Transportation Research Part C 76 (2017) 51–70

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach ☆

Xiqun (Michael) Chen *, Majid Zahiri, Shuaichao Zhang

College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China

ARTICLE INFO

Article history:
Received 30 June 2016
Received in revised form 12 December 2016
Accepted 30 December 2016
Available online 10 January 2017

Keywords:
On-demand ride service
Rideplitting
Ensemble learning
Boosting
Decision tree

ABSTRACT

In this paper, we present an ensemble learning approach for better understanding ridesplitting behavior of passengers of ridesourcing companies who provide prearranged and on-demand transportation services. An ensemble learning model is a weighted combination of multiple classification models or week classifiers to form a strong classification model. The goal of ensemble learning is to combine decisions or predictions of several base classifiers to improve prediction, generalizability, and robustness over a single classifier. This paper employs the Boosting ensemble by growing individual decision trees sequentially and then assembling these trees to produce a powerful classification model. To improve the prediction accuracy of ridesplitting choices, we explored real-world individual level data extracted from the on-demand ride service platform of DiDi in Hangzhou, China. Over one million trips of the four service types, i.e., Taxi Hailing Service, Express, Private Car Service, and Hitch, are explored with descriptive statistics. A variety of features that may impact ridesplitting behavior are ranked and selected by using the ReliefF algorithm, such as trip travel time, trip costs, trip length, waiting time fee, travel time reliability of origins/destinations and so on. The Boosting ensemble trees with full features and selected features are trained and validated using two independent datasets. This paper also verifies that ensemble learning is particularly useful and powerful in the ridesplitting analysis and outperforms three other widely used classifiers. This paper is one of the first quantitative studies that empirically reveal the real-world demand and supply pattern by exploring the city-wide data of an on-demand ride service platform.

1. Introduction

One of the most important information required for demand modelers and transportation practitioners is to find out passenger's urban travel behavior. In fact, realizing the trip origin and destination, travel time, trip monetary costs and other individual travel information in an area leads to better understanding of travel behavior and demand patterns. This issue is of significance for policy makers to improve comprehensive transportation planning, and formulate urban traffic congestion mitigation strategies.

Most recently, Shaheen et al. (2016) comprehensively reported the various modes of shared mobility that enabled users to obtain short-term access to transportation as needed, rather than requiring ownership, such as carsharing, personal

☆ This article belongs to the Virtual Special Issue on "Emerging Urban Mobility Services: Characterization, Modeling and Application".

* Corresponding author at: B828 Anzhong Building, College of Civil Engineering and Architecture, Zhejiang University, 866 Yuhangtang Rd, Hangzhou 310058, China.

E-mail address: chenxiqun@zju.edu.cn (X. (Michael) Chen).

http://dx.doi.org/10.1016/j.trc.2016.12.018
0968-090X/

vehicle sharing (i.e. P2P carsharing and fractional ownership), bikesharing, scooter sharing, ridesharing, and on-demand ride services. Ridesourcing or transportation network companies (TNCs) refer to an emerging urban mobility service that private car owners drive their own vehicles to provide for-hire rides. Zha et al. (2016) analyzed the ridesourcing market using an aggregate model and concluded that without any regulatory intervention a monopoly ridesourcing platform would maximize the joint profit with its drivers. Recent technologies enable on-demand ridesourcing services via smartphone applications, some of which enable passengers to choose to split a ride and fare in a ridesourcing vehicle, named ridesplitting (e.g., UberPOOL, Lyft Line, and DiDi Hitch). Shaheen et al. (2016) defined ridesplitting as "*a form of ridesourcing where riders with similar origins and destinations are matched to the same ridesourcing driver and vehicle in real time, and the ride and costs are split among users*".

This research analyzes the passenger ridesplitting behavior and travel patterns of an on-demand ride service platform using real-world data provided by DiDi in Hangzhou, China. In particular, understanding the emerging ridesplitting behavior needs a high quality and acquisitive dataset. Usually traditional manual methods of gaining knowledge and solving the problems are difficult and unreliable in case of complex engineering problems in real-world networks, thus advanced models and methods are of urgent necessity especially when there are abundant data. Nowadays, on the basis of exploring huge transportation data from roadways cameras, GPS, smartphones, traffic sensors and so on, traffic engineers can predict valuable traffic behavior and travel patterns in cities or solve the problems with innovative technologies, e.g., machine learning algorithms.

Fig. 1 shows the service model of TNCs, e.g., DiDi. The mobile connection and work flow between passengers and drivers are completed through the on-demand ride service platform. A TNC receives and analyzes requests of passengers and drivers according to program rules. For example, a passenger sends a request to the on-demand ride service platform for hailing a taxi or private car with the information such as the type of service, destination, departure time, number of passengers, car level and ridesplitting willingness. Then the TNC analyzes the request according to the real-time demand and supply nearby, car type, weather, holiday, responses of drivers and other passengers' requests. A matching procedure between the passenger and the most suitable driver nearby is implemented accordingly. The matched driver will receive the information about the passenger, price and suggested route. After finishing the ride, the passenger will pay a trip fee to the platform. The TNC will check out and charge commission at a certain rate. For example, DiDi charges zero commission for taxi drivers, while approximate 20% price paid by passengers of Express, Private Car Service and Hitch. The driver will then be paid by the platform at the remaining price. Different TNCs usually prepare a variety of promotion strategies for their customers according to their archived passenger information and real-time demand. For example, Leng et al. (2016) quantitatively analyzed the impact of promotion fees of two TNCs on the pattern of taxis services using 40-day trip data of over 9000 taxis in Beijing.

More research of ridesplitting is needed to better understand the behavioral characteristics on congestion, mobility, reliability and economy. In these years, solving transportation problems by machine learning has become more and more popular. Researchers can predict many travel activities from various perspectives. Many transportation problems can be

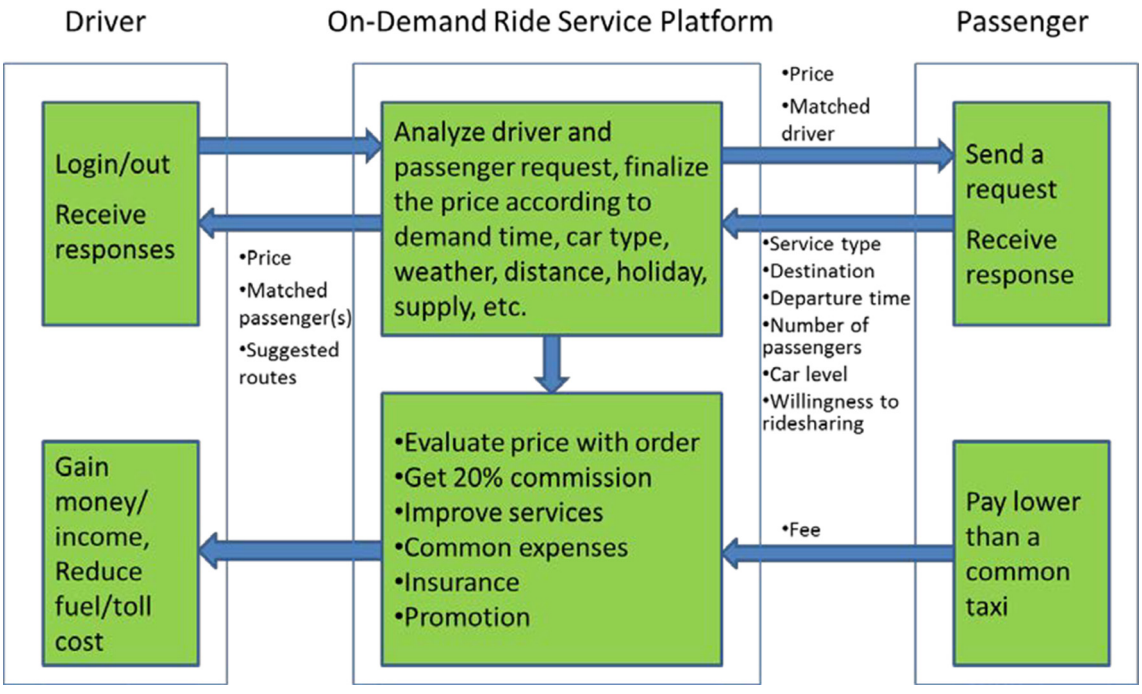

Fig. 1. Service model and work flowchart of an on-demand ride service platform.

modeled as a classification problem. To solve the problem, a decision tree classification model is represented by a tree-like structure, where each internal node represents a test of a feature, with each branch representing one of the possible test results and each leaf node represents a classification.

The improved classification performance achieved by using an ensemble of decision trees rather than a single tree for classification tasks is well established (Wolpert, 1992). The goal of ensemble learning is to combine decisions or predictions of several base classifiers to improve prediction, generalizability, and robustness over a single classifier. Some developed models include the stacked generalization (Wolpert, 1992), AdaBoost (Freund and Schapire, 1995), Bagging predictors (Brieman, 1996), random forests (Brieman, 2001), gradient tree Boosting (Friedman, 2001), and stochastic gradient Boosting (Friedman, 2002).

This paper employs the Boosting ensemble learning by growing individual decision trees sequentially and then assembling these trees to produce a powerful ensemble. In general, ensemble weak classifiers outperform an individual strong classifier. Ensemble learning has been successfully used in a variety of applications. For example, Alam et al. (2009) proposed an ensemble approach for conflict detection in free flight. Li et al. (2014) presented the application of multimodel ensemble techniques in traffic state estimation problems, and demonstrated the ensemble learning framework was capable of handling uncertainties in model initial conditions, model parameters, and model structures. Jiang et al. (2014) proposed a hybrid short-term high speed rail passenger flow forecasting approach by combining the ensemble empirical mode decomposition and grey support vector machine.

In this paper, we present an ensemble learning approach for better understanding ridesplitting behavior and improving the prediction accuracy of ridesplitting choices. It is assumed that none of individual models are perfect. Such that the key idea is to ensemble weak classifiers to formulate a strong classifier, which can make more accurate estimations and predictions. To improve the classification accuracy for ridesplitting choices, we develop the ensemble learning framework that assembles information gotten from every individual model to improve ridesplitting prediction accuracy. To the best knowledge of the authors, this paper is one of the first attempts to apply ensemble learning to the ridesplitting analysis based on real-world city-wide on-demand ridesourcing data.

The rest of this paper is organized as follows. Section 2 first reviews the existing research in shared mobility and its effects from a variety of perspectives. To presents descriptive statistics of the emerging service type of smartphone based on-demand travels, Section 3 performs an exploration of real-world data from the DiDi on-demand ride service platform in Hangzhou, China. Section 4 discusses how to implement the ensemble learning framework for the ridesplitting behavior analysis. Section 5 provides some numerical test results to evaluate the benefit of the ensemble learning approach. We use statistical learning to specify the feature selection and learning parameter tuning, then compare the ensemble learning with other well-defined classification algorithms. Finally, Section 6 concludes this paper and outlooks the future research.

2. Literature review

Shared mobility, the shared use of a vehicle, bicycle, or other mode, is an innovative transportation strategy that enables users to gain short-term access to transportation modes on an as-needed basis (Shaheen et al., 2016). The term shared mobility includes various forms of carsharing, bikesharing, ridesharing (carpooling and vanpooling), and on-demand ride services. Ridesharing is a term used to describe grouping travelers into common origin and/or destination by a car or van, and has become a powerful strategy to reduce congestion. Chan and Shaheen (2012) categorized North American ridesharing into five key phases, i.e., World War II car-sharing clubs (1942–1945), major responses to energy crises (late 1960s to 1980), early organized ridesharing schemes (1980–1997), reliable ridesharing systems (1999–2004), and technology-enabled ridematching (2004 to present).

