k $(1 \leq k \leq K)$, then v_{avg} is:

$$v_{avg} = \frac{\sum_{k=1}^{K} d_k}{\sum_{k=1}^{K} T_{arrive,k} - T_{board,k}}. \quad (2)$$

We use *average delivery speed* as a feature, as it reflects a driver's ability to choose faster routes in serving a passenger, which is an important metric in driver evaluation. It is also a reflection of the traffic condition along the delivery routes, when there is no faster routes to choose from.

Hourly Average Revenue. The *hourly average revenue* of a driver in a timeslot is the target of our model. Calculating the hourly average revenue requires the trip fare of every single passenger delivery trip of a driver, but our order dataset has a limitation that there is not a total trip fare or dynamic price multiplier associated with each order. This limitation may be due to the privacy policy. Hence, we try to estimate the trip fare as well as the dynamic price multiplier.

Specifically, we divide the map of Beijing into 2500 $(= 50*50)$ rectangular cells of the same size, so the boarding location of the order falls in one cell. Then we gather all *EstimateFee* events from the event-log dataset in this cell taking place during the same hour and on the same day-of-week with the order, and use the average price multiplier contained in these events to approximate the price multiplier of the order. We use p_k to denote the estimated price multiplier for the k-th order $(1 \leq k \leq K)$ and f_k as the estimated trip fare of the k-th order, then we have:

$$f_k = p_k * (15 + 2.8 * d_k), 1 \leq k \leq K. \quad (3)$$

In (3), the service provider sets the flag-fall to be 15 Yuan in RMB (≈ 2.18 USD), and each additional kilometre costs 2.8 Yuan (≈ 0.41 USD). As an approximation, our estimated trip fare omits the waiting charge, as it is hard to accurately estimate the waiting time only from the GPS trajectories.

Visualizations and Analysis. We show some visualizations of the intermediate quantities mentioned above, as well as the *average delivery speed* and *hourly average revenue*, based on our RoD service datasets. The goal is provide some intuitive understanding and insights about these features.

Fig. 3 shows the hourly variation of order distance d_k, order revenue f_k and order average speed. The red dot is the corresponding mean value, and the error bar indicates the standard deviation. Interesting observations include:

- For order distance, it is significantly higher during night, and becomes the lowest during morning/evening rush hours on weekdays. This agrees to our intuition that people take longer rides at night.
- For order revenue, the temporal difference is smaller during the day, compared to that of order distance. But during morning rush hours the revenue is still the lowest. The reduced temporal difference is a result of higher dynamic prices during rush hours.
- The average speed also shows similar patterns with more obvious fluctuation. It can be regarded as an indication of the traffic condition, and it is clear that the speed is much slower during rush hours.
- During weekends, the fluctuation of these three quantities is much less obvious: during the day (e.g., [8am, 8pm]), the order distance, revenue and speed all remain more stable.

Fig. 4 and 5 show the distribution of hourly average revenue among all drivers in each timeslot. The y-axis of these figures is proportional to the probability, and may not integrate to one. Similarly, Fig. 6 and 7 show the distribution of average delivery speed among all drivers in each timeslot.

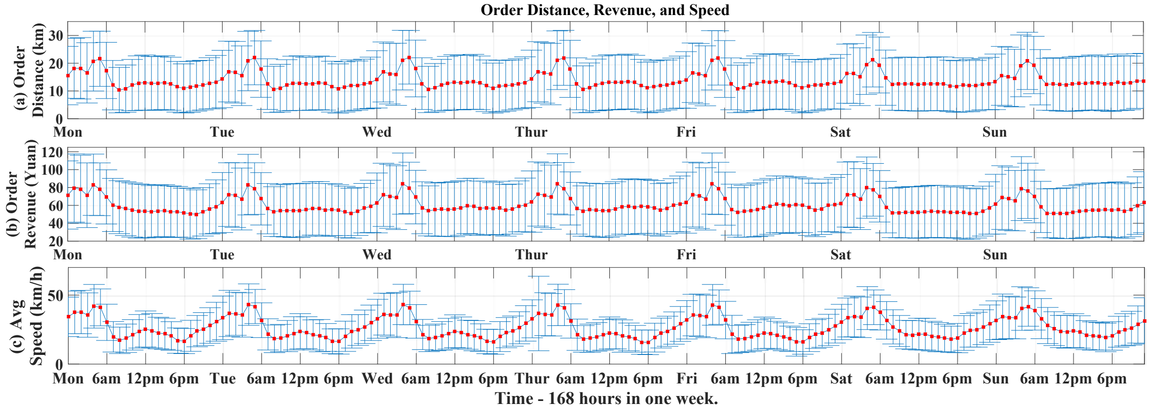


Fig. 3. The variation of order distance, revenue and speed in one week.

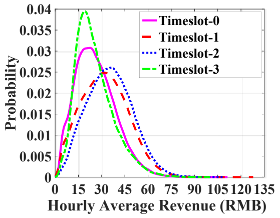


Fig. 4. The distribution of hourly average revenue on weekdays.

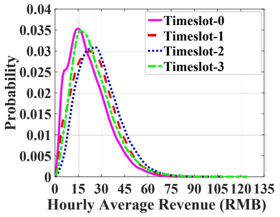


Fig. 5. The distribution of hourly average revenue on weekends.

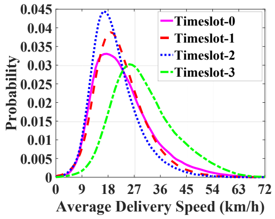


Fig. 6. The distribution of average delivery speed on weekdays.

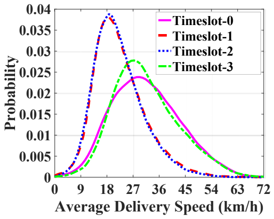


Fig. 7. The distribution of average delivery speed on weekends.

Regarding the hourly average revenue of drivers, we have the following observations:

- The distribution is more even during weekends: drivers make similar hourly revenue during the day. Comparatively, the hourly average revenue fluctuates more obviously between different timeslots.
- For weekdays, the hourly average revenue is the highest during the evening rush hours (i.e., timeslot-2), and then the non-rush hours around noon (i.e., timeslot-1), on the average driver level.
- Comparing between morning rush hours (i.e., timeslot-0) and night hours (i.e., timeslot-3), the hourly average revenue distributes more evenly during morning rush hours than during night hours. In other words, only very few drivers can make higher revenue during night hours, and it is comparatively easier to earn more during morning rush hours.
- Similar to the order revenue observed in Fig. 3, it is interesting to find that the hourly average revenue is higher during the non-rush hours around noon, than during the morning rush hours. The hourly average revenue around noon is also very close to that during the evening rush hours. This is, to some extent, counter-intuitive to our experience, and we will validate this later in our model.

Similarly, we have the following observations regarding the average delivery speed:

- On weekends, the average delivery speed has roughly the same distribution during the day (i.e., timeslot-1 and 2), and becomes higher during other timeslots. More accurately, there is not such concepts of rush hours during weekends; instead, people's activities are more evenly spread across the day.
- On weekdays, the average delivery speed is significantly lower during morning and evening rush hours, than during other timeslots. Additionally, driving at high speed during evening rush hours requires more careful strategies than during morning rush hours, as the distribution of delivery speed is much sharper during evening rush hours, meaning that fewer drivers can achieve higher speeds.
- Comparing between Fig. 4 and Fig. 6, most drivers are able to maintain a relatively high speed during night hours, but only a few can make higher revenue during this timeslot. This shows that the hourly average revenue has a rather complex relationship with average delivery speed – being able to driving faster or choosing clearer routes does not necessarily lead to a higher revenue.

5.1.2 Features about Driver Seeking Trips

For driver seeking trip, our goal is to generate features to describe some high-level characteristics of seeking strategies. Taking the same notations in §5.1.1, if the number of delivery trips a driver serves during a timeslot is K, then the number of seeking trips is $M = K - 1$. In the following, we first identify the price-chasing strategy, then the price multipliers around seeking locations. We also extract features from other datasets to characterize the seeking locations, and more details can be found in §5.2.

