Received August 6, 2018, accepted October 29, 2018, date of publication November 12, 2018,
date of current version December 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879824

Large-Scale Price Interval Prediction at OTA Sites

JINCHAO HUANG[1], LIN ZHU[2], BO FAN[3], YIHONG CHEN[2], WEN JIANG[2], AND SHENGHONG LI[1]

[1]School of Cyber Security, School of Electronic Information and Electrical Engineering, MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China
[2]Ctrip Travel Network Technology (Shanghai) Co., Ltd., Shanghai 200050, China
[3]School of International and Public Affairs, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Shenghong Li (shli@sjtu.edu.cn)

This work was supported by the National Key Research and Development Project of China under Grant 2016YFB0801003, and in part by the National Nature Science Foundation of China under Grants 61771342 and 61731006.

ABSTRACT With the rapid growing proliferation of online travel agent (OTA) services, personalized recommendations are highly valuable as they can improve customer experiences by preventing the information overload problem. The accurate prediction of users' expectations for price plays an important role in the personalized recommendation of hotels in OTA platforms. Considering that customers' preferences for hotel prices are actually acceptable ranges and traditional point estimations may neglect some informative aspects of the prediction, interval estimation is more suitable for the problem investigated in this paper. However, existing related methods are not applicable due to some specific issues. To provide a better personalized recommendation of hotels in OTA platforms, this paper proposes a novel interval forecasting solution to improve the accuracy of predicting users' price preferences. The novel interval forecasting solution first puts forward a customized objective function which could directly measure the quality of constructed intervals, while also allowing for adjustable tradeoffs between interval tightness and prediction reliability. Then, it combines alternating direction optimization and the gradient boosting framework to efficiently aggregate weak individual predictors to optimize the introduced learning objective. Empirical comparisons conducted on several benchmark standard datasets and a large-scale dataset shared with us by a major Chinese OTA site demonstrate the effectiveness of the proposed approach.

INDEX TERMS Prediction interval, customized objective function, gradient boosting, alternating direction optimization, online travel agent (OTA).

I. INTRODUCTION

By combining the offline business opportunities with the convenience offered by the rapid development of information technology, online travel agents (OTAs) provide customers with timely tourism information consultation, hotel reservation, transportation ticketing, and various other services. Its rapid development greatly promotes the growth of online tourism, while also making itself an indispensable part of people's travel experience [1]–[4]. For example, in 2015, OTAs have become the second-largest hotel booking channel in three big European countries (France, Germany and the UK) and the third-largest one in the US, with 24% and 17% market share, respectively. These numbers are expected to further rise to 30% and 25% by 2020 [5]; in China, tourists spent $87 billion via OTA platforms in 2016, much more than the amount spent via traditional travel agencies [6]. With such growing proliferation of OTA services, a massive amount of user transaction log data are being collected at major OTA platforms on a daily basis, which can be exploited to construct data-driven approaches for predicting the intentions of users and then suggesting proper accommodations to them. Such recommendations are highly valuable as they can improve customer experiences by preventing the information overload problem, and efficiently prioritize relevant information to travelers [7].

When customers enter an OTA site to make choices about hotels, their decision making are influenced by the attributes of the viewed hotels that they consider as important, such as star ratings, amenities, locations, availability of breakfast services, and proximity to certain positions of interest. Among all the factors that would affect the users' evaluation process, the hotel price level that he/she would prefer is of great importance. The underlying reason is that due to the remarkable transparency of e-commerce markets and low search costs, it has become much easier for consumers to compare between different OTA venues and hotels, and then select the best bargain; meanwhile, as OTAs typically display the recommendation results as a web page that lists a number of hotels, each one of which are supplemented with only a few pictures and brief textual descriptions that summarize

its attractive features, it is generally challenging for hotels to clearly differentiate among each other [8]. Under such a scenario, the hotel price stands out as one aspect that can be easily and clearly assessed by the customers, and thus figures prominently during their evaluation process. As a result, the accurate prediction of users' expectations for price plays an important role in the personalized recommendation of hotels in OTA platforms.

Computationally, the prediction of hotel price can be cast as a regression task and solved using various off-the-shelf tools. Most of these standard regression models return a single value for each test instance, representing the predicted conditional mean of the associated response variable [9]. However, such point estimations may neglect other informative aspects of the prediction, such as how reliable the predicted results are [10]. Furthermore, customers' preferences for hotel prices are influenced by variations of travel purposes, destinations, social context, and so on. It is necessary to properly account for such inherent uncertainties when designing a well-performing price predictive system.