In the most recent years, the widespreading applications of mobile internet, smartphones, and cloud computing stimulate online on-demand ride software platforms. Ridesourcing which is introduced to people after releasing smartphones provides is an efficient transportation service type. There are some useful smartphone applications that enable people to manage their individual car use better for instance by drivers share real-time traffic information (Waze Mobile, 2014). This new generation of ridematching platforms has gained a massive rise in ridesharing drivers and passengers. It is a new kind of point-to-point transportation network service based on a dynamic platform on which drivers offer their own cars with matched passengers who are looking for a similar destination. He and Shen (2015) proposed a spatial equilibrium model that balances the supply and demand of taxi services in a regulated taxi market with the smartphone-based e-hailing application.

Ridesourcing or on-demand ride services are different from traditional and common taxi services which the payments are only permitted by the platform. Some examples of on-demand ride service platforms include Uber, Lyft and DiDi. It is a good opportunity to use empty seats of cars which are available. Rayle et al. (2016) conducted a survey of ridesourcing users in San Francisco and found that ridesourcing differed from taxis in important ways, especially inconsistently shorter waiting times. In that research, it was found that at least half of ridesourcing trips replaced a mode other than taxi, indicating the two services have overlapping but different markets. Besides, ridesourcing competed with public transit for some individual trips, but it might sometimes serve as a complement.

Actually full potential of ridesharing is not obvious because it depends on using social networks and new technologies which both are critical for the future of it, or financial incentives and enhanced casual carpooling (Chan and Shaheen, 2012). Atasoy et al. (2015) proposed the Flexible Mobility on Demand (FMOD) system that provided flexibility to both pas-

sengers (with different levels of service) and transportation operators (dynamic allocation of vehicles to different services) by utilizing utilizes three services: taxi, shared-taxi and mini-bus. An agent based model was presented for solving a dynamic ridesharing problem that allowed single or multiple drivers to be matched with single or multiple passengers (Nourinejad and Roorda, 2016). These years some companies work on developing self-driving cars and we will probably meet Autonomous Vehicles (AVs) in the near future. These driverless cars emerge new human travel behavior (Fagnant and Kockelman, 2014). AVs will facilitate ridesharing behavior, as the technology can overcome some key barriers, especially the limited accessibility and reliability of today's carsharing and ridesharing programs.

Sharing economy involves of principle of sharing services, goods, knowledge and experiences and collaborates to gain the best result of sharing (Teubner, 2014). An idea which causes ownership finds more benefit and consumer can get more benefit, too. The increasing price of fuel and insurance, environmental and traffic jam problems are some reasons for tending and orient to ridesharing (Gargiulo et al., 2015). People these days use the private cars more wisely by their smartphones (Speed and Shingleton, 2012). Car sharing led to less expenses, reduction of overall vehicle miles and purchasing the car will be reduced, consequently (Le Vine et al., 2014). The benefit by obtaining personal auto mobility without the need to own a private vehicle; this can result in considerable monetary saving. For example, evidence from the North American carsharing member survey showed that carsharing facilitated a substantial reduction in household vehicle holdings (Martin et al., 2010), and a statistically significant overall decline in public transit use (Martin and Shaheen, 2011b). De Oliveira et al. (2013) pointed out that between 1990 and 2010, 754 million urban inhabitants emerged in Asia, more than the total population of the US and Western Europe combined. This rapid urbanization and motorization will bring more potential to enhance ridesharing economy. In a recent research, the diverse range of impacts of car2go on vehicle ownership, modal shift, vehicle miles traveled, and greenhouse gas emissions were investigated and analyzed in 5 North America cities (Martin and Shaheen, 2016).

When vehicles have higher occupancies, they can help our environment better. Martin and Shaheen (2011a) found that carsharing had a statistically significant impact in reducing annual household greenhouse gas emissions. On the contrary, the private car usage creates negative externalities. For example, the annual cost of congestion in the US in terms of lost hours and wasted fuel was estimated to be $78 billion in 2007 (Schrank and Lomax, 2007). Air pollution in Beijing worsened as the number of motor vehicles in the city increased from 2 million in 2004 to 4.8 million in 2010 (Chen and Zhao, 2013). Sharing economy has a good potential for reducing eco-footprints. In fact there is an enormous amount of new economic value being created in this field (Schor, 2014). Shared taxis have a good potential to make benefits, including increased efficiency, lower costs for passengers, and reduced congestion and overall vehicle travel (Santi et al., 2014). As the number of residents or jobs goes up in an improving economy, or the miles or trips that those people make increases, the road and transit systems also need to, in some combination, either expand or operate more efficiently (Schrank et al., 2015). Carsharing also has social benefits (Cervero et al., 2007). Carsharing makes more alternative to travel modes, such as public transit, biking, and walking. It is clear that this lifestyle improve health also causes to decreased traffic, congestion, and parking demand in urban areas.

To better understand ridesplitting behavior in the big data era, this paper aims to employ advanced machine learning models to analyze the data provided by on-demand ride service platforms. In the literature, machine learning algorithms such tree based ensemble methods are powerful in problems of classification and prediction. After the emergence of smartphone-based mobile sensing, gaining big data of a huge size in real time becomes easier. Advanced machine learning methods have been applied to solve different transportation problems. For example, Zhang and Haghani (2015) employed a gradient boosting regression tree to analyze and model freeway travel time to improve the prediction accuracy and model interpretability. Fei et al. (2011) presented a Bayesian inference-based dynamic linear model to predict online short-term travel time on a freeway stretch. Min and Wynter (2011) developed a highly accurate and scalable method for traffic prediction at a fine granularity and over multiple time periods. Shi and Abdel-Aty, 2015 used the microwave sensors for a proactive real-time traffic monitoring strategy to evaluate operation and safety simultaneously. Li et al. (2015) built a model for traffic flow prediction based on the massive data collected by different sensors in cities.

3. On-Demand ride service platform data

DiDi was founded in 2012 as a taxi-hailing app and later developed private-car hailing business. Currently, DiDi is one of the largest on-demand ride service platforms in the world (Shih, 2015). In China, it offers taxi hailing services in 380 cities for 1.6 million registered taxi drivers, peer-to-peer private car services in over 400 cities, and hitch services in more than 300 cities in China (DiDi, 2016). The rides completed in 2015 reached 1.43 billion, or over 11 million rides completed per day. By April 2016, there were over 300 million registered passengers and over 14 million drivers on the on-demand ride service platform. Currently, there are four service types on the DiDi platform, i.e., Taxi Hailing Service, Express, Private Car Service, and Hitch. The taxi-hailing platform of DiDi serves 3 million rides per day and with 99% mobile hailing market share in China (Analysys International, 2016). DiDi Express provides economical rides which are generally more cost effective than DiDi Taxi or traditional taxi services. DiDi Private Car Service provides more comfortable and high-quality ride services that can be divided into DiDi Premium and DiDi ACE. DiDi Premium offers comfortable and professional services, while DiDi ACE is a luxury and customized service, provided by chauffeurs with luxury cars. The peer-to-peer private car services have also reached 3 million rides per day and accounted for 86% of the market, according to Analysys International (2016). The most recently launched type of service in June 2015 is DiDi Hitch, which matches drivers and passengers who share similar routes.

The datasets used in this paper were extracted from the on-demand ride service platform of DiDi between September 7 and 13, 2015, and between November 1 and 30, 2015 in Hangzhou, China. The total successful orders or completed trips during the two time periods were 251344 and 1678289, respectively, which were randomly sampled at the rate of 20% from the on-demand ride service platform of DiDi. An individual trip record includes the pickup and drop-off location and time, order ID, passenger ID, passenger order time, driver response time, trip distance, type of service, car level, trip cost, and ridesplitting (1 yes; 0 no). The personally identifiable information of drivers and passengers has been properly anonymized to avoid any privacy issues. The information of longitude and latitude locations of each trip is obtained by converting the received GPS data into a planar coordinate system.

The spatial distribution of request origins is shown in Fig. 2. The patterns of different services indicate that DiDi Taxi and Express have denser passenger travel requests than Hitch and Private Car Service. Although Taxi and Express Services are more prevalent in downtown areas, the abundance of Hitch and Private Car Services are relatively low.

Fig. 3 presents the ridesplitting trips provided by Hitch in a temporal manner. Note that ridesplitting trips can be completed by the service types of Hitch, Express, and Private Car Service. All of the Hitch trips are ridesplitting, while Express and Private Car Service enable passengers to make choices whether to split rides with others. If the passenger(s) is willing to split rides and are successfully matched with other passenger(s), the ridesplitting trip is identified. There will be a fare discount for ridesplitting passengers. App-enabled ridesplitting is also allowed by other TNCs. Options like UberPOOL and Lyft Line allow unrelated passengers whose routes overlap to split rides and fares (Rayle et al., 2016). DiDi Hitch is different from Taxi, Express, and Private Car Service in terms of travel distance, travel fee, waiting time, travel time, etc. More percentages of Hitch requests are made primarily in AM/PM rush hours.

Splitting drivers in terms of service types, Fig. 4 shows the number of drivers for Taxi, Express/Private Car Service and Hitch with respect to time of day and day of week. Hitch is more sensitive to the off-peak hours and weekends when available drivers are significantly fewer than peak hours and weekdays. The temporal distributions of drivers of different service types are also in agreement with the pattern observed in Fig. 3.

Fig. 5 shows that Hitch, Express and Private Car Services have a similar pattern especially in AM/PM rush hours, however, DiDi Taxi only has one peak in the morning. The spreading PM peak of DiDi Taxi orders received by the on-demand ride service platform may be because hailing taxi drivers can also easily pick up road-side waiting passengers without using the DiDi app during the PM peak hours.

In this paper, the waiting time of passengers is defined as the period between they send requests and get on the car. Fig. 6 depicts that the distribution of the passenger waiting time of Hitch is normally the largest. It sometimes takes longer than 60 min because many passengers prefer to schedule a pickup time in advance, while passengers who choose other three service types generally prefer to depart immediately after sending requests. The mean waiting time for Express/Private Car Service is 6.13 min, while its standard deviation is 3.96 min. In addition, the mean waiting time for Express/Private Car Service during weekdays is 6.10 min, while its standard deviation is 3.94 min. The distribution of passenger waiting time is possibly useful for the on-demand ride service platform to measure the level of service and determine the best fare for drivers and passengers.

Fig. 7 shows the travel time distribution of different service types. The travel time of Hitch is obviously longer than other services due to its larger trip distance and more concentration in both peak hours. The overall statistics for Taxi, Express, and Private Car Service are: peak-hour (5–7 PM) mean travel time is 18.72 min, and peak-hour standard deviation of travel time

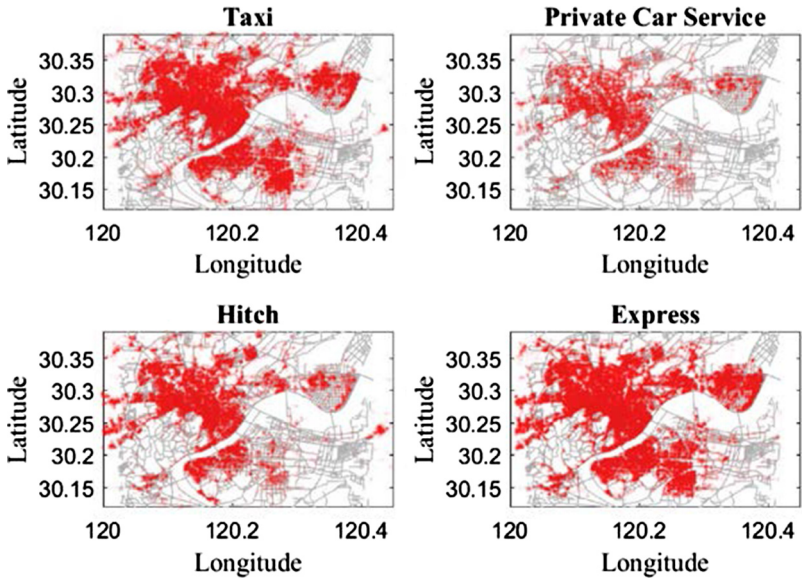

Fig. 2. Origin distribution of passenger travel requests in terms of the service type.