Basically, the characterization of seeking trips is based on the corresponding starting and ending points. After closing an order, a driver start a seeking trip at the arriving location of this order, and this seeking trip comes to an end when another order starts. We compare the starting and ending point of a seeking trip from different perspectives, to fully characterize the seeking strategy represented in such a trip.

It may be more accurate to dig into more details about the seeking trip than just the starting and ending points, for example, the exact GPS trajectories between these points and how drivers take turns, accelerate, brake, etc. But we don't do that in our study due to two reasons. Firstly, going into that details is over-fitting to some extent, as these driver behaviors may not be the result of seeking strategies; instead, they may be spontaneous and due to some unplanned reasons such as traffic condition, events, accidents, etc. Secondly, an accurate description of these behaviors requires datasets other than our RoD service datasets currently in use, such as data obtained from wearable devices.

For one particular driver during a particular timeslot, there may be more than one seeking trips. The ideal way is to design some features that describe *each* of these seeking trips. One possible solution is to divide the city into a number of pre-defined regions with some pre-defined characteristics (e.g., business regions, residential regions, airports, etc.), and then consider the number of seeking trips that fall in each region. But choosing regions and corresponding characteristics requires prior knowledge of the city and thus seems too artificial. Instead, we choose to describe the starting and ending points of a seeking trip based on the features extracted from our multi-source urban datasets. However, the number of seeking trips is not fixed, and if we describe *each* of these trips, the dimension of features will also become variable, adding complexity to our model. Hence, we choose to describe the *collective* properties (e.g., *average*, *minimum* or *maximum* of some properties) of all seeking trips of one driver during a timeslot, similar to the way we generate the *hourly average revenue* and *average delivery speed* features. Though losing some information, collective properties still retain the essence of information about seeking trips.

Strategy Factor. Strategy factor focuses on whether a driver is chasing dynamic price multiplier. We first define the *price-chasing strategy* of each seeking trip. The rationale of identifying price-chasing strategy is that we want to see if strategies such as "surge chasing" work. A seeking trip is categorized into three different strategies:

- *chasing current*: in the hour of starting seeking, if the average price multiplier around the ending point is higher than that around the starting point;
- *chasing future*: in the hour of starting seeking, if the average price multiplier around the ending point in the next hour is higher than that around the starting point;
- *no chasing*: if neither of the above holds.

A seeking trip can be of *chasing* or *no chasing* strategy; and for *chasing*, it can be either *chasing current* or *chasing future*.

The *strategy factor* vector indicates a driver's preference on seeking strategy in a timeslot, and is defined as the number of seeking trips of each category of a driver in a timeslot. It is a 3-dimension vector $(N_{current}, N_{future}, N_{non})$, referring to the number of seeking trips of each strategy.

Price Multipliers of Seeking Locations. The strategy factor describes the relative relationship of dynamic prices between the starting and ending points of seeking trips. Now we turn to the absolute values of dynamic price multipliers of the starting and ending points.

We discuss how to define a feature *starting points' price multipliers*, and for ending points the procedure is similar. For the m-th seeking trip $(1 \leq m \leq M)$, we calculate the average price multiplier around its starting point in the last hour, now, and next hour, denoted by $p_{m,last}, p_{m,now}, p_{m,next}$, respectively, from the event-log data. We then traverse all seeking trips, and calculate the average, minimum and maximum values among all $p_{m,last}, p_{m,now}, p_{m,next}$ for $1 \leq m \leq M$. Thus, we generate 9 different values: p_{avg_last}, p_{min_last}, p_{max_last}, p_{avg_now}, p_{min_now}, p_{max_now}, p_{avg_next}, p_{min_next}, and p_{max_next}. They form the vector *starting points' price multipliers*. Similarly, we define the vector *ending points' price multipliers*.

These two vectors, *starting/ending points' price multipliers*, describe the price multipliers around the seeking trips' starting and ending points, in the current and neighboring hours. We consider these vectors as a representation of dynamic prices of every seeking trip, at a high and average level.

5.2 Features from Other Datasets

Features extracted from other datasets (i.e., taxi service, bus & metro, and POI) are used to characterize the starting and ending points of seeking trips, from perspectives such as the status of taxi service, the traffic condition, the availability of public transportation services and POI information.

5.2.1 Features from Taxi Service

Features from taxi service data are used to describe the status of taxi service as well as traffic condition around the starting/ending points of seeking trips, and they form feature vectors *starting/ending points' taxi status*. We discuss how to form *starting points' taxi status*, and for ending points the procedure is similar.

From the taxi trip information, we extract two features:

- *up/down count*: the number of orders starting/ending around the starting points.

From taxi GPS trajectories, we extract five features describing taxis around the starting points of seeking trips:

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

Among these features, *average speed* and *speed variance* describe the traffic condition around the starting points of seeking trips, and other features describe either the availability of taxis, the popularity of the location, or passengers' demand for taxis around the starting points. For all these seven features, we calculate each of them based on the GPS trajectories that fall in the same hour-of-day (i.e., "hourly taxi features"), and in the same hour-of-day and day-of-week (i.e., "daily taxi features"). In this way, we obtain 14 different features from taxi service, and they form the feature vector *starting points' taxi status*. For ending points of seeking trips, the vector *ending points' taxi status* is constructed similarly, but with features around the ending points.

5.2.2 Features from Bus & Metro, and POI Data

Features from the bus & metro distribution data describe the availability of public transportation services such as bus or metro around the starting or ending points of seeking trips. Specifically, we build a feature vector *starting points' bus & metro*, a 4-dimension vector containing the number of bus stations, bus lines, metro stations, metro lines around the starting points of seeking trips. We also build the vector *ending points' bus & metro* in a similar way.

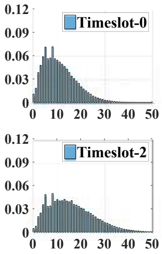


Fig. 8. The histogram of bus station counts around ending points of seeking locations on weekdays.

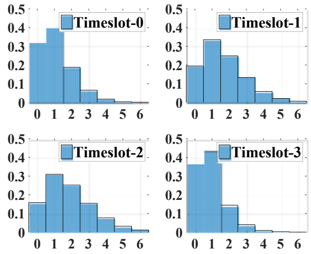


Fig. 9. The histogram of metro station counts around ending points of seeking locations on weekdays.

Regarding the driver seeking behavior mined from our data, Fig. 8 and 9 show the histogram of bus and metro station counts around ending points of seeking locations during each timeslot on weekdays. Figures of weekends are omitted due to the limited space. Observations are:

- On weekdays and weekends, drivers tend to go to regions with very few bus stations to seek for passengers during night hours – the lack of bus stations means higher demand for taxi or RoD service. Things are similar during the morning on weekends, due to the lack of human activity in this timeslot.
- On weekdays, the histograms of bus stations during the day (i.e., timeslot-1 and -2) are more evenly spread – drivers have fewer specific choices regarding bus stations, but locations with more bus stations are favoured, as having more bus stations means that the location has a higher popularity.
- On weekdays, in morning rush hours the average number of bus stations around seeking locations is between that during the day and night hours.
- For metro stations, we have the similar observations, but with fewer number of stations, as metro stations are much more sparsely distributed in the city than bus stations. This also shows that drivers take into account bus or metro stations in the same way in determining their seeking strategies.

Features from the POI data help to describe the starting and ending points of seeking trip by characterizing the usage of these locations – *the number of POIs of different categories around these locations*. Our POI data categorizes POIs into 14 categories (see §4.4), and these 14 values form the feature vector *starting/ending points' POI counts*.

6 Feature Extraction: Composite Features

The need to introduce composite features comes from the lack of non-linearity in a linear regression model. Without non-linear terms, a linear regression model is unable to involve the non-linear relationship between features, and thus has a relatively lower accuracy in fitting the data.