In regression analysis, a commonly adopted statistical concept for assessing uncertainty is the prediction interval (PI) [11]. Rather than returning a single point estimation of the response variable, PI estimates an interval in which it probably lies within [12], and by adaptively assigning different interval widths to different instances, the varying uncertainty of each prediction can be better measured [10]. Compared with point estimation, interval estimation is also more suitable for our considered application of price-based hotel recommendations to customers, since intuitively such preferences may better be described as a pair of upper and lower bounds instead of only a fixed value.

Due to the potential usefulness of interval forecasting in regression, a number of related approaches have been proposed in the literature for tackling it [10]–[24]. Nevertheless, as is discussed in Section II, price interval prediction problem investigated in this paper is accompanied with some specific issues that are not well-addressed in previous interval prediction works, such as scalability, missing values handling, and distribution skewness.

In this paper, we propose a novel interval forecasting solution to improve the accuracy of predicting OTA users' price preferences. Firstly, we apply feature engineering techniques to extract relevant features from customers' recent and historical data for model training and prediction. Based on the constructed data, we propose a customized objective function which could directly measure the quality of constructed intervals, while also allowing for adjustable tradeoffs between interval tightness and prediction reliability. Then we combine alternating direction optimization and the gradient boosting framework [25]–[28] to efficiently aggregate weak individual predictors to optimize the introduced learning objective. Empirical comparisons conducted on a large-scale dataset shared with us by a major Chinese OTA site demonstrate the effectiveness of the proposed approach.

The rest of this paper is organized as follows. In Section II, we give a brief introduction about the related works and their suitableness for the considered problems. The new method is described in section III. Descriptions of the datasets and the experiment results are shown in section IV. Section V concludes the paper and discusses some guidelines for future work.

Notations: In the following, $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ denote the training set and the corresponding value of the response variable, while $\widetilde{Y} = \{\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_n\}$, $\widetilde{U} = \{\widetilde{u}_1, \widetilde{u}_2, \ldots, \widetilde{u}_n\}$ and $\widetilde{L} = \{\widetilde{l}_1, \widetilde{l}_2, \ldots, \widetilde{l}_n\}$ denote the predicted mean, upper bound, and lower bound of the response variable.

II. PREVIOUS WORKS

A number of approaches have been proposed in the literature for constructing intervals in regression problems. Based on the statistical theory, some of the commonly adopted techniques include Bayesian posterior inference [11], [13]–[16] and bootstrap [12], [17], [18], however, due to limitations such as high computational costs and restrictive assumptions about the data distribution (e.g., Gaussian), it is difficult to apply these techniques to large-scale interval prediction problems [12]. Alternatively, inspired by related methodologies such as quantile regression [19] and support vector regression machines [29], a number of works attempt to learn regression functions that directly predict the upper and lower bounds of the intervals [10], [20]–[23]. For example, quantile-based regression methods aim at estimating quantiles of the posterior distribution of the response variable. Concretely, let the distribution function of random variable y_i be:

$$F_{y_i}(b) = prob(y_i \leq b) \tag{1}$$

Given $\tau in(0, 1)$, then τ-quantile of y_i is denoted as:

$$Q_{y_i}(\tau) = inf\{b : F_{y_i}(b) \geq \tau\} \tag{2}$$

The objective function of τ-quantile regression is then defined as [30]:

$$Obj = \sum_{i=1}^{n} (\tau |y_i - \widetilde{y}_i| \mathbb{I}(y_i \geq \widetilde{y}_i) + (1 - \tau)|y_i - \widetilde{y}_i| \mathbb{I}(y_i < \widetilde{y}_i)) \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function which returns 1 if the input argument is true and 0 otherwise. Given the objective function (3), the choices of the regression function can be flexible. Linear quantile regression leads to a convex optimization that can be solved via various approaches such as simplex method, interior point method and smoothing method [31]. On the other hand, nonlinear regression methods such as gradient boosting decision tree [28], [32] has also been applied to realize the quantile regression in machine learning software packages such as Scikit-learn [33].