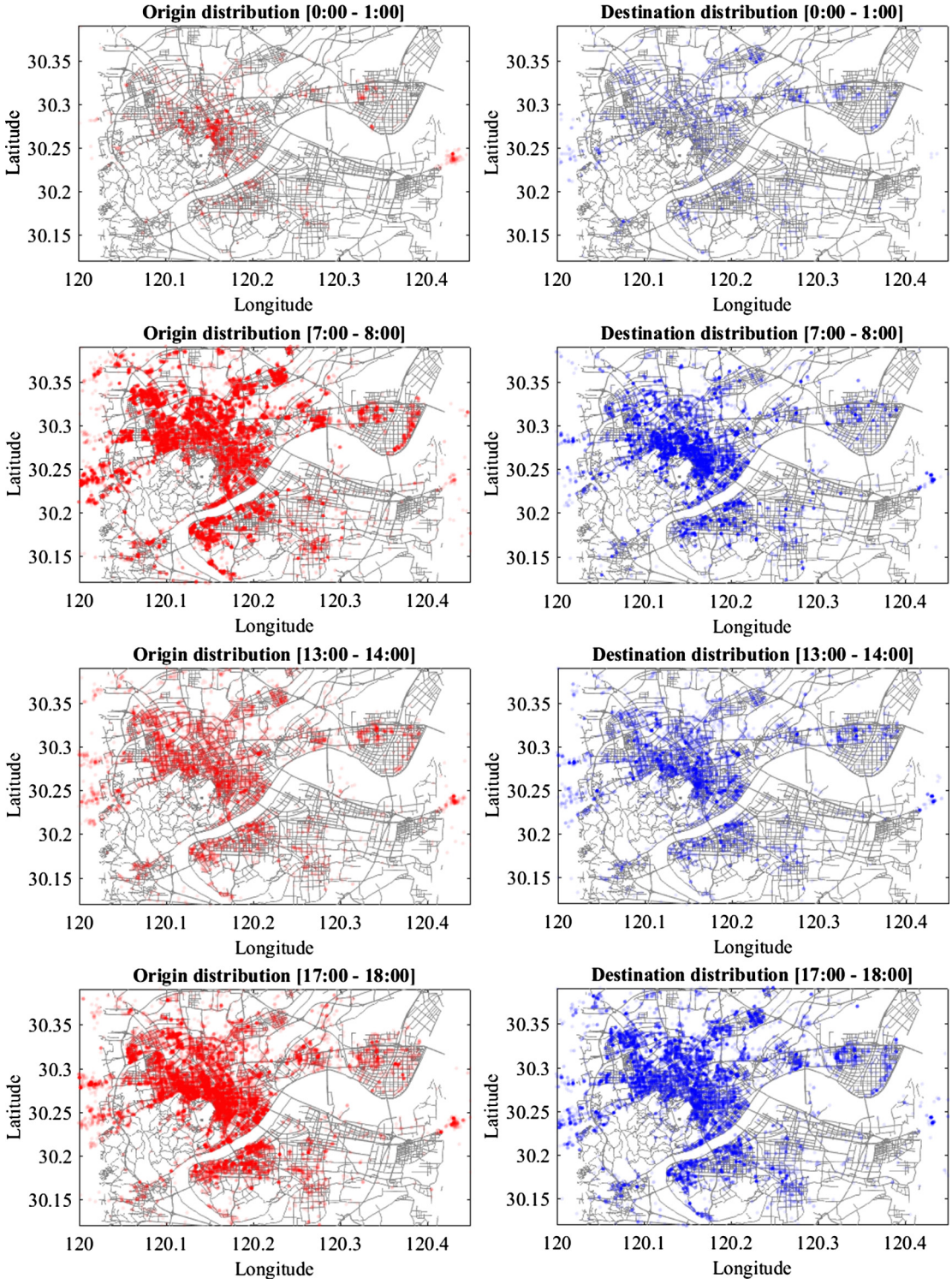

Fig. 3. Origins and destinations of ridesplitting trips.

is 12.83 min; non-peak-hour (1–4 PM) mean travel time is 18.06 min, and non-peak-hour standard deviation of travel time is 12.78 min.

Since a lot of taxi passengers stop using the platform during their trips, and they can choose to pay cash or use the platform to complete their trips, records that contain taxi fees in our datasets are very limited. So we remove the column of the taxi price from the dataset for the actual price analysis. Fig. 8 displays the distributions of trip distance, travel fee and travel

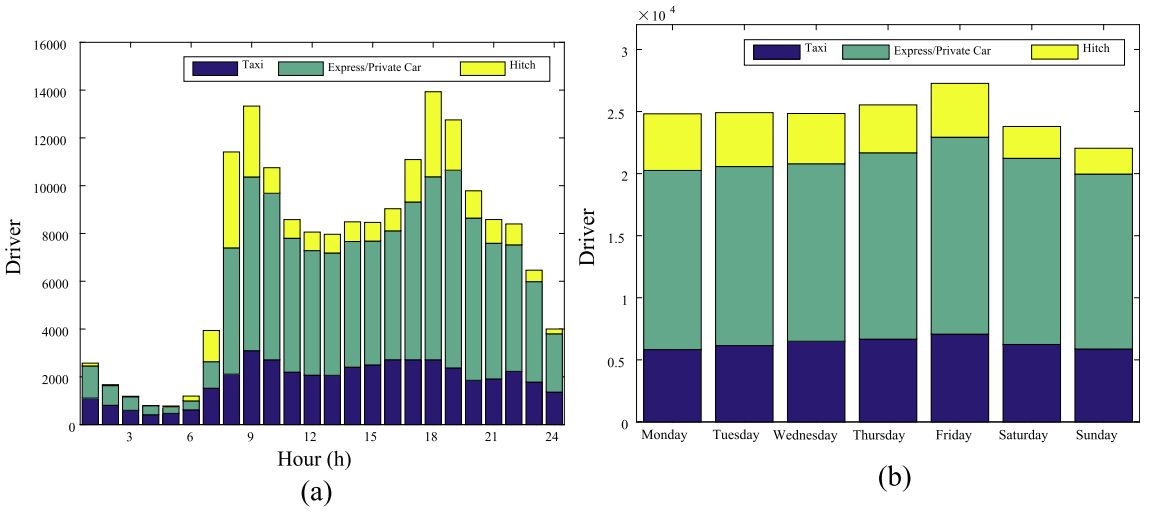

Fig. 4. (a) Hourly and (b) daily distributions of drivers with respect to different service types.

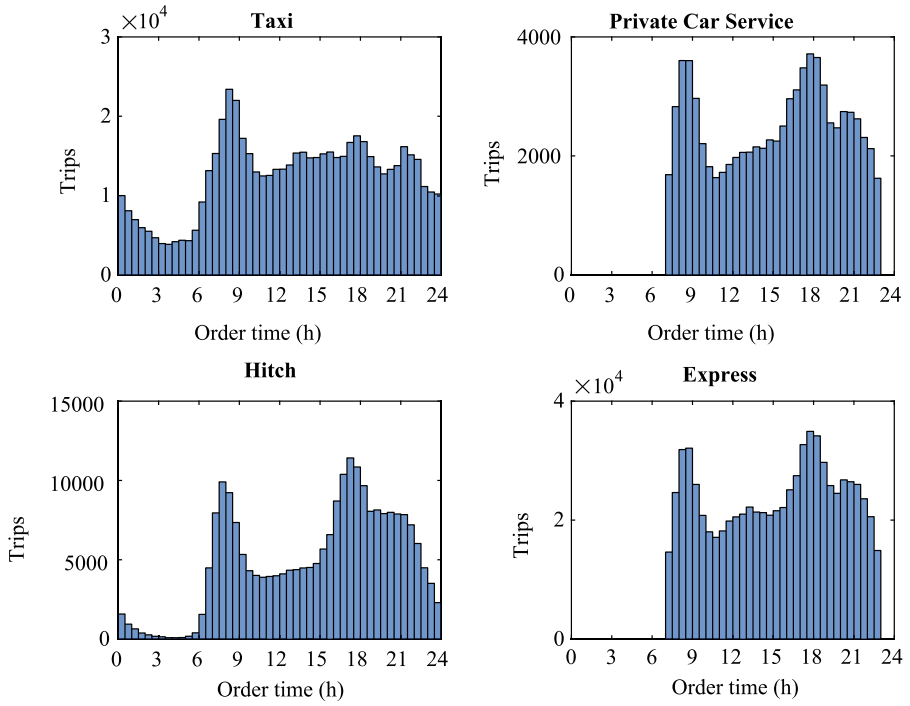

Fig. 5. Order time distribution in terms of the service type.

time rate (i.e., travel time per unit distance) in terms of other three service types. The average trip distance of Hitch is more than twice of the Private Car Service and Express. For the latter two service types, the peak-hour (5–7 PM) mean travel distance is 6.32 km, while the non-peak-hour (1–4 PM) mean travel distance is 6.64 km. The mean unit trip cost in travel distance during weekdays is 3.09 RMB/km, and the standard deviation is 1.46 RMB/km. The mean unit trip cost in travel time during weekdays is 1.04 RMB/min, and the standard deviation is 0.50 RMB/min. Fig. 8 also shows that Hitch travel fee doesn't increase as highly as the travel distance. In fact, it shows that passengers who travel long distances usually use the Hitch service because it is more affordable. Trip distance/fee distributions of Express and Private Car Service are approximately the same. Travel time rates of three service types perform similarly, too.

In this Section, descriptive statistics are shown to understand the temporal and spatial distributions such as passenger requests, waiting time, travel time, trip distance, trip fee, and travel time rate. These intuitive results provide with valuable information on the characteristics of different service types. To the best knowledge of the authors, these results may be one

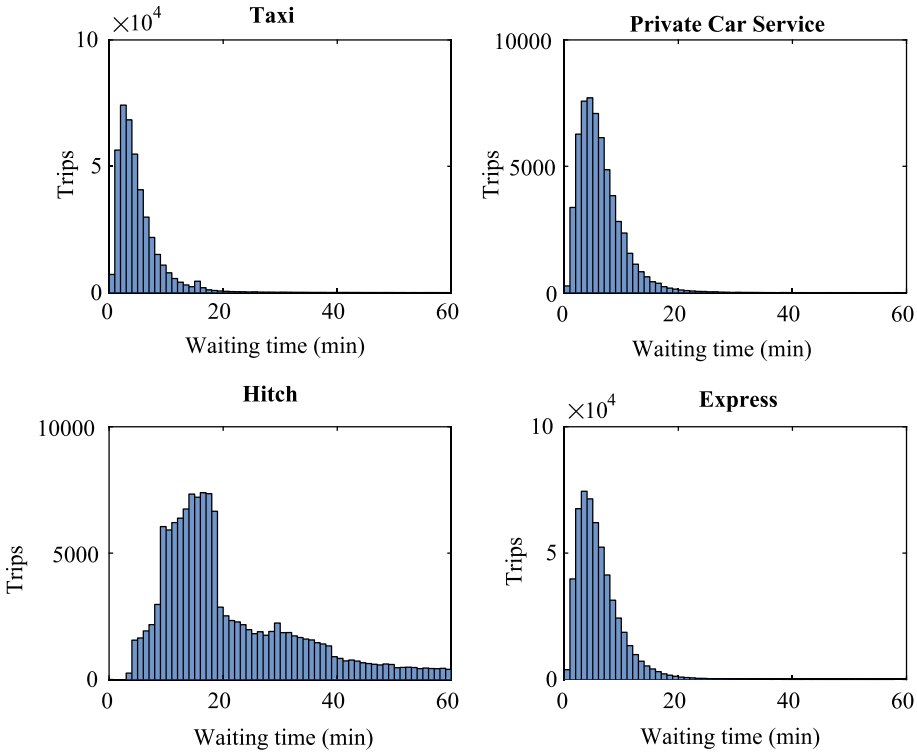

Fig. 6. Passenger waiting time distribution in terms of the service type.