Adding 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

$$y = \omega_1 x_1 + \omega_2 x_2 + b. \tag{4}$$

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:

$$y = \omega_1' x_1 + \omega_2' x_2 + \omega_3' x_3 + b'. \tag{5}$$

In (4) and (5), ω_1, ω_2, b, ω_1', ω_2', ω_3' and b' are the model parameters learned. Changing from (4) to (5) 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 composite features in our study are just the multiplicative product of multiple basic features.

6.1 Normalization of Basic Features

It is necessary to normalize basic features for fast convergence of SGD (stochastic gradient descent) regression, and for unifying the units and meanings of different basic features.

There are two different kinds of feature: numerical and categorical feature. For a numerical feature (e.g., *average delivery speed*, the number of bus stations of *starting points' bus & metro*, etc.), we apply the min-max normalization [27] to make it between 0 and 1. For a categorical feature (e.g., *day-of-week*), we apply one-hot extension to transform it into a vector, the dimension of which is the number of categories. There is nothing to normalize for a categorical feature, as the maximum value of any component is 1.

6.2 Combination of Basic Features

We have already defined composite features as the combination of basic features in a product form. Specifically, now we show how to perform this combination under different circumstances, i.e., whether the basic features are numerical or categorical features.

We use x_1 and x_2 to denote two basic features, and x_3 to denote the resulting composite feature. The calculation of x_3 can be one of the following three circumstances:

- if x_1 and x_2 are numerical features, then x_3 is also a numerical feature, and $x_3 = x_1 x_2$;
- if x_1 is a numerical feature and $\vec{x_2}$ is a categorical feature, then x_3 is also a vector and $\vec{x_3} = x_1 \vec{x_2}$.
- if $\vec{x_1} = (x_{11}, x_{12}, ..., x_{1n_1})$ and $\vec{x_2} = (x_{21}, x_{22}, ..., x_{2n_2})$ are categorical features of dimension n_1 and n_2, then the resulting vector $\vec{x_3}$ has a dimension $n_3 = n_1 n_2$ and can be written as $\vec{x_3} = (x_{11}x_{21}, x_{11}x_{22}, ..., x_{11}x_{2n_2}, x_{12}x_{21}, x_{12}x_{22}, ..., x_{12}$ $x_{2n_2}, ..., x_{1n_1}x_{21}, x_{1n_1}x_{22}, ..., x_{1n_1}x_{2n_2})$.

TABLE 2
Feature extraction: some selected composite features.

Type	Datasets	Examples of combinations
Same dataset	RoD+RoD	(*day-of-week*, *timeslot-of-day*), (*strategy factor*, *starting points' price multipliers*), (*average delivery speed*, *ending points' price multipliers*)...
	Taxi+Taxi	(*starting points' taxi status*, *ending points' taxi status*)
Different dataset	RoD+Taxi	(*timeslot-of-day*, *starting/ending points' taxi status*)...
	RoD+Bus&metro	(*timeslot-of-day*, *starting/ending points' bus & metro*)...
	RoD+POI	(*strategy factor*, *starting points' POI counts*), (*day-of-week*, *starting points' POI counts*)
	Taxi+POI	(*starting points' taxi status*, *ending points' POI counts*)...
	Taxi+Bus&metro	(*starting points' taxi status*, *ending points' bus & metro*)...

6.3 Examples of Composite Features

It is possible to combine virtually any two basic features to form composite features, and judge the effects of the combination (i.e., whether this composite feature is necessary) by the corresponding weight in the trained linear model. Because of limited space, we give some illustrative examples below. More examples can be found in Tab. 2.

Combining features from the same dataset. Combination of this kind tries to express the correlation between features in the same dataset. For example, a composite feature (*day-of-week*, *timeslot-of-day*) tries to correlate the *day-of-week* feature with *timeslot-of-day* feature, and its weight shows the joint impact of both *day-of-week* and *timeslot-of-day* on the hourly average revenue.

Combining features from different datasets. Combining basic features from different datasets not only helps to express the correlation between these features, but also shows the relationship between different datasets. For example, (*timeslot-of-day*, *ending points' taxi status*) reflects how RoD service interacts with the status of taxi service. As another example, the weight of (*timeslot-of-day*, *ending points' bus & metro*) indicates different levels of profitability of finding bus or metro stations during different timeslots.

In fact, the use of composite features combined from different datasets enables us to quantify and interpret the relationship between driver revenue and relevant features extracted from other datasets, such as the status of taxi, bus and metro service, POI information, etc. These composite features also tell us, when coupled together, the importance of features under different circumstances.

6.4 The Growing Dimensions of Features

The dimension of features grows tremendously with the introduction of composite features. When combining two basic features of dimension n_1 and n_2, the resulting composite feature has a dimension of $n_1 n_2$. The growth is much faster if we combine more than two basic features to form composite features. To be more concrete, our basic features account for a dimension of 97, and when combining any two basic features, the resulting dimension of both basic and composite features climbs to $3,730$. If we choose some groups of three basic features to form composite features, the dimension will be higher than $13,000$.

In our study we only combine any two basic features to form composite features. There are multiple considerations of not using more than two basic features in combination:

- We actually try to combine three basic features to form composite features, and our evaluation shows that while the training time is more than tripled, the model's accuracy does not improve significantly.
- Composite features combined from three or more basic features have a reduced interpretability. It becomes harder to quantify and interpret the joint effects of three or more features as well as the relationship between driver revenue and these features.

Generating composite features with only any two basic features does not bring huge challenges to our model's performance because:

- When only combining two basic features, the total dimension of our basic and composite features is $3,730$. This is not a very high dimension and we can directly apply our linear model on the features.
- At this dimension of features, it takes about 12 minutes to train the model based on the complete training set (taking about 70% of data, as discussed in §7) on an ordinary Intel Core i7-8700K personal computer. Though this seems to be a long time, the mini-batch training paradigm of a linear regression model can help to significantly reduce the training time. With mini-batch SGD (stochastic gradient descent) incremental model training, a batch is much smaller than the whole training set, and it takes less than 15 seconds to train.

7 Model & Prediction

"Model & Prediction" sits on the highest level of RoD-Revenue's framework. Basically, we build a model to learn the relationship between driver revenue and seeking strategies, and use the model to predict driver revenue based on their seeking strategies. The prediction part serves as an evaluation of the model's accuracy: we split our dataset into a training set and a test set, and the model is learned based on the training set, and then is evaluated by predicting driver revenue based on the test set. Undoubtedly, the prediction can also be performed on new or incoming data, as long as they share similar characteristics with the training set.

In the following, we first present the model we use, then the evaluation metrics, followed by our experiment results in predicting driver revenue based on the learned model.

7.1 The Linear Regression Model

The choice of the model to mine the relationship between driver revenue and seeking strategies and solve the problem defined in §3.1 is actually a trade-off between accuracy and interpretability. Complex, non-linear models such as neural network and deep learning models may give highly accurate results when carefully tuned, with a relatively small dimension of features; but these models are generally hard to interpret, and even quantifying the feature contribution at a high level sometimes requires complicated methodologies. Some simpler, linear models such as the decision tree family, though interpretable by nature, require a complex structure to improve accuracy, and these derivatives such as random forest or GBRT thus have a diminished interpretability. On the other hand, a linear regression model is one of the simplest models with interpretability – the weight of each feature or rather, each component of a multi-dimensional feature, shows how important this feature or feature component is on the target variable. But an increased level of interpretability leads to a decreased accuracy: the lack of non-linear terms in a linear regression model makes it hard to characterize non-linear correlations between features.

In ROD-Revenue, we want to quantify and explain the relationship between driver revenue and seeking strategies specifically, so that it is possible to offer concrete suggestions to drivers about how to earn more. Hence, we choose to use the linear regression model. To deal with the inaccuracies due to the lack of non-linear terms, we introduce composite features, a product-form term based on basic features, to add non-linear terms into the linear regression model.