Once the regression function is chosen, the prediction interval can be obtained by minimizing (3) with different values

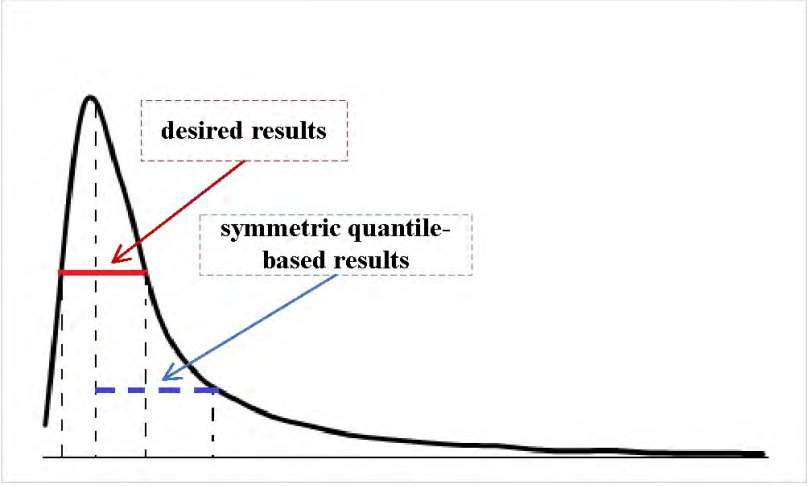

FIGURE 1. The case of unsymmetrically distributed data. The blue and red intervals cover the same amount of probability mass, and yet the red interval is shorter.

of τ [9]. For example, let $\widetilde{y}_{i,0.05}$ and $\widetilde{y}_{i,0.95}$ denote the predictions of y_i given by regression functions that respectively minimize (3) with $\tau = 0.05$ and $\tau = 0.95$, then the 90% prediction interval for y_i is given by

$$[\widetilde{l}_i, \widetilde{u}_i] = [\widetilde{y}_{i,0.05}, \widetilde{y}_{i,0.95}] \tag{4}$$

On the other hand, instead of minimizing a cost function, quantile regression forests [9] makes use of the alternate explanations of random forests (RF) [34] as an adaptive nearest neighbor method [35], and modify RF to directly estimate the entire conditional distribution, from which quantiles can be estimated and then fed into (4) to obtain the final prediction interval.

Despite the proven usefulness of interval prediction methods discussed so far, price interval prediction problem investigated in this paper has some specific characteristics that are not well-addressed in previous works, especially when these characteristics are combined together:

1) Efficiency: The large amount of daily-gathered transaction data call for algorithms those are efficient both when training and predicting. Despite improvements of algorithmic designs, existing approaches are generally not well-suited for very large-scale data.
2) Prediction Error with Skewed Distributions: Similar to (4), many existing intervals approaches still output a "symmetric" prediction interval, in other words, it is implicitly assumed that deviations from the predicted mean/median are symmetrically distributed. However, such assumptions may not be realistic and there may exist a more informative summary than the central interval for highly asymmetric data. For example, considering the asymmetric distribution illustrated in Fig.1, the interval specified by the red line covers the dense part of the posterior distribution, while the symmetric quantile-based interval is comparably longer and awkwardly cut off a part of the posterior distribution with high density.
3) Missing value: Compared with general e-commerce sites such as Amazon and Taobao, the user activities of OTA sites are often of much lower level: OTA users may visit and book only a few times a year, for example just on holidays [36]. As a result, features that describe users' historical behavior may be missing for some of the data instances.
4) Nonlinearity: Compared with linear models, nonlinear models may better represent the complex interactions between data features and lead to significant accuracy improvement for e-commerce applications [37], [38]. However, a significant portion of current interval-based regression models are still linear, although nonlinear extensions to these methods are straightforward (e.g., kernel extensions and hand-crafted feature-interaction terms), and such extensions could nevertheless require solving a more computationally intensive optimization problem, making them less practical for large-scale problems.

III. OUR APPROACH

A. PROBLEM SETUP

To motivate our problem set-up, we first introduce an intelligent screening system that is being designed at a major Chinese OTA platform which serves 250 million users with a reservation network of around 1 million hotels in 200 countries. This system aims at recommending hotels within acceptable price ranges of users. As shown in Fig.2, after getting a destination city, the intelligent screening system firstly retrieves a list of hotels that are sorted based on their historical merits (e.g., popularity and overall user ratings), from which appropriate hotels that fall into the predicted price interval are then picked out and displayed to the users. Under such a scenario, more accurate price interval prediction can better assist the users in finding the most relevant hotels satisfying their requirements.