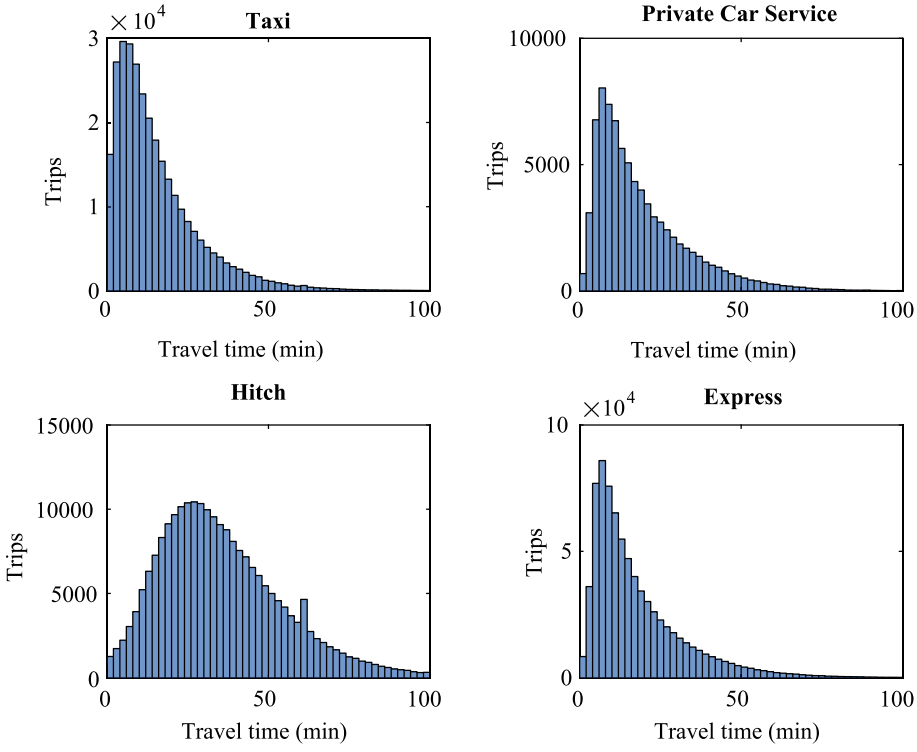

Fig. 7. Trip travel time distribution in terms of the service type.

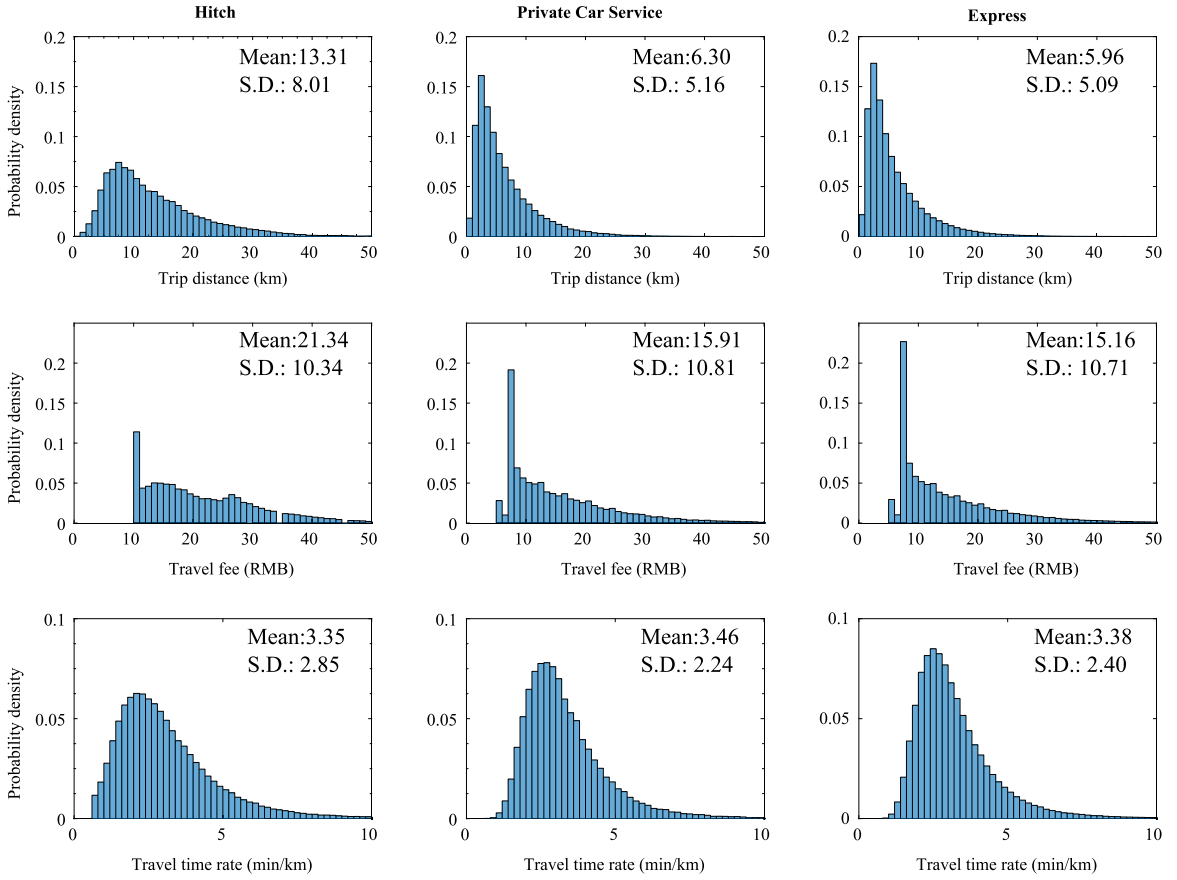

Fig. 8. Distributions of trip distance, travel fee and travel time rate in terms of the service type.

of the first quantitative studies to reveal the real-world demand and supply pattern by exploring the city-wide data of an on-demand ride service platform. To better understand the ridesplitting behavior on the basis of aforementioned observations, Section 4 will present an ensemble learning approach.

4. Ensemble learning model for ridesplitting behavior

4.1. Ensemble trees for ridesplitting analysis

A classification ensemble learning model is a weighted combination of multiple classification models or week classifiers. In general, ensemble learning models increase the predictive performance.

As shown in Fig. 9, ensemble trees combine multiple base/weak classifiers for combined/strong classification. The base classifier can be any kind of supervised classification algorithms such as decision trees, support vector machines (SVMs), or neural networks. There are three main categories of ensemble learning algorithms:

(1) Bagging or Bootstrap aggregating algorithm (Brieman, 1996). The training set of each base classifier is sampled by randomly selecting a subset with replacement. The bagging classification is based on the majority voting scheme;
(2) Boosting algorithm (Freund and Schapire, 1995; Friedman, 2001). Classifiers are created sequentially where the next base classifier assigns more weights to the mistakes that the previous classifier made;
(3) Stacking algorithm or stacked generalization (Wolpert, 1992). Outputs of base classifiers are inputs for the second level of classifiers. Stacking is used to learn how the base classifier make errors, and the second level classifier tries to overcome the errors.

Decision tree classifiers have many advantages such as a hierarchy of simple decisions with each decision based on only one or several features, light computing burden, non-parametric and no a priori requirement. However, individual decision trees are considered as weak classifiers in this paper. To overcome unstable, overfitting, inaccurate problems that face individual decision trees, this paper presents a benchmark ensemble algorithm for Boosting the accuracy of decision trees, i.e., AdaBoost, which is one of the most popular Boosting algorithms.

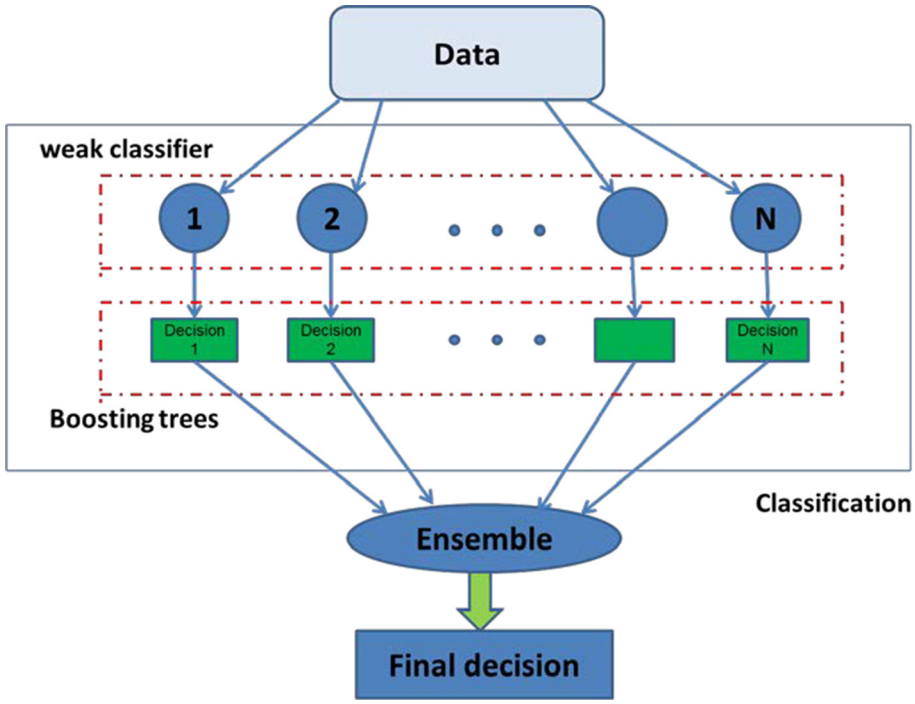

Fig. 9. Framework of the ensemble learning.

Initially, AdaBoost starts by building the first base classifier, which is trained on the dataset with equal weights. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. The predictions from a sequence of weak classifiers are combined through a weighted majority vote (or sum) to produce the final prediction. At a given step, those training samples that were incorrectly predicted by the Boosting model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. Then the weights of all instances in the whole dataset are then normalized and used for sampling for the next classifier. The final classification is based on weighted base classifiers.

Mathematically, let $\{h_t(\mathbf{x}) : t = 1, \ldots, T\}$ be the prediction of T classifiers (hypotheses) with index t defined on an input vector $\mathbf{x}$ and let $[c_1, \ldots, c_T]$ be their weights satisfying $c_t \geqslant 0$ and $\sum_{t=1}^{T} c_t = 1$. In the training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_n$ is the vector of predictor values for observation n, y_n is the true class label. $w_t(\mathbf{x}_i, y_i)$ is the weight of observation n at step t. In this paper, we consider the binary ridesplitting choice behavior, i.e., $h_t : \mathbf{x} \mapsto \{\pm 1\}$, for all $t = 1, \ldots, T$. The AdaBoost algorithm increases weights for observations misclassified by classifier t and reduces weights for observations correctly classified by t. The next classifier $t + 1$ is then trained on the data with updated weights $\mathbf{w}_{t+1}$.

The main parameters to tune of the AdaBoost algorithm include: the number of weak classifiers, the learning rate that controls the contribution of the weak classifiers in the final combination, and the complexity of the base decision tree (e.g., depth, minimum required number of samples at a leaf). The aforementioned parameter tuning results will be presented in Section 5.