Following the notations in §3.1, y denotes the *hourly average revenue* of one driver during a timeslot, and $x \in \mathbb{R}^m$ denotes the feature vector corresponding to y, with m as the dimension of the feature vector. As discussed in §5 and §6, x contains the basic and composite features extracted from multi-source urban data, and m is $3,730$. We write our raw feature dataset as $D = \{(x_i, y_i) | i = 1, 2, \ldots, N\}$, where (x_i, y_i) represents the i-th sample. For N, we compress the 4-month data into one week as we only consider the differences between days-of-week, and obtain $N = 802,600$ data entries for drivers in timeslots. We train the linear model batch-by-batch, and D also represents the raw feature dataset in each batch.

The parameters to be learned is a parameter vector, ω, of the same dimension of x, and an intercept b. The output of the model, p_i, with the input of x_i, can be written as $p_i = \omega^T x_i + b$. In training the model, the goal is to calculate ω and b such that the sum of the squared differences between y_i and p_i is minimized. Specifically, we use a simple form of linear regression: the squared error loss objective function with $L1$ and $L2$ regularization. The objective function can be written as:

$$obj(\omega, b) = \sum_{(x_i, y_i) \in D} (y_i - p_i)^2 + \lambda_1 ||\omega||_1 + \lambda_2 ||\omega||_2. \quad (6)$$

In (6), the first term is the squared error loss, and the latter two terms are for $L1$ and $L2$ regularizations, with λ_1 and λ_2 as the trade-off parameters. The $L1$ regularization uses a $L1$-norm of ω to control the sparsity of the learned parameter ω, so that there are more components of ω becoming zero or close to zero. Controlling the sparsity is for better interpretability, as it makes fewer features have a strong influence on the driver revenue, while a lot others get a zero weight in the learned model, meaning that these features are irrelevant to the driver revenue. On the other hand, using $L2$ regularization is a common practice in machine learning to avoid over-fitting, so that there will not be a huge gap between the model's performance on the training and on the test set. It controls the $L2$-norm of ω so that any component of ω (and features in x) should not have an overwhelming influence on the driver revenue.

With the objective function (6) to minimize, we then use the stochastic gradient descent (SGD) to minimize the function based on the training set, and obtain a linear regression model with parameters ω and b.

7.2 Evaluation Metrics

In our evaluation in this section, we first examine the correlation between driver revenue and different features, then evaluate the performance of our model in describing the relationship between driver revenue and seeking strategies. Finally we will provide both qualitative and quantitative discussions on feature contribution in §8.

To examine the correlation between driver revenue and the extracted features about seeking strategies, we use the Pearson correlation coefficient (PCC) to measure the correlation between driver revenue and extracted features. Higher PCC means higher correlation, and the corresponding feature is more relevant to driver revenue. For a particular numerical feature (i.e., a numerical feature or a component of a multi-dimensional numerical feature) with values $r_i (1 \leq i \leq N)$, its PCC, denoted by $PCC(r, y)$, with the hourly average revenue y_i, can be calculated as:

$$PCC(r, y) = \frac{\sum_{i=1}^{N} (r_i - \overline{r})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N} (r_i - \overline{r})^2} \sqrt{\sum_{i=1}^{N} (y_i - \overline{y})^2}} \quad (7)$$

In evaluating the performance of the model, we randomly choose 70% and the remaining 30% entries as the training and test set, out of our raw feature dataset containing 802,600 entries with a dimension of $3,730$. The random selection is performed for 10 times and the average metric is used for our evaluation. As to the evaluation metric of model performance, we use MAE (mean absolute error) to evaluate model accuracy. We use N_{test} to denote the number of data entries in the test set, and MAE is defined as:

$$MAE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |y_i - p_i| \quad (8)$$

The target variable, i.e., the hourly average revenue (in RMB), is already a quantity easy to interpret and understand; and the absolute error itself also has significant meaning. Thus, it is not necessary to use a scale-independent evaluation metric such as sMAPE (symmetric mean absolute percentage error).

7.3 Experiment Results

7.3.1 Correlation Analysis

We calculate the Pearson correlation coefficient (PCC) between driver revenue and features relevant to seeking strategies to see the effectiveness of our extracted features. Tab. 3 shows the PCC between driver revenue and some selected numerical basic features. Basically, we select some

TABLE 3
The PCC between driver revenue and some selected numerical basic features, on weekdays.

Feature	**PCC** in timeslot-0/-1/-2/-3
average delivery speed	0.3290, 0.2207, 0.2119, 0.4065
ending points' price multipliers:	
p_{max_now}	0.3203, 0.3148, 0.3599, 0.2746
p_{max_next}	0.3530, 0.3412, 0.3783, 0.3119
ending points' bus & metro:	
the number of bus stations	0.3472, 0.3818, 0.3944, 0.3338
the number of metro stations	0.3302, 0.4200, 0.4585, 0.3692
ending points' taxi status:	
full taxi ratio, daily	0.3304, 0.2577, 0.2896, 0.2312
down count, daily	0.3175, 0.2199, 0.2627, 0.2411
ending points' POI counts:	
business POI count	0.0121, 0.0317, 0.0335, 0.0047
shopping POI count	0.0088, 0.0245, 0.0323, 0.0051

(2 or 3) basic features extracted from each of our multi-source urban datasets that have the highest PCC. For simplification, we only consider the ending points of seeking trips when choosing relevant features. Features relevant to the starting points of seeking trips have similar but slightly smaller PCCs, and are now shown in Tab. 3. For temporal differences in PCC, we show the PCC of each feature in each timeslot on weekdays.

We have the following observations regarding Tab. 3:

- Except for features in *ending points' POI counts*, top features from other datasets show relatively close PCCs. In other words, all these features are correlated, to some extent, with the hourly average revenue of a driver, but non of them is purely linearly correlated with the hourly average revenue – indicating that a linear regression model with only basic features is not enough to characterize the relationship between driver revenue and seeking strategies.
- The PCCs of one particular feature may vary significantly in different timeslots-of-day. For example, The highest PCC between hourly average revenue and *average delivery speed* (i.e., in timeslot-3) is 91.84% more than the lowest corresponding PCC (i.e., in timeslot-2). This is another perspective showing that it is not enough to consider only basic features: a composite feature from *timeslot-of-day* and *average delivery speed* should add useful information to our model.
- Features extracted from POI information (i.e., *ending points' POI counts*) have much smaller correlation coefficients, compared with other features. This indicates that POI counts features are less correlated with driver revenue – we hypothesize that the reasons are the inability of POI counts to accurately describe location characteristics and the fact that location information is also revealed by features from other datasets such as the status of taxi, bus and metro. We will discuss them later.

In summary, results from correlation analysis verify that simply calculating the Pearson correlation coefficients is not enough to describe the complex relationship between driver revenue and seeking strategies. A linear regression model with only basic features is not enough neither. It is thus necessary to use a linear regression model with composite features to mine the desired relationship.

7.3.2 Evaluation of the Linear Model

Statistics show that among the test set, the average of the target variable (i.e., *hourly average revenue*), is $\overline{y} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} y_i = 29.113$. In other words, the average of "hourly average revenue" among all data entries in the test set is 29.113 Yuan.

Our linear regression model tries to predict the hourly average revenue given the input feature vectors in the test set, and finally calculates the difference between the predicted revenue and ground truth. The model gives a mean deviation of about 3.478 Yuan on the test set. Roughly speaking, the linear model can predict the hourly average revenue at an accuracy of 88.05%. For the prediction error (i.e., deviation) of individual test set entries, Fig. 10 and 11 show the distribution (i.e., a continuous version of histogram) of absolute and relative prediction errors.

We have the following observations:

- For absolute prediction error, during timeslot-3 the errors are smaller compared to other timeslots. Also, the distributions of absolute prediction errors of other timeslots very similar shapes.
- For relative prediction error, during timeslot-2 the relative errors are the smallest, followed by the the whole day, timeslot-1, -3 and -0. This agrees to earlier observations with Fig. 4: the hourly average revenue is probabilistically higher during timeslot-2, followed by timeslot-0 and -3.
- For absolute prediction error, the probability decreases steadily between 1 to 7 RMB. For timeslot-3, the decrease rate is faster, and for other three timeslots and the whole day, the decrease rate is slower. Moreover, for error larger then 7 RMB, the probability drops sharply, with very rare cases having an error greater than 8.5 RMB.
- For relative prediction error, the distributions of all timeslots have a long-tailed shape, and most relative prediction errors are between 5% and 20%.