AdaBoost algorithm for ensemble classification trees

Input: Training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, a vector of feature values and the class values

Initialization: Let initial weights $w_1(\mathbf{x}_i, y_i) = 1/n$ for all $i = 1, \ldots, n$

for every classifier with index $t = 1$ to T do

Train classifier with respect to the weighted sample set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n), \mathbf{w}_t\}$ and obtain the prediction of classifier with index t, i.e. hypothesis $h_t : \mathbf{x} \mapsto \{\pm 1\}$;

Compute the weighted classification error $\varepsilon_t = \sum_{i=1}^{n} w_t(\mathbf{x}_i, y_i) \mathrm{I}(h_t(\mathbf{x}_i) \neq y_i)$, where I is the indicator function;

Set $b_t = \log(1/\varepsilon_t - 1)$;

Update the weights $\mathbf{w}_t$ as $w_{t+1}(\mathbf{x}_i, y_i) = w_t(\mathbf{x}_i, y_i) exp\{-b_t \mathrm{I}(h_t(\mathbf{x}_i) = y_i)\}$;

Normalize the weights $\mathbf{w}_{t+1}$ as $w_{t+1}(\mathbf{x}_i, y_i) = w_{t+1}(\mathbf{x}_i, y_i) / \sum_{i=1}^{n} w_{t+1}(\mathbf{x}_i, y_i)$

end for

return prediction for new data $\mathbf{x}$ using $f(\mathbf{x}) = \sum_{t=1}^{T} c_t h_t(\mathbf{x})$, where weights of the weak hypotheses in the ensemble are computed as $c_t = b_t / \sum_{t=1}^{T} |b_t|$.

4.2. Network mobility and reliability

Despite the abundance of research on reliability optimal flow configurations and their implications in the transportation network mobility and reliability literatures, there is a shortage of works that use big data sources (e.g., on-demand ride service platform established by DiDi) to understand the role of traffic mobility and reliability in metropolitan regions that may affect on ridesplitting behavior. This highlights a need to build a model that can use the on-demand ride service data described in Section 3 and can be replicated to systematically generate meaningful origin-destination (OD) level performance measures of effectiveness (MOEs), e.g., travel time reliability indicators.

In this paper, empirically observed travel time data from DiDi are used to estimate the OD level travel time reliability measures, which will be used as influencing features of ridesplitting behavior to develop ensemble trees in Section 5.1.

The travel time rate represents travel time per unit distance. Richardson and Taylor (1978) first investigated the use of unit travel time to indicate the relationship between congestion and the variability of travel times. Travel time rate helps exclude the source of variability coming from trip distance and focuses on the travel time variability caused by the variation of speed (Mahmassani et al., 2013).

Parameter notation	
d_{ijk}	Travel distance of trip k from origin i to destination j
$f_{ij}(\tau)$	Probability density function of τ for the OD pair (i,j)
$F_{ij}(\tau)$	Cumulative distribution function of τ for the OD pair (i,j)
n_{ij}	Number of trips from origin i to destination j
$NBTR$	Network buffer time rate
$NBTRI$	Network buffer time rate index
$NFFTR$	Network free-flow travel time rate
$NPTR$	Network planning time rate
$NTTR$	Network travel time rate
t_{ijk}	Travel time of trip k from origin i to destination j
w_{ij}	Weight of the OD pair (i,j)
β_{ij}	Buffer time rate of the OD pair (i,j)
β'_{ij}	Buffer time rate index of the OD pair (i,j)
τ	Random variable of travel time rate
τ_{ijk}	Travel time rate of trip k from origin i to destination j
$\bar{\tau}_{ij}$	Mean travel time rate of the OD pair (i,j)
$\tau_{ij,\alpha}$	Travel time rate percentile at the confidence level of α for the OD pair (i,j)

The application of travel time rate can be widely extended to travel time reliability related research. The definition for the trip level travel time rate is given by

$$\tau_{ijk} = \frac{t_{ijk}}{d_{ijk}}, i \in I,\ j \in J,\ 1 \leqslant k \leqslant n_{ij} \tag{1}$$

The mean travel time rate on the OD pair level, i.e., $(i,j),\ i \in I,\ j \in J$, is calculated as

$$\bar{\tau}_{ij} = \frac{1}{n_{ij}} \sum_{k} \tau_{ijk},\ \tau_{ijk} > 0 \tag{2}$$

For the same OD pair, the empirical probability density function of τ, i.e. $f_{ij}(\tau)$, can be estimated based on observations τ_{ijk}, $1 \leqslant k \leqslant n_{ij}$. Then the percentile of travel time rate at the confidence level of α is calculated as the inverse function of the cumulative distribution function of τ, i.e., $F_{ij}(\tau)$, given by

$$\tau_{ij,\alpha} = F_{ij}^{-1}(\alpha),\ 0 \leqslant \alpha \leqslant 100\% \tag{3}$$

The buffer MOEs quantify the extra percentage of travel time due to travel time variability on a trip that a traveler should take into account in order to arrive on time. Buffer time rate is defined as the extra travel time rate a traveler plans to add to the median travel time rate for arrival on time with the confidence level of 95%. Buffer time rate index is defined as the ratio between the buffer time rate and the mean travel time rate. Both MOEs are useful in the assessment for uncertainty in travel conditions. Hence, their definitions are given by

$$\beta_{ij} = \tau_{ij,95\%} - \tau_{ij,50\%},\ \beta'_{ij} = \beta_{ij}/\tau_{ij,50\%} \tag{4}$$

Based on $\tau_{ij,\alpha}$, five network travel time reliability quantities are used in this study. The mathematical expressions of these quantities are shown in Eqs. (5–9).

The network free-flow travel time rate (NFFTR), which is distance-weighted, is calculated by

$$NFFTR = \frac{\sum_i \sum_j w_{ij} \cdot \tau_{ij,5\%}}{\sum_i \sum_j w_{ij}} \tag{5}$$

where $w_{ij} = \sum_k d_{ijk}$ represents the total observed travel distance generated by n_{ij} trips for the OD pair (i,j), and $\tau_{ij,5\%}$ denotes the free-flow travel time rate of (i,j).

The network travel time rate (NTTR) is defined as

$$NTTR = \frac{\sum_i \sum_j w_{ij} \cdot \tau_{ij,50\%}}{\sum_i \sum_j w_{ij}} \tag{6}$$

where $\tau_{ij,50\%}$ represents the median travel time rate out of n_{ij} trips of (i,j).

The network planning time rate (NPTR) is another concept often used, which presents the total time rate for planning an on-time arrival on the 95% confidence level. Unlike the planning time index that is computed as the 95th percentile travel time and generally used for a specific link/path/corridor, the NPTR and proposed quantities in this paper have physical units and meanings, more importantly, can handle the heterogeneous trips with various origin and destination locations. Specifically, NPTR takes into account more explicitly the extreme travel time delays, hence, given by

$$NPTR = \frac{\sum_i \sum_j w_{ij} \cdot \tau_{ij,95\%}}{\sum_i \sum_j w_{ij}} \tag{7}$$

$$NBTR = \frac{\sum_i \sum_j w_{ij} \cdot \beta_{ij}}{\sum_i \sum_j w_{ij}} \tag{8}$$

$$NBTRI = \frac{\sum_i \sum_j w_{ij} \cdot \beta'_{ij}}{\sum_i \sum_j w_{ij}} \tag{9}$$

The network mobility and reliability features discussed above will be combined with individual trip records directly extracted from the on-demand ride service platform of DiDi to form a comprehensive dataset for the ridesplitting analysis. The dataset includes both individual trip features and the population level of features estimated on the aggregate origins, destinations, and OD pairs. Results will be shown in Section 5.1 and Appendix A.

4.3. Feature Selection: The ReliefF algorithm

Feature selection refers to the process of identifying the few most important variables or parameters and using only this subset as features in the ridesplitting behavior classification. It has become a strategy of data dimension reduction by selecting a subset of important predictor variables to create a classification model. In this paper, the reasons to conduct dimension reduction for understanding ridesplitting behavior can be summarized as follows:

(1) Reduce the impact of highly correlated predictor variables on generalization capability of the model prediction. Sometimes too much information can reduce the effectiveness of classification. Some features assembled for building and testing a model may not contribute meaningful information to the model. Some may even detract from the quality and accuracy of the model. Moreover, datasets with many attributes may contain groups of attributes that are correlated. For example, if two or more of the independent variables or predictors are correlated to the dependent or predicted variable, then the estimates of coefficients in a classification model tend to be unstable or counter intuitive.
(2) Simpler models are more interpretable, less likely to be overfitting with training datasets, simpler to explain, and more robust on small datasets. Feature selection can be viewed as a method for replacing a complex classifier (using all features) with a simpler one (using a subset of the features). Due to the bias-variance tradeoff in statistic learning, weaker models are often preferable when limited training data are available.
(3) Feature selection can reduce time complexity and space complexity of predictor variables, result in less computation, fewer parameters, and save the cost of observing the feature. It makes training and applying a classifier more efficient by decreasing the size of the effective features. For example, the larger dimension of predictors, the higher probability of having missing values in the training data. More features involved in a classification model, more efforts should be made to prepare the training data. If cases were removed which had missing values for some predictors, the learning model may end up with a shortage of observations.
(4) Irrelevant features simply add noise to the training data and affect the classification accuracy, while feature selection often increases classification accuracy by eliminating noise features. For example, a noisy feature is one that, when added to the ridesplitting representation, increases the classification error on new ride orders. Such an incorrect generalization from an accidental property of the training set is called overfitting. Noise increases the size of the model and the time and system resources needed for model building and scoring. Irrelevant features can skew the logic of the classification algorithm and affect the accuracy of the model.

Feature importance is useful as a preprocessing step in classification modeling. To identify an influential subset of significant predictor variables, this paper employs the ReliefF algorithm for feature selection. To minimize the effects of noise, correlation, and high dimensionality, selecting the most relevant features is sometimes a desirable preprocessing step for understanding ridesplitting behavior. It should be worthy to note that though the decision tree algorithm includes components that rank features as a part, it does not give a generalized subset of features due to the dependence on the classifier, like all other wrapper methods that usually are of high computational complexity. We therefore use a computationally inexpensive ReliefF algorithm to rank features.

The original Relief algorithm presented by Kira and Rendell (1992a, 1992b) is a feature selection algorithm for the binary classification that uses the instance based learning to assign a relevance weight to each feature. The key of the Relief algorithm is that the nearest instances of the same class should be closer than the nearest instances of any other classes. It assumes that a feature is important if it can easily segregate different classes from one another. The ReliefF algorithm (Kononenko, 1994) is an extension of the original Relief algorithm. The primary logic is that an important feature should differentiate between instances from different classes and has the same value for instances from the same class. ReliefF can deal with multiclass problems and incomplete and noisy data. It can be applied to the feature selection problem that includes interaction among features, and/or local dependencies. The strength of the ReliefF algorithm is its independence on heuristics, low-order polynomial time in computation, and noise-tolerance and robustness to feature interactions.

For each iteration i, the ReliefF algorithm randomly selects an instance, i.e. R_i, from the training data and then searches for k of its nearest neighbors from the same class (nearest hit H_j) and k nearest neighbors from each of the different classes (nearest misses $M_j(C)$). The user-defined parameter k controls the locality of the estimates. In this paper, it can be safely set to 10 (refer to empirical discussions in Kononenko (1994)). The weight of a feature is updated according to how well its values distinguish the sampled instance from its nearest hit and nearest miss. The quality estimation or weight $W[A]$ for every feature A depends on R_i, H_j, and $M_j(C)$. The weight updating scheme is illustrated as follows: if instances R_i and H_j have different values of the feature A, we decrease its quality estimation $W[A]$; otherwise, we desirably increase $W[A]$. The contribution for each class of the misses is weighted with the prior probability of that class $P(C)$ (estimated from the training set). The sum of probabilities for the misses' classes is represented by $1 - P(\text{class}(R_i))$. The whole iteration is repeated for user-defined m times.