To justify our choice of 6-hour timeslots, we also evaluate the prediction error when we choose 4-hour timeslots. It is shown that the mean deviation becomes 4.136, about 18.91% higher. This proves that our choice leads to better results.

7.3.3 Linear v.s. Non-linear Model

To compare between linear and non-linear model, we also build a neural network model to perform the exactly the same task. Neural network is a typical non-linear model, and the existence of non-linear correlation makes it enough to use basic features only. 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 input data fed to the input layer is a feature vector of 97 dimensions, containing only basic features. After careful tuning, our neural network model gives a mean deviation of about 2.977 Yuan on the test set – roughly an accuracy of 89.77%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2921959, IEEE Transactions on Mobile Computing

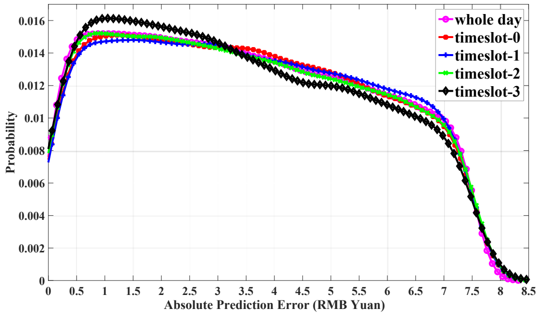


Fig. 10. The distribution of absolute prediction errors.

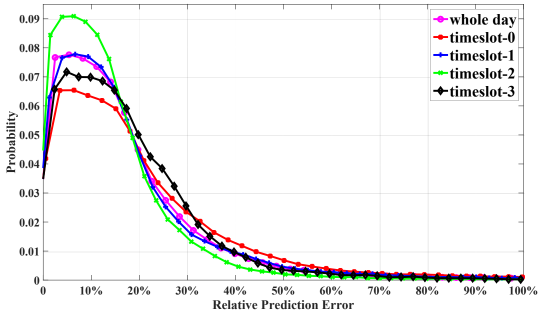


Fig. 11. The distribution of relative prediction errors.

The goal of comparing linear and non-linear model is to justify our model choice. The above results show that a non-linear model, even as simple as a four-layer neural network, achieves a higher (though slightly) accuracy than our linear model with composite features. But the differences are not only on the model accuracy and we give some discussions on other differences below, which justify our model choice:

The need for hyper-parameter tuning. A non-linear model such as neural network always requires careful tuning to perform well – our above-mentioned 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. Moreover, when the feature set changes, these hyper-parameters need to be re-tuned; in the linear model, on the other hand, it is only necessary to reconstruct composite features and re-train the linear model – a much easier task. Hence, a linear model works better in the case where features are updated constantly.

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 feature component is enough. This allows us to judge “what factors and what seeking strategies lead to a higher revenue, and by how much?”. A neural network model, on the other hand, does not offer this level of interpretability with simple inspection.

7.3.4 Basic v.s. Composite Features

In the correlation analysis, we observed that only using basic features in a linear regression model is not enough to describe the relationship between driver revenue and seeking strategies. In this section we validate this by comparing the MAE of using different combinations of features.

As mentioned in §7.3.2, with all composite features and basic features, our linear regression model gives a mean deviation in the predicted hourly average revenue of 3.478 Yuan on the test set. We train another linear regression model with *only basic features*, and the resulting deviation is 6.231 Yuan. This shows that using composite features from multi-source urban datasets can reduce the MAE of hourly average revenue by 44.18%.

Regarding the effects of generating composite features from different datasets, we also train linear regression models using both basic and composite features, with features coming from different combinations of datasets. The corresponding MAEs are shown in Tab. 4. If we only use the RoD data, but with composite features, the MAE is 4.874, and involving the additional taxi data, or bus & metro data, or POI data further reduces the MAE by 10.59%, 12.27%, 1.05%, respectively. When using all of the datasets, the reduction is 28.64%. This is a rough representation of the importance of multi-source urban datasets – “*how much the model's accuracy can be improved with the introduction of specific datasets*”.

TABLE 4
The MAE of using different combinations of datasets and composite features.

Feature sources	Resulting MAE
all datasets	3.478
RoD + bus & metro	4.276
RoD + taxi	4.358
RoD + POI	4.823
RoD only	4.874

We also try to combine more than two basic features to form composite features. Combining some groups of three basic features makes the dimension of feature vector more than $13,000$. We train a linear regression model correspondingly, and the resulting MAE is 3.336. Considering the significant increase in memory usage and a training time 3.8 times that of the original model, the slight improvement in the MAE is not worthwhile.

7.3.5 Effects of L1 and L2 Regularizations

The goal of using $L1$-regularization is to increase the number of zero weights in the trained model. In our linear regression model with $L1$-regularization, we have 514 zero weights out of the $3,730$-dimension weights. We also train a linear regression model without $L1$-regularization, and the resulting model has 216 zero weights, with a MAE of 3.476. These results show that, at a relatively the same accuracy, using $L1$-regularization almost doubles the number of zero weights, leading to better interpretability of the model.

$L2$-regularization, on the other hand, is used to avoid over-fitting, and we compare the MAE on the training and test set with and without $L2$-regularization. To do this, we use the model learned from the training set to predict the hourly average revenue based on the feature vectors in the training set. With $L2$-regularization, the model achieves a MAE of 3.478 on the test set, and 3.365 on the training set; without $L2$-regularization, the corresponding figures are 3.765 on the test set, and 3.298 on the training set. These results verify that using $L2$-regularization indeed reduces the difference of prediction accuracy between the training and test set.

7.3.6 A Summary of Experiment Results

In §7.3, we go through every aspect in training and evaluating our linear regression model. Basically, we train a linear regression model, with squared-error loss function and $L1/L2$-regularizations, based on the training set, and predict the hourly average revenue given feature vectors in the test set. In short, our results show that:

- The linear model can achieve a MAE close to a non-linear model, with basic and composite features. In the meantime, the linear model have interpretable results.
- Using composite features can indeed significantly improve prediction accuracy. Also, using multi-source urban datasets also improves prediction results, and we can quantify the improvement with the introduction of different datasets.
- We combine any two basic features to form composite features, and do not combine more than two basic features. Our results show that combining three or more basic features does not lead to significant improvement on prediction accuracy while having a much longer training time.

8 Feature Contribution and Discussions

In §7 we mainly discuss the model and its evaluation. In this section, we take a different perspective, and dig into the learned model itself, trying to inspect feature contribution – *"what features are more important in determining hourly average revenue?"* – and seeking strategies – *"what seeking strategies lead to a higher revenue?"*. Besides, we also provide discussions on miscellaneous relevant topics.

8.1 Feature Contribution

As mentioned earlier, a very important reason of choosing a linear regression model is its interpretability – it allows a quantitative analysis of feature contribution, so that we can identify the seeking strategies leading to a higher revenue.

We study feature contribution by inspecting the weight ω learned in the model. Specifically, features, either basic or composite, may be one-dimension (e.g., *average delivery speed*) or multi-dimension (e.g., *timeslot-of-day* or *starting points' price multipliers*). For a multi-dimension feature, we inspect the weight of each component of this feature. The weight of a feature or a feature component quantifies the contribution of this feature or component. We rank features or feature components according to the absolute values of their weights, and in Tab. 5 we show some selected features (for one-dimension features) or feature components (for multi-dimension features) from the top-100 weights.

The interpretability of our model is justified by the distribution of weights. The largest weight shown in Tab. 5 is 8.84512. Among the absolute values of all the weights, statistics show that there are about 42% falling in between $[0.00, 0.05)$, 18% in $[0.05, 0.10)$, 10.5% in $[0.10, 0.15)$, 8% in $[0.15, 0.20)$, 6% in $[0.20, 0.25)$, etc. In other words, about 84.5% weights have an absolute value smaller than 0.25, a value far smaller than any one listed in Tab. 5 – this means only a small number of features are significant. Hence, the number of significant features, or those that are worth analysis, is much fewer than the number of non-zero weights. In our study, we only show top-100 weights due to the limited space, and these weights are already enough to give us enough findings and insights on the desired relationship.