The function diff which calculates the difference between the values of the feature A for two instances I_1 and I_2. For nominal features, it is defined as:

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0 & \text{value}(A, I_1) = \text{value}(A, I_2) \\ 1 & \text{value}(A, I_1) \neq \text{value}(A, I_2) \end{cases} \tag{10}$$

and for numerical attributes as

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(\text{A}) - \min(\text{A})} \tag{11}$$

ReliefF algorithm for feature selection$*$
Input: A training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, a vector of feature values and the class values
Initialization: Let all weights $W[A] = 0$
for $i = 1$ **to** m **do**
  Randomly select an instance R_i;
  Find k nearest hits H_j;
  for each class $C \neq \text{class}(R_i)$ **do**
    From class C find k nearest misses $M_j(C)$;
  end for
  for all features $A = 1$ **to** a **do**
    $W[A] = W[A] - \sum_{j=1}^{k} \frac{\text{diff}(A, R_i, H_j)}{m \cdot k} + \sum_{C \neq \text{class}(R_i)} \frac{P(C) \cdot \sum_{j=1}^{k} \text{diff}(A, R_i, M_j(C))}{[1 - P(\text{class}(R_i))] \cdot m \cdot k}$;
  end for
end for
return the vector **W** of estimations of the qualities of features

5. Results

5.1. Preprocessing of features

In this paper, we study the 30-day (November 1–30, 2015) historical trip and fare logs of DiDi on-demand ride service data of Hangzhou, China, to investigate the ridesplitting behavior of passengers. The city was divided into 30 km × 43 km, i.e., 1290 grids, each area of which is 1 km^2. The boundaries are 120.00 degrees east longitude, 120.45 degrees east longitude, 30.12 north latitude, and 30.39 degrees north latitude.

As shown in Table 1, trips with any features out of the predefined ranges or sets were removed from this analysis. For example, if a trip's origin or destination was beyond the rectangular study area, then the trip would not be taken into consideration. After the preprocessing, the dataset used in this paper includes 718043 trips of e-hailing Taxi using the DiDi app, Private Car Service, Hitch, and Express made during November 2015. The data were further randomly split into a training set and a test set, both of which were about half of the extracted individual trip records.

In addition to the individual records (see descriptive statistics in Section 3), we estimated OD trip statistics on the population level using all of the available historical on-demand ride service platform data. For each origin/destination grid of 1 km^2, statistics of all trips that started from the origin grid or ended in the destination grid were estimated, such as the average number of daily trips, average trip distance, average waiting time, and average response time. For each OD pair, some percentiles of the travel time rate (e.g., 5%, 50%, and 95%) were also estimated based on historical trips that started from the origin grid and ended in the destination grid. For example, the average number of OD daily trips as a possible influencing feature on ridesplitting behavior is listed in Table 1.

Ridesplitting behavior can be influenced by many factors, possibly including the aforementioned features. Moreover, the weather and air quality may also affect the ridesplitting choices. In Table 1, we list all of the measurable features in the dataset that may make effect on ridesplitting behavior.

5.2. Selected important features

While using the ensemble learning model for the ridesplitting behavior analysis, the first step is to determine feature importance. Fig. 10 shows the feature importance ranking results returned by the ReliefF algorithm described in Section 4.3. The features used to build the classification model are ranked in the order of their significance in predicting the target. The results show that the feature of trip travel time is the most important in predicting the ridesplitting behavior for the training set. Together with the surge pricing ratio, trip fee, trip distance, pickup time or passenger waiting time, these top five features have the most important effect on whether or not a passenger will choose to split rides with other passengers. Negative ranking indicates noise. Thus features ranked at zero or less do not significantly contribute to the prediction and should probably be removed from the data. By considering the features with positive importance weights, 13 features are finally selected. In addition to the top five features, the remaining 8 important features include destination average daily trips, ride order response time, origin average response time, destination average response time, OD travel time rate percentile at 5%, OD travel time rate percentile at 50%, OD travel time rate percentile at 95%, and destination average waiting time.

5.3. Parameter tuning for ensemble learning

To identify the best settings of parameters for the ensemble learning approach, this paper is devoted to finding out the best combination of parameters for the Boosting ensemble decision tree model. These include the following:

Table 1
Features for the ridesplitting behavior analysis.

ID	Features	Type	Level	Unit	Range/set	Mean	S.D.*
1	Day of week	Category	Individual	day	{Monday, …, Sunday}	NA	NA
2	Hour of day	Discrete	Individual	hour	[1, 24]	NA	NA
3	Weather	Category	Individual	NA**	{0 = sunny; 1 = rainy; 2 = cloudy}	NA	NA
4	Air quality index***	Discrete	Individual	NA	(0, 500]	76.87	20.07
5	Trip distance	Continuous	Individual	km	(0, 100]	5.95	5.04
6	Trip fee	Continuous	Individual	RMB****	(0, 200]	15.28	10.82
7	Ride order response time	Continuous	Individual	sec	(0, 600]	22.28	30.88
8	Pickup time or waiting time	Continuous	Individual	sec	(0, 3600]	365.09	232.97
9	Trip travel time	Continuous	Individual	min	[1, 120]	17.91	14.33
10	Origin average daily trips	Continuous	Population	trips	$(0,+\infty)$	147.50	108.31
11	Origin average trip distance	Continuous	Population	km	(0, 100]	6.62	1.69
12	Destination average daily trips	Continuous	Population	trips	$(0,+\infty)$	147.29	115.65
13	Destination average trip distance	Continuous	Population	km	(0, 100]	6.60	2.13
14	Origin average waiting time	Continuous	Population	min	(0, 1440]	6.12	1.41
15	Origin average response time	Continuous	Population	sec	(0, 3600]	22.58	7.76
16	Destination average waiting time	Continuous	Population	min	(0, 1440]	26.66	22.76
17	Destination average response time	Continuous	Population	sec	(0, 3600]	196.62	171.54
18	OD travel time rate (5%)	Continuous	Population	min/km	(0, 30]	2.14	0.97
19	OD travel time rate (50%)	Continuous	Population	min/km	(0, 30]	3.12	1.05
20	OD travel time rate (95%)	Continuous	Population	min/km	(0, 30]	5.75	3.01
21	OD average daily trips	Continuous	Population	trips	$(0,+\infty)$	87.21	150.83
22	Type of service	category	Individual	NA	{1 = Taxi; 2 = Private; 3 = Hitch; 4 = Express}	NA	NA
23	Surge pricing ratio	Continuous	Individual	100%	(0, 5]	1.00	0.34

* S.D.: standard deviation.
** NA: not applicable.
*** Provided by China's Ministry of Environmental Protection based on the level of 6 atmospheric pollutants (SO_2, NO_2, PM_{10}, $PM_{2.5}$, CO and O3).
**** US$ 100 = RMB 664.

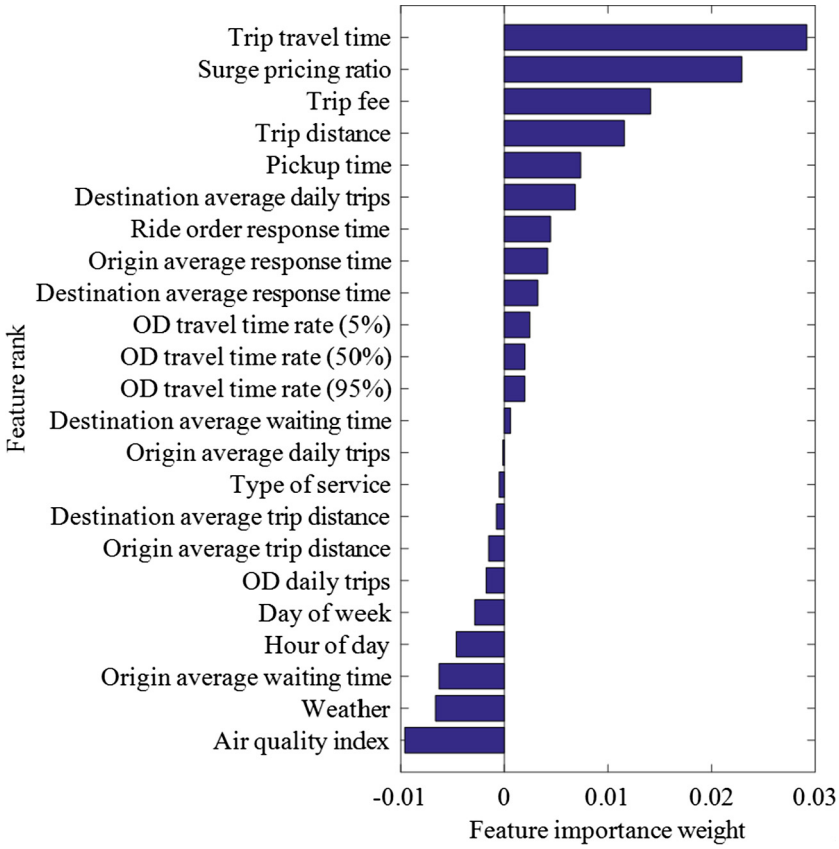

Fig. 10. Feature importance ranking by the ReliefF algorithm.

(1) Number of ensemble trees. The larger of the number the better, but the classification error will stop getting significantly smaller beyond a critical number of trees.
(2) Complexity of the decision tree. Deeper trees can be grown for better accuracy. When Boosting decision trees, the simplest trees named stumps (a tree with one split) are grown by default. The default values of the tree depth controllers in this paper are 1 for the maximal number of decision splits or branch nodes (i.e., growing stumps), 5 for the minimum number of leaf node observations, and 10 for the minimum number of branch node observations. In the decision tree, splitting branch nodes layer by layer will terminate until the maximal number of decision splits is reached, or the number of observations at least in one leaf/branch node is fewer than the minimum number of leaf/branch node observations. To reduce the dimension of parameter tuning for the tree depth control, this paper will demonstrate the choice of maximal splits in the following.
(3) Learning rate for shrinkage between 0 and 1 (default value). It controls the contribution of the weak learners in the final combination. Smaller values of the learning rate require larger numbers of weak classifiers to maintain a constant training error or achieves a better accuracy. Empirical evidence suggests that small values of the learning rate favor better test error (Hastie et al., 2009). Thus, the learning rate strongly interacts with the number of ensemble trees.

The optimization yields values for the aforementioned three parameters which give the best classification accuracy or the least classification error. Then the best performing tree after parameter tuning will be checked against the test dataset to verify if there is overfitting. Finally, these optimized parameters will be used in the final ensemble learning model.

In this paper, we first grow and cross validate a deep classification tree and a stump for comparison purposes. Then the Boosting ensembles are trained using 200 classification trees by exponentially increasing the maximum number of splits from the set {1, 2, 4, 8, 16, 32, 64}, and adjusting the learning rate to each value in the set {0.1, 0.25, 0.5, 1}. The ensembles are cross validated in the 5-fold mode by estimating their mean squared errors (MSE). The parameter tuning criterion is to select the ensemble with the lowest MSE.