In Tab. 5, we group these top features/feature components into 6 categories according to the key factor in each feature. *Taxi*, and *bus & metro* are two categories representing the respective datasets; and we divide the top features from RoD data into other four categories: *delivery speed*, *dynamic prices*, *timeslot*, and *strategy factor*. Because of the limited space, for each category we only list at most the top-3 basic features, and one or two composite features' components that fall in the top-100 among all. Below we discuss our findings regarding feature contribution.

From the level of datasets, the RoD service data has an overwhelming influence on the driver revenue: there are more than 90 features out of top-100 that are either from RoD service data, or combined from at least more than one basic features from RoD data. This figure becomes 4 and 5 for the bus & metro data and taxi data, respectively. Even with smaller impact, the status of taxi, bus and metro services all have a non-negligible influence on driver revenue.

In the following we discuss feature contribution from the level of individual features.

Delivery speed. Among all the features, *average delivery speed* has the highest impact on a driver's revenue. Its weight, being 8.84512, is much higher than other features or feature components. As a result, a number of composite features derived from *average delivery speed* also have higher weights, but the fact that *average delivery speed* itself has a much higher weight diminishes the importance of these composite features. We thus do not list these composite features in Tab. 5.

The importance of *average delivery speed* shows that driving faster or choosing faster or clearer routes in delivering passengers leads to a higher revenue. This is natural, as it saves more time for the driver to seek for more opportunities. This result also holds in the traditional taxi service, as suggested in [19].

Increasing the delivery speed is important to increase driver revenue, but how to operate so that the delivery speed is maximized is out of the scope of this study – it should be a job left to navigation systems or applications, including route planning, real-time traffic information analysis, avoiding congestion, etc.

Dynamic prices. This is the unique part of RoD service, and is also the second most influential category of features on driver revenue. For a seeking trip, the ending point (i.e., where to seek for passengers) is more important, but the starting point (i.e., where a driver drops the last passenger and starts seeking) also plays a role. This may be due to the fact that a significant proportion of seeking trips have close starting and ending points. In fact, our data shows that more than half seeking trips have a straight line distance between the starting and ending points smaller than 5km.

It is shown that seeking for passengers in regions with higher price multipliers increases driver revenue. More

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2921959, IEEE Transactions on Mobile Computing

TABLE 5
Selected top features/components ranked by weights in the trained model.

Category	**Feature (:feature component)**	**Weight**	**Rank**
delivery speed	average delivery speed	8.84512	1
dynamic prices	ending points' price multipliers: p_{max_now}	4.26815	22
	ending points' price multipliers: p_{max_next}	2.92965	33
	ending points' price multipliers: p_{max_last}	2.46655	42
	starting points' price multipliers: p_{max_now}	2.49441	41
	starting points' price multipliers: p_{max_last}	1.83793	56
	starting points' price multipliers: p_{max_next}	1.53844	79
timeslot	timeslot-of-day: 1	2.11808	51
	timeslot-of-day: 2	1.44048	85
	timeslot-of-day: 0, ending points' taxi status: full taxi ratio (daily)	1.40173	95
strategy factor	strategy factor: chasing future	2.06670	52
	strategy factor: no chasing	1.63245	73
	strategy factor: chasing current	1.42835	89
	strategy factor: chasing future, ending points' price multipliers: p_{max_now}	1.79426	59
bus & metro	ending points' bus & metro: the number of metro station	1.72366	65
	ending points' bus & metro: the number of bus station	1.39342	98
taxi	ending points' taxi status: full taxi ratio (daily)	1.42848	88
	ending points' taxi status: down count (daily)	1.40559	94

specifically, during one timeslot, the driver should always try to find some seeking locations with higher price multipliers so that the maximum price multipliers among all seeking trips' ending points gets larger. As to the weights, both the maximum price multipliers around the seeking locations during the hour of starting seeking and the next hour have higher weights, being 4.26815 and 2.92965.

Timeslot. The timeslot-of-day of seeking trips is also a key role in the profitability. We observe in §5.1.1 and Fig. 4 a counter-intuitive fact that the non-rush hours around noon are more profitable than the morning rush hours. The weights listed in Tab. 5 agree to previous observations. The weights of timeslot-1 (i.e., [10am, 4pm)) and timeslot-2 (i.e., [4pm, 10pm)) are higher, followed by timeslot-0 (i.e., [4am, 10am)). In other words, seeking for passengers during the non-rush hours around noon, as well as during the evening rush hours, helps the driver to earn more; seeking during morning rush hour is not as profitable as one may intuitively guess, and night hours are even less profitable.

This phenomenon can be understood by combining information from §5.1.1 and Tab. 5. During non-rush hours, the average delivery speed is faster, so that a driver may have more orders during the whole timeslots. Also, the lower price multipliers are compensated by the longer order distance during the non-rush hours. In morning rush hours, things are just the opposite: even though the price multipliers are high, the driving speed is much lower, and sometimes drivers need a very long time to find the next passenger after dropping the last one in busy regions.

For morning rush hours (i.e., timeslot-0), the relevant weight suggests that a driver should go to locations where the full taxi ratio is high, in order to earn more. This is also an interesting result, considering the concerns on the competition between RoD and taxi service. The reason may be that during morning rush hours the supply of cars is not enough to meet demand, so a high full taxi ratio signifies that the corresponding location is highly popular, thus having more unfulfilled demand. More potential demand brings a higher revenue to drivers.

Strategy factor. This is about chasing the price multipliers or not. Weights in Tab. 5 show that chasing higher price multipliers indeed has positive impacts on driver revenue, but the target of chasing should be the future price multipliers instead of the current ones. Specifically, "future" price multipliers means the prices in the next hour – and this requires a driver to have a clear picture or estimate about how the dynamic price multipliers may change over time, so that s/he can chase for a higher one. Another interesting result is that "no chasing" is better than "chasing current", which agrees with the observations from [25] and some blog articles [2] – they generally propose that "surge chasing" is not a good way to earn more money.

The profitability of "chasing future" strategy makes dynamic price prediction important. In [16], the authors have discussed dynamic price prediction in RoD service and pointed out that it is useful to improve passenger experience. Our results show that dynamic price prediction is also beneficial for drivers: if a third party is able to predict the variation of dynamic price multipliers, drivers can use the results in chasing for future higher prices. The methodologies to perform price prediction has been discussed in [16] and are not the scope of our study here.

Bus & metro. The influence of the status of bus and metro services on the driver revenue can be studied in our paper because of the introduction of multi-source urban data. Our regression results verify that the distribution (or the availability) of bus and metro services is also an important factor. According to the weights of top features shown in Tab. 5, drivers should go to locations with more metro or bus stations to look for passengers, and that metro station is much more important than bus station. "The number of metro stations" already has a higher weight, and its importance is further amplified considering the relatively smaller number of metro stations compared to bus stations.

Similar to previous discussions of the effects of the status of taxi service during morning rush hours, the number of bus or metro stations is a representation of a location's popularity: the more stations, the more potential RoD service demand. We hypothesize the reason behind this is that people may take a car service to home after leaving a metro or bus station, so providing this transit service becomes profitable for drivers. Similarly, such transit service can also happen when people take a car ride to work after leaving a metro or bus station.

Taxi service. In a similar fashion, we find that the status of taxi service is an indication of a location's popularity, rather than a reflection of the competition between taxi and RoD service. In seeking for passengers, drivers should choose locations with higher full taxi ratio, and with more passengers getting off taxis. These two indications actually reflect the popularity of locations, and thus drivers should go to these more popular locations.

Another interesting observation is that the status of taxi service is not as crucial as that of bus or metro service, according to the weights in Tab. 5. In other words, drivers should pay more attention to providing transit service to those passengers from public transportation services. It is true that both the status of taxi service and that of public transportation services are indicative of a location's popularity, but people from bus or metro may take a RoD ride then; comparatively, people from taxis may not take such a ride immediately.

Heuristics for drivers to earn more. Discussions above justify the following heuristics for drivers under dynamic pricing in a RoD service. Note that our work is not on recommending seeking routes step-by-step to drivers, so these heuristics are more a suggestion for drivers to keep in mind than a real-time guidance to choose directions and intersections. They are tenable as they are from real-data:

- Most importantly, try to increase the average delivery speed by choosing better routes or driving faster.
- Seek for passengers in regions with higher price multipliers; and try to increase the maximum of price multipliers among all seeking locations in one timeslot.
- Counter-intuitively, the morning rush hours is not the most profitable timeslot. Instead, seeking for passengers during the non-rush hours around noon is helpful to earn more. Evening rush hours is the second most profitable timeslot.
- During morning rush hours, go to regions with higher full taxi ratio – this means highly popular regions.
- Don't do "Surge chasing". Instead, try to get a prediction or estimate about the price multiplier in the next hour in neighboring regions, and chase for that.
- The status of taxi, bus and metro services are important signals in choosing seeking locations. Try to seek for passengers in locations with more metro stations, bus stations, higher full taxi ratio, and with more passengers getting off taxis. In particular, pay more attention to bus and metro services than taxi service.

8.2 Discussions

The influence of POI features. It is clear from Tab. 5 that *POI counts* are not influential as one may anticipate – they don't appear in top-100 weights. Characteristics of seeking locations definitely have impacts on driver revenue, and we hypothesize that there are two reasons for this phenomenon.

Firstly, the location information is also partly revealed by features from other datasets, though implicitly. For example, the number of full taxis around, the average speed of taxis, the number of passengers getting on/off taxis, the number of bus/metro stations all help to describe a picture about the supply, demand, traffic condition, location popularity that can characterize the location.

Secondly, the *POI counts* features we design may not be representative enough. We have pointed out in §4.4 that using the POI and its category that is nearest to a location is not enough to characterize a location, but our results suggest that our description may not be enough neither. An example around the airport terminal can clearly illustrate this. A passenger is standing in the airport terminal and requests for a ride. Clearly the "transportation facility" property is the reason why s/he is here. The POI counts, on the other hand, may not suggest this. The number of "transportation facility" POIs may be only one – the terminal; but there may be a number of shops, restaurants or hotels around, and the number may far exceed that of "transportation facility". In other words, the *POI counts* features emphasize the number of POIs, but in some cases POIs' "importance" is the key.

There are multiple solutions to describe a POI's "importance". For example, we can calculate the TF-IDF statistics of each POI category, so that the more common a category of POI is, the more its count gets diminished. In other words, a POI category that is more common turns out to be less important. Specifically, for the i-th POI category, we use p_i to denote the number of POIs around. For the whole city, we use N to denote the total number of POIs and use N_i to denote the total number of POIs of this category. Then, instead of using p_i as the POI counts, now we use $p_{tfidf,i} = p_i * log(\frac{N}{1+N_i})$ to involve the importance of this category of POIs. The replacement of POI features results in a MAE of 3.477 (very close to the original one), and features relevant to the TF-IDF features have the largest weight about 13 times than features related to *POI counts*. Another example is about obtaining new datasets. If we can obtain, say, the check-in data of a location-based service, we are able to claim that the more check-ins a location receives, the more important it is. We want to compare the effects of these solutions, and this task is left as future work when we obtain such check-in data.

Sources of inaccuracies. In §7.3 we show that our linear regression model can achieve an accuracy of 88.05% in predicting driver revenue. In the following we discuss sources of inaccuracies and possible ways to improve our model.

The first source is the lack of comprehensive urban data. For example, hourly weather data can help us study whether bad weather brings higher revenue to drivers; large scale check-in data can help in making POI information more effective; the distribution of buses and metro trains, possibly from smart-card data, instead of the distribution of bus and metro stations or lines, can help to describe the status of public transportation services more accurately. All these possible improvements require further research collaboration or new methodologies of data collection.

Another source of inaccuracies is in the estimation of dynamic price multipliers and trip fares. We approximate the price multiplier of one trip using the average price multipliers of all trip fare estimation events taking place around this trip in the same hour, losing some information about sudden unplanned price changes such as events or traffic accidents. The estimation of trip fare also omits waiting charge as it is hard to estimate precisely the total waiting time from GPS trajectories. The first problem may not be solved easily, as price information is always sensitive and the core secret in the service provider. The solution of the second problem requires techniques with finer granularity to estimate the waiting time during a trip.

The last source has been discussed in §5.1.2: when describing the starting/ending points of seeking trips in one timeslot, we use collective properties instead of describing each location. The reason is that we cannot use a feature vector of fixed dimension to accommodate a varying number of seeking trips, unless we reserve a space for each city cell, which is unrealistic because the number of cells is prohibitively large. Dividing a city into fewer cells makes the results artificial and not convincing enough. Our methodology is thus a compromise, and hence a possible solution is to consider this trade-off between artificiality and feature dimension and choose a reasonable division of city cells in trying to describe each of all seeking locations.

Generalizability of models and results. We claim that our study is generalizable and could be applied to other cities. It is true that the results (i.e., the quantitative relationship between driver revenue and seeking strategies and feature contributions) may differ across cities, as different cities have varying characteristics such as size, demographic patterns, distributions of functional areas, etc., leading to different profitable seeking strategies. But the models and methodologies (i.e., the linear regression model for learning and predicting, the methodologies to construct features and the way to analyze feature contribution) are not specific to any city, and could be generalized and applied to other cities, or to similar problems that require both accuracy and a certain level of interpretability. In other words, as long as one can collect similar datasets, s/he can perform feature extraction, model building, relationship mining and feature contribution analysis, in a similar way to our study, and obtain corresponding results, for, maybe, other cities.

Even for the results, we try our best to make them representative enough, so that it may not be too special to be considered in further studies. Real operational data from RoD service is still rare, considering our requirement that the information of dynamic prices and trips must be involved. We currently choose Beijing as the target, and it is one of the most representative metropolitan cities in China, East Asia or even around the world, in terms of size, demographic patterns, operational status of different transportation services, distribution of functional areas, etc. Also, as one of the major service cities, the RoD service provider has invested enough resources (e.g., car fleet management, drivers, algorithm design, etc.), which is the premise for drawing realistic, representative and tenable results.

Regarding the datasets, researchers interested in RoD service are also able to obtain similar datasets. It may be difficult to obtain datasets with perfect coverage, precision or accuracy, but there are possible ways to approximate such datasets. For example, synthetic data could be generalized by utilizing APIs released by service providers and combining results from multiple runs; crowdsourcing applications could be developed to encourage passengers of different RoD services to report their trips and experiences; etc.

As an extension, studying how the results may change across cities is one of our future work. It is thus possible to gain insights about different seeking strategies across different cities, and to understand how profitable seeking strategies are related to characteristics of cities. These topics would be studied as soon as we obtain the required datasets.

9 Conclusion

In this paper we focus on driver revenue under dynamic pricing in RoD services. Basically, we propose a system, RoD-Revenue, to mine the relationship between driver revenue and seeking strategies, and to predict driver revenue based on features extracted from multi-source urban data. We go through each level in the RoD-Revenue's framework, including the datasets, feature extraction, and model & prediction. As to the learning model in RoD-Revenue, we choose a linear regression model with high-dimensional composite features for its interpretability.

Our linear regression model has a feature dimension of $3,730$, and can predict driver revenue based on features relevant to seeking strategies at an accuracy of 88.05%. In our evaluation, we use correlation analysis to show the need to involve composite features, compare between linear and non-linear models, evaluate the effects of using composite features and discuss the effects of regularization terms.

Our findings suggest that the average delivery speed, the timeslot of seeking, the way of chasing price multipliers and the status of taxi, bus and metro services all have significant impacts on driver revenue in RoD service. Correspondingly, increasing delivery speed, seeking in non-rush hours, chasing future price multipliers, and seeking in locations with busier taxi and public transportation services all are tenable heuristics for drivers to earn more.