The parameter tuning process is presented in Fig. 11, which shows how the 5-fold cross-validated classification error behaves as the number of trees in the ensemble increases for a few of the ensembles, the deep tree (minimum one branch node observation), and the stump (maximal one decision split). Each curve contains a minimum cross-validated MSE occurring at the optimal number of trees in the ensemble. Optimal values of the number of trees, maximum number of splits, and

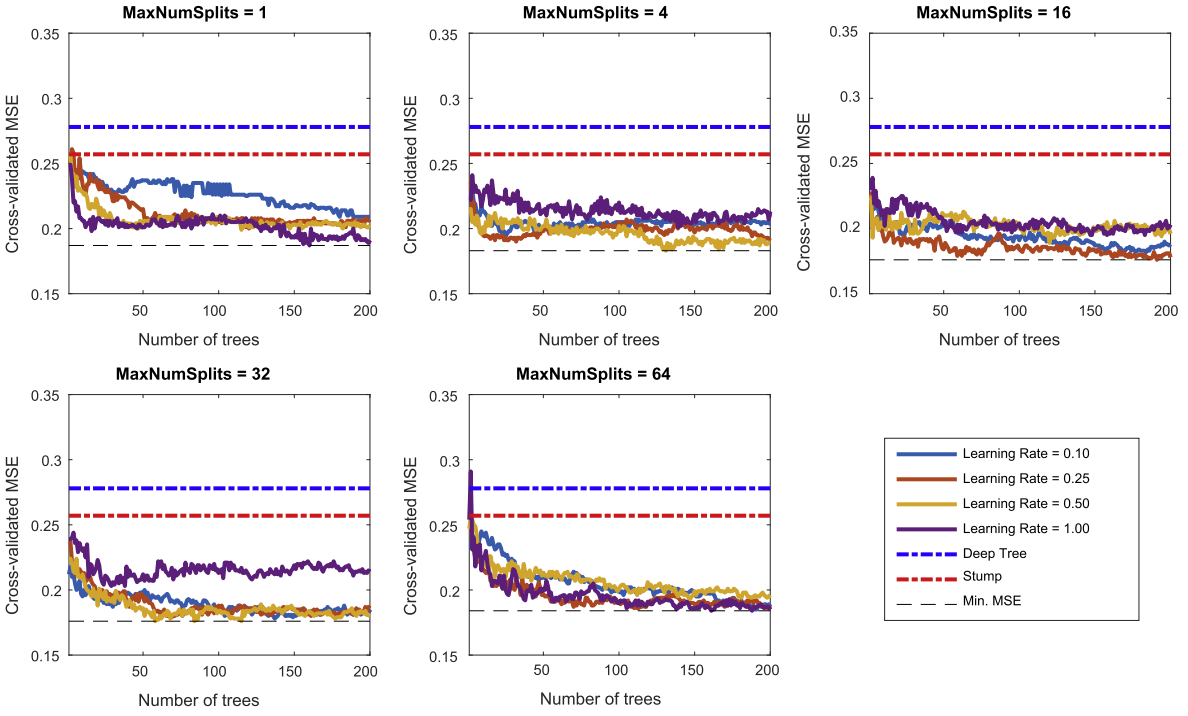

Fig. 11. Learning curves with ensemble trees.

learning rate are identified that yields the lowest MSE overall. Fig. 11 shows that the minimum MSE is 0.176, corresponding to the optimal number of ensemble trees as 191, the maximal number of splits as 16, and the learning rate as 0.25.

5.4. Evaluation of ensemble learning

The ensemble learning model is evaluated by applying it to the test set with known response values that are compared with the predicted values. In this paper, both the training set and test set are built using data from the same historical dataset. Specifically, 50% of the records is randomly selected and used to train the ensemble learning model, while the remaining records are used to test the model. Before applying trained ensemble trees to the independent test set, the 5-fold cross-validation is used to estimate the accuracy of our models. It counts the overall frequency of different types of errors. For the binary ridesplitting classification problem, the accuracy of a model is then measured by the values in Table 2.

Four ensemble learning models are compared considering the following options: applying full features, selecting important features (see Section 5.2); using default parameters of the Boosting ensemble trees, and using optimal parameters (see Section 5.3). As shown in Table 2, the ridesplitting choice is denoted by class 0; otherwise, class 1. The full features based ensemble learning involves all of the 23 features into a strong classifier, while the selected feature based ensemble learning only utilizes the most important 13 features in this paper. Results show that ensemble learning models after parameter tuning outperform those models using default parameter values. The reduction in feature dimension maintains most classification power of the ensemble learning model without losing much accuracy (e.g., 79.6% and 79.3% for full features and selected features with optimal ensemble learning parameters, respectively) and AUC, which is the area under the receiver operating characteristic (ROC) curve mearing how well a parameter can distinguish between binary classes. AUC measures the discriminating ability of a binary classification model. The larger the AUC, the higher the likelihood that an actual positive case will be assigned a higher probability of being positive than an actual negative case. When the feature dimension is reduced, there is even an increase from 78.8% to 79.2% in the overall accuracy for the models using default ensemble learning parameters.

In practice, the selection of the best model according to these measures depends on the importance or cost of misclassification in each class. In our experiments, we select models on the basis of their overall accuracy. Since the models are compared and evenly selected by the cross-validation on the training set, this model performance might be slightly overoptimistic and, in any case, suffers from high variance when the sample size is small. Hence, we continue to test ensemble learning with optimal parameters using classification errors. The evaluation is done both on the training set and on the test set of independent observations.

The classification error or loss measures the predictive inaccuracy of classification models. When comparing the same type of loss among many models, a lower loss indicates a better predictive model. It is defined as the weighted fraction of misclassified observations:

Table 2
Performance of different ensemble learning classifiers on the training set.[a]

Performance indicators	Full features based ensemble learning (default parameters) (%)	Selected feature based ensemble learning (default parameters) (%)	Full features based ensemble learning (optimal parameters) (%)	Selected feature based ensemble learning (optimal parameters) (%)
TPR (true class 0)	85.5	85.8	83.8	84.3
FNR (true class 0)	14.5	14.2	16.2	15.7
TPR (true class 1)	68.6	69.1	73.2	71.6
FNR (true class 1)	31.4	30.9	26.8	28.4
PPV (predicted class 0)	80.7	81.0	82.7	82.0
FDR (predicted class 0)	19.3	19.0	17.3	18.0
PPV (predicted class 1)	75.5	76.0	74.7	74.9
FDR (predicted class 1)	24.5	24.0	25.3	25.1
FPR (positive class 0)	31.4	30.9	26.8	28.4
TNR (positive class 0)	68.6	69.1	73.2	71.6
FPR (positive class 1)	14.5	14.2	16.2	15.7
TNR (positive class 1)	85.5	85.8	83.8	84.3
Overall accuracy	78.8	79.2	79.6	79.3
AUC	85.7	84.8	87.4	87.0

[a] TPR: true positive rate or sensitivity; FNR: false negative rate or miss rate; FPR: false positive rate or fall-out; TNR: true negative rate or specificity; PPV: positive predictive value or precision; FDR: false discovery rate; AUC: the area under the ROC curve.

$$L = \sum_{i=1}^{n} \theta_i \mathrm{I}(f(\mathbf{x}_i) \neq y_i) \tag{12}$$

where L is the weighted average classification loss or error. y_i is the i th observed binary class label. The weight for observation i is θ_i, which is normalized so that all of the weights sum to the corresponding prior class probability. Therefore, $\sum_{i=1}^{n} \theta_i = 1$.

Table 3 shows classification errors of the ensemble learning models of the last two columns in Table 2. The 5-fold cross-validation errors on the training set are smaller than those on the test set. But the validation results of both ensemble learning models on the test set are comparable, which means the prediction capability of the selected feature based ensemble learning model is still promising.

Table 3
Classification errors of ensemble learning models with optimal parameters.

Ensemble quality validation methods	Full features based ensemble learning	Selected feature based ensemble learning
Cross-validation on the training set	0.177	0.202
Validation on the independent test set	0.225	0.224

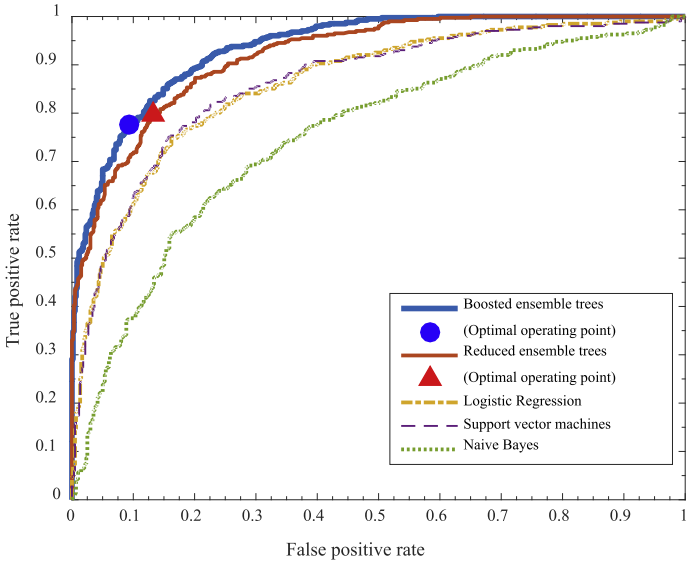

Fig. 12. ROC curves of ensemble learning, logistic regression, SVM, and naive Bayes classification for the ridesplitting analysis (true class 1).

5.5. Comparison with other models

The ROC is a useful metric for comparing predicted and actual target values in a classification model, and evaluating how a classifier behaves with different probability thresholds of false positive rates. It gains insight into the comparison of decision-making ability for various models, as shown in Fig. 12. The ROC curve for a model represents all the possible combinations of values in its confusion matrix. For the ridesplitting class (true class), this paper compares the optimized Boosting ensemble trees using all features or selected features with three other widely used classifiers: the logistic regression model to estimate the posterior probabilities for ridesplitting behavior, an SVM classifier on the same sample data, and a naive Bayes classifier. The optimal operating point of the ROC curve is determined as the cut point such that both sensitivity and specificity are maximized (Gallop et al., 2003). The optimal operating points of Boosting ensemble trees based on full/selected features are (0.094, 0.776) and (0.132, 0.796), respectively. ROC curves are computed for those classifiers. It shows that both Boosting ensemble trees outperform other single models because the ensemble ROC curves are closer to the upper left corner.

6. Conclusions

In this paper, we present an ensemble learning approach for better understanding ridesplitting behavior of passengers of ridesourcing companies who provide prearranged and on-demand transportation services. To improve the prediction accuracy of ridesplitting choices, we explored real-world data extracted from the on-demand ride service platform of DiDi in Hangzhou, China. Based on the empirical observations of over a million trips of four service types, i.e., Taxi Hailing Service, Express, Private Car Service, and Hitch, the ridesplitting behavior is modeled as a general binary classification problem taking into account a variety of features that may impact ridesplitting such as order time, trip length, fee, weather, air quality, travel time and reliability of origins/destinations and so on. Important features are ranked and selected by using the ReliefF algorithm. The ensemble learning approach is presented to train both the full (with all features) and reduced (with only important features) ridesplitting models by appropriately fusing the estimations obtained by multiple weaker but efficient classification decision trees. Results show that the Boosting ensemble trees with full features return classification errors of 0.177 on the training set, and 0.225 on the test set, respectively. While the Boosting ensemble trees with selected features return comparable results without losing too much accuracy. This paper verifies that ensemble is particularly useful and powerful in the ridesplitting analysis and outperforms other widely used classifiers such as logistic regression, SVM, and naive Bayes classification. Ensemble learning may not be widely applied in the transportation field but it seems like a clean window which helps us to understand the world better. The case study shows that it can predict ridesplitting behavior with better accuracy in comparing with other methods.

In addition, it should be pointed out that ensemble learning is a widely used tool in statistics and statistical learning. On one hand, only Boosting ensemble learning for classification and its applications in assembling week decision trees are discussed for the ridesplitting behavior analysis. It could be also used to assemble other non-machine learning individual ridesplitting choice models such as discrete choice models. On the other hand, ensemble regularization can be incorporated to the learning framework that is a process of choosing fewer weak classifiers but does not diminish predictive performance, e.g., lasso regularization. This paper has shed some light on exploring the real-world on-demand ride service platform data of individual levels. In the near future, we expect to extend this ensemble learning technique into some other applications based on the ridesourcing data, including the assessment and prediction of gaps between travel demand (passengers) and supply (drivers), and on-demand matching of drivers and passengers.

Acknowledgements

This research is financially supported by Zhejiang Provincial Natural Science Foundation of China LR17E080002, and National Natural Science Foundation of China 51508505, 51338008.

Appendix A. Network-wide statistics

Travel time reliability is very important to travelers. The influence of network reliability on behavioral choice such as mode, route, destination, departure time, etc. is worthy to investigate. Many travelers are risk-averse that they will arrive on time at their destination. But travel time usually follows a right-skewed distribution which makes the low probability of some extreme delays. A number of reliability measures have been adopted by several studies. However, travel time reliability on the network level needs to be developed when exploring urban transportation mobility.

The distributions of OD-pair travel time reliability indices of the study network is shown in Fig. A1. For the specific ridesplitting analysis in this study, the OD level travel time rates at 5%, 50% and 95% are employed as features. Overall, the weighted average network-wide statistics of Hangzhou are calculated as NFFTR = 2.05 (min/km), NTTR = 3.14 (min/km), NPTR = 5.86 (min/km), and NBTR = 0.84. This implies that the on-demand ride service platform data can be used for the reliability measurement and performance evaluation. Important feature ranking results in Section 5.2 shows incorporating the OD-pair travel time reliability into the ridesplitting analysis is worthy trying in this study.

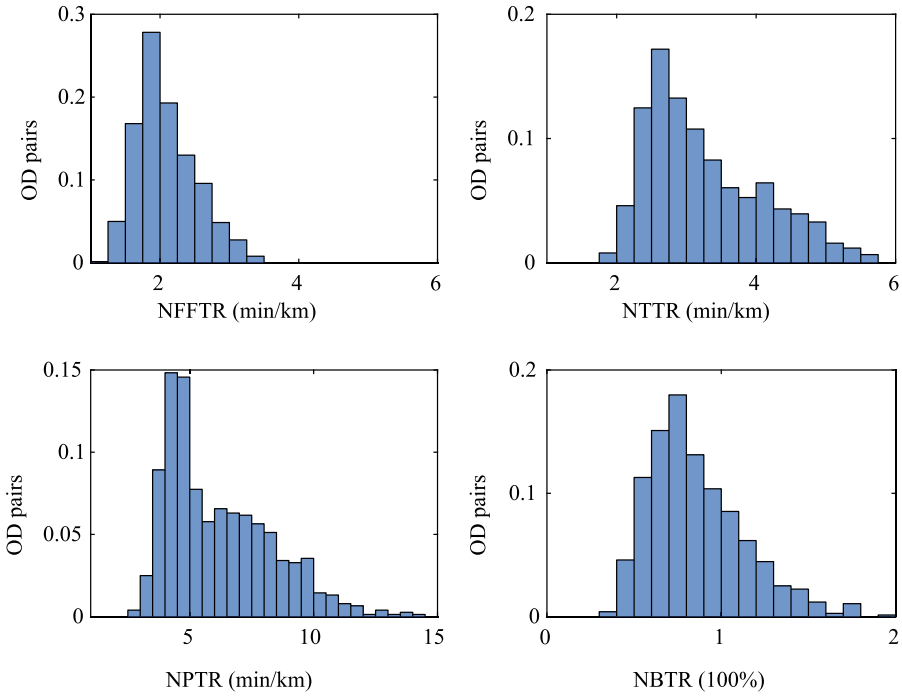

Fig. A1. Network reliability indicator distributions of OD pairs.

References

Alam, S., Shafi, K., Abbass, H.A., Barlow, M., 2009. An ensemble approach for conflict detection in free flight by data mining. Transport. Res. Part C: Emerg. Technol. 17 (3), 298–317.

Analysys International, 2016. <http://www.analysyschina.com> (accessed on June 30, 2016).

Atasoy, B., Ikeda, T., Song, X., Ben-Akiva, M.E., 2015. The concept and impact analysis of a flexible mobility on demand system. Transport. Res. Part C Emerg. Technol. 56, 373–392.

Brieman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.

Brieman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Cervero, R., Golub, A., Nee, B., 2007. City carshare: Longer-term travel demand and car ownership impacts. Transport. Res. Rec. J. Transport. Res. Board 1992, 70–80.

Chan, N.D., Shaheen, S.A., 2012. Ridesharing in North America: past, present and future. Transp. Rev. 32 (1), 93–112.

Chen, X., Zhao, J., 2013. Bidding to drive: car license auction policy in Shanghai and its public acceptance. Transp. Policy 27, 39–52.

De Oliveira, J.A.P., Doll, C.N., Kurniawan, T.A., Geng, Y., Kapshe, M., Huisingh, D., 2013. Promoting win–win situations in climate change mitigation, local environmental quality and development in Asian cities through co-benefits. J. Cleaner Prod. 58, 1–6.

DiDi, 2016. <http://www.xiaojukeji.com/en/company.html> (accessed June 30, 2016).

Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transport. Res. Part C Emerg. Technol. 40, 1–13.

Fei, X., Lu, C.C., Liu, K., 2011. A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. Transport. Res. Part C Emerg. Technol. 19 (6), 1306–1318.

Freund, Y., Schapire, R., 1995. A decision-theoretic generalization of on-line learning and an application to Boosting. In: Proceedings of the Second European Conference on Computational Learning Theory. Springer, Berlin, pp. 23–37.

Friedman, J.H., 2001. Greedy function approximation: a gradient Boosting machine. Ann. Stat. 29 (5), 1189–1232.

Friedman, J.H., 2002. Stochastic gradient boosting. Comput. Stat. Data Anal. 38 (4), 367–378.

Gargiulo, E., Giannantonio, R., Guercio, E., Borean, C., Zenezini, G., 2015. Dynamic ride sharing service: are users ready to adopt it? Proc. Manuf. 3, 777–784.

Gallop, R.J., Crits-Christoph, P., Muenz, L.R., Tu, X.M., 2003. Determination and interpretation of the optimal operating point for ROC curves derived through generalized linear models. Understand. Stat. 2 (4), 219–242.

Hastie, T., Tibshirani, R., Friedman, J., 2009. Elements of Statistical Learning. Springer.

He, F., Shen, Z.J.M., 2015. Modeling taxi services with smartphone-based e-hailing applications. Transport. Res. Part C Emerg. Technol. 58, 93–106.

Jiang, X., Zhang, L., Chen, X., 2014. Short-term forecasting of high speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China. Transport. Res. Part C 44, 110–127.

Kira, K., Rendell, L.A., 1992a. The feature selection problem: Traditional methods and new algorithm. In: Proceedings of AAAI'92.

Kira, K., Rendell, L.A., 1992b. A practical approach to feature selection. In: Sleeman, D., Edwards, P. (Eds.), Machine Learning: Proceedings of International Conference (ICML'92). Morgan Kaufmann, pp. 249–256.

Kononenko, I., 1994. Estimating attributes: analysis and extensions of Relief. In: De Raedt, L., Bergadano, F. (Eds.), Machine Learning: ECML-94. Springer Verlag, pp. 171–182.

Le Vine, S., Lee-Gosselin, M., Sivakumar, A., Polak, J., 2014. A new approach to predict the market and impacts of round-trip and point-to-point carsharing systems: case study of London. Transport. Res. Part D Transp. Environ. 32, 218–229.

Leng, B., Du, H., Wang, J., Li, L., Xiong, Z., 2016. Analysis of taxi drivers' behaviors within a battle between two taxi apps. IEEE Trans. Intell. Transp. Syst. 17 (1), 296–300.

Li, L., Chen, X., Zhang, L., 2014. Multimodel ensemble for traffic state estimations. IEEE Trans. Intell. Transp. Syst. 15 (3), 1323–1336.

Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., Li, Y., 2015. Robust causal dependence mining in big data network and its application to traffic flow predictions. Transport. Res. Part C Emerg. Technol. 58, 292–307.
Mahmassani, H., Hou, T., Saberi, M., 2013. Connecting network-wide travel time reliability and the network fundamental diagram of traffic flow. Transport. Res. Rec. J. Transport. Res. Board 2391, 80–91.
Martin, E., Shaheen, S., 2011a. Greenhouse gas emission impacts of carsharing in North America. IEEE Trans. Intell. Transp. Syst. 12 (4), 1074–1086.
Martin, E., Shaheen, S., 2011b. The impact of carsharing on public transit and non-motorized travel: an exploration of North American carsharing survey data. Energies 4 (11), 2094–2114.
Martin, E., Shaheen, S., 2016. The impacts of Car2go on vehicle ownership, modal shift, vehicle miles traveled, and greenhouse gas emissions: An analysis of five North American cities. Working Paper, Transportation Sustainability Research Center, University of California, Berkeley.
Martin, E., Shaheen, S., Lidicker, J., 2010. Impact of carsharing on household vehicle holdings: results from North American shared-use vehicle survey. Transport. Res. Rec. J. Transport. Res. Board 2143, 150–158.
Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transport. Res. Part C Emerg. Technol. 19 (4), 606–616.
Nourinejad, M., Roorda, M.J., 2016. Agent based model for dynamic ridesharing. Transport. Res. Part C Emerg. Technol. 64, 117–132.
Rayle, L., Dai, D., Chan, N., Cervero, R., Shaheen, S., 2016. Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco. Transp. Policy 45, 168–178.
Richardson, A., Taylor, M., 1978. Travel time variability on commuter journeys. High Speed Ground Transport. J. 12 (1), 77–99.
Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S.H., Ratti, C., 2014. Quantifying the benefits of vehicle pooling with shareability networks. Proc. Natl. Acad. Sci. 111 (37), 13290–13294.
Schor, J., 2014. Debating the Sharing Economy. <http://www.greattransition.org/publication/debating-the-sharing-economy>.
Schrank, D., Eisele, B., Lomax, T., Bak, J., 2015. Urban Mobility Scorecard. Texas A&M Transportation Institute and INRIX, College Station.
Schrank, D.L., Lomax, T.J., 2007. The 2007 Urban Mobility Report. Texas Transportation Institute, Texas A & M University, pp. 5–6.
Shaheen, S., Cohen, A., Zohdy, I., 2016. Shared Mobility: Current Practices and Guiding Principles. U.S. Department of Transportation, Federal Highway Administration. Report No. FHWA-HOP-16-022.
Shi, Q., Abdel-Aty, M., 2015. Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. Transport. Res. Part C Emerg. Technol. 58, 380–394.
Shih, G., 2015. China taxi apps Didi Dache and Kuaidi Dache announce $6 billion tie-up, Reuters February 14, 2015 available at: https://www.yahoo.com/tech/china-taxi-apps-didi-dache-kuaidi-dache-announce-023235253–finance.html.
Speed, C., Shingleton, D., 2012. An internet of cars: connecting the flow of things to people, artefacts, environments and businesses. Proceedings of the 6th ACM workshop on Next Generation Mobile Computing for Dynamic Personalised Travel Planning, pp. 11–12.
Teubner, T., 2014. Thoughts on the sharing economy. Proceedings of the International Conference on e-Commerce, vol. 11, pp. 322–326.
Waze Mobile, 2014. Get the Best Route, Every Day, with Real-time Help from Other Drivers. <https://www.waze.com>.
Wolpert, D.H., 1992. Stacked generalization. Neural Networks 5 (2), 241–259.
Zha, L., Yin, Y., Yang, H., 2016. Economic analysis of ride-sourcing markets. Transport. Res. Part C Emerg. Technol. 71, 249–266.
Zhang, Y., Haghani, A., 2015. A gradient boosting method to improve travel time prediction. Transport. Res. Part C Emerg. Technol. 58, 308–324.