References

[1] W. Hudson, "Chasing the Surge: 3 Tips for Maximizing Uber Earnings," 2016. [Online]. Available: https://bit.ly/2NtrZKh

[2] H. Campbell, "Advice for New Uber Drivers – Don't Chase The Surge," 2016. [Online]. Available: https://bit.ly/2O1DdoM

[3] A. Picchi, "Uber vs. Taxi: Which Is Cheaper?" 2016. [Online]. Available: http://bit.ly/2DMgrMc

[4] V. Salnikov, R. Lambiotte, A. Noulas, and C. Mascolo, "Openstreetcab: exploiting taxi mobility patterns in new york city to reduce commuter costs," *arXiv preprint arXiv:1503.03021*, 2015.

[5] Y. M. Nie, "How can the taxi industry survive the tide of ridesourcing? evidence from shenzhen, china," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 242–256, 2017.

[6] S. Jiang, L. Chen, A. Mislove, and C. Wilson, "On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, ser. WWW'18, 2018, pp. 863–872.

[7] L. Rayle, D. Dai, N. Chan, R. Cervero, and S. Shaheen, "Just a better taxi? a survey-based comparison of taxis, transit, and ridesourcing services in san francisco," *Transport Policy*, vol. 45, pp. 168–178, 2016.

[8] J. D. Hall, C. Palsson, and J. Price, "Is Uber a substitute or complement for public transit?" 2017. [Online]. Available: https://bit.ly/2K6Vs7L

[9] T. Berger, C. Chen, and C. B. Frey, "Drivers of disruption? estimating the uber effect," *European Economic Review*, vol. 110, pp. 197–210, 2018.

[10] J. Hall, C. Kendrick, and C. Nosko, "The effects of Uber's surge pricing: a case study," Oct. 2015. [Online]. Available: http://bit.ly/2kayk9O

[11] J. Gan, B. An, H. Wang, X. Sun, and Z. Shi, "Optimal pricing for improving efficiency of taxi systems." in *Proceedings of the 22th International Joint Conferences on Artificial Intelligence*, ser. IJCAI '13. AAAI, 2013, pp. 2811–2818.

[12] L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero, "App-based, on-demand ride services: Comparing taxi and ridesourcing trips and user characteristics in San Francisco," 2014. [Online]. Available: http://bit.ly/2kVkahg

[13] L. Chen, A. Mislove, and C. Wilson, "Peeking beneath the hood of Uber," in *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, ser. IMC '15. New York, NY, USA: ACM, 2015, pp. 495–508.

[14] S. Guo, Y. Liu, K. Xu, and D. M. Chiu, "Understanding ride-on-demand service: Demand and dynamic pricing," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 509–514.

[15] S. Guo, C. Chen, Y. Liu, K. Xu, and D. M. Chiu, "Modelling passengers' reaction to dynamic prices in ride-on-demand services: A search for the best fare," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 136:1–136:23, 2018.

[16] S. Guo, C. Chen, J. Wang, Y. Liu, K. Xu, D. Zhang, and D. M. Chiu, "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.*, vol. 2, no. 3, pp. 112:1–112:24, 2018.

[17] M. K. Chen, "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*, ser. EC '16. New York, NY, USA: ACM, 2016, pp. 455–455.

[18] P. Cohen, R. Hahn, J. Hall, S. Levitt, and R. Metcalfe, "Using big data to estimate consumer surplus: The case of uber," 2016. [Online]. Available: http://bit.ly/2pqXiWo

[19] B. Li, D. Zhang, L. Sun, C. Chen, S. Li, G. Qi, and Q. Yang, "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, 2011, pp. 63–68.

[20] D. Zhang, L. Sun, B. Li, C. Chen, G. Pan, S. Li, and Z. Wu, "Understanding taxi service strategies from taxi gps traces," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 123–135, 2015.

[21] H. Rong, X. Zhou, C. Yang, Z. Shafiq, and A. Liu, "The rich and the poor: A markov decision process approach to optimizing taxi driver revenue efficiency," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16. ACM, 2016, pp. 2329–2334.

[22] Y. Gao, D. Jiang, and Y. Xu, "Optimize taxi driving strategies based on reinforcement learning," *International Journal of Geographical Information Science*, vol. 32, no. 8, pp. 1677–1696, 2018.

[23] C.-M. Tseng, S. C.-K. Chau, and X. Liu, "Improving viability of electric taxis by taxi service strategy optimization: A big data study of new york city," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–13, 2018.

[24] N. Garg and S. Ranu, "Route recommendations for idle taxi drivers: Find me the shortest route to a customer!" in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1425–1434.

[25] H. A. Chaudhari, J. W. Byers, and E. Terzi, "Putting data in the driver's seat: Optimizing earnings for on-demand ride-hailing," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 90–98.

[26] AMap, "API of AMap Service," 2017. [Online]. Available: http://bit.ly/2n8YRbZ

[27] Wikipedia, "Wikipedida: Feature scaling," 2018. [Online]. Available: https://bit.ly/2OEddNg

Suiming Guo received the Ph.D. degree from the Department of Information Engineering, the Chinese University of Hong Kong. Before that, he received the Bachelor and Master degree from Tsinghua University, Beijing, China. He is currently an Associate Professor in College of Information Science and Technology/College of Cyber Security in Jinan University, Guangzhou, China. His research interests include data mining, urban computing, pervasive computing and smart cities studies.

Chao Chen received the Ph.D. degree from UMPC (Paris 6) and Telecom SudParis. He is currently a full professor of computer science at Chongqing University, China. His research interests include pervasive computing, social network analysis, mobile crowdsensing, green logistics, and big data analytics for smart city applications.

Jingyuan Wang received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is currently an Associate Professor of School of Computer Science and Engineering, Beihang University, Beijing, China. His is also the leader of Beihang Interest Group on SmartCity (BIGSCity), and Vice Director of the Beijing City Lab (BCL). He published more than 20 papers on top journals and conferences, as well as named inventor on several granted US patents. His general area of research is data mining and machine learning, with special interests in smart cities.

Yaxiao Liu received his Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is currently a Sr. manager of solution architecture in AWS China. Mr. Liu is certified as a Distinguish/Lead architect by the Open Group. His research interests are in AI based cloud architecture, spatio-temporal big data, stream computing and smart cities.

Ke Xu received his Ph.D. from the Department of Computer Science of Tsinghua University. He is currently a full professor in the Department of Computer Science and Technology, Tsinghua University. He has published more than 100 technical papers and holds 20 patents in the research areas of next generation Internet, P2P systems, Internet of Things(IoT), network virtualization and optimization. He is a member of ACM. He has guest edited several special issues in IEEE and Springer Journals. He is serving as associate editor of IEEE IoT Journal.

Zhiwen Yu received his Ph.D. in computer science and technology from Northwestern Polytechnical University in 2005. He is a professor with Northwestern Polytechnical University, China. He has worked as a research fellow in Kyoto University, Japan from 2007 to 2009, and a post-doctoral researcher in Nagoya University in 2006-2007. He has been an Alexander von Humboldt fellow with Mannheim University, Germany from 2009 to 2010. His research interests include pervasive computing, context-aware systems, human-computer interaction, mobile social networks and personalization.

Daqing Zhang received his Ph.D. from University of Rome "La Sapienza" and University of L'Aquila, Italy in 1996. He is a full professor of Department of Networks and Mobile Multimedia Services, Institut Mines-Telecom/Telecom SudPais. His research interests include large-scale data mining, urban computing, context-aware computing, and ambient assistive living.

Dah Ming Chiu received his Ph.D. from Harvard University. He worked in industry for several hi-tech companies: Bell Labs, DEC and Sun Microsystems Labs. He returned to academia in 2002 to become a professor in the Department of Information Engineering at the Chinese University of Hong Kong. He served as department chairman from 2009 to 2015. Dah Ming is an IEEE Fellow since 2008. His recent research interests include Internet content distribution, data-driven modeling and analysis of large scale systems, and network economics.