B. FEATURE ENGINEERING

Based on our understanding of the studied problem and previous experience of analyzing OTA booking data, we define

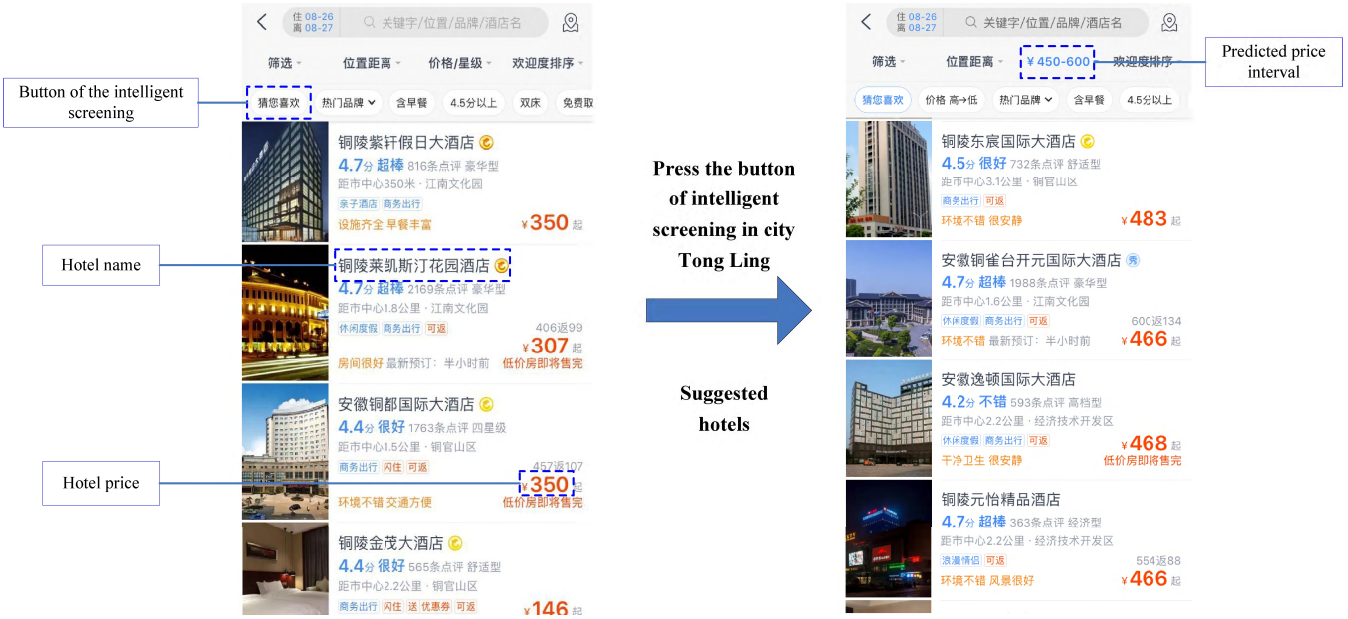

FIGURE 2. Example of the considered intelligent screening system in a Chinese city (Tong Ling).

TABLE 1. Basic features used in price interval prediction.

Category	Features	Descriptions
Historical data (simple statistics)	Order_num_lastyear (last6months/last3months/lastmonth/lastweek...)	Number of historical orders for a user
	Avgprice (medprice/minprice/maxprice...)_lastyear (last6months/last3months/lastmonth/lastweek...)	Average (medium/minimal/maximal) price of hotels that a user reserved in the specified time period
	Avgstar (medstar/minstar/maxstar...)_lastyear (last6months/last3months/lastmonth/lastweek...)	Average (medium/minimal/maximal) star of hotels that user reserved in the specified time period
	Avgprice_sameday_week_lastyear (last6months)	Average price of hotels that user reserved on the same day of a week in the specified time period
	Avgprice_dif_stars	Average price of hotels that user has ordered for different star ratings
Historical data (user attributes)	Prefer_star	User's favorite hotel star rating
	Ratio_weekday (weekend/holiday/workday)	Ratio of orders on different days
	Ratio_businesshotel (friendorder/...)	Ratio of different trip purposes: business purpose, trip with friends and so on
	Ratio_breakfast (double/big/single_bed, free_cancelpolicy...)	Ratio of various add-value services that the user ordered, such as breakfast, double bed, free-cancel policy
Recent browsing data	Browse_avgprice_lastweek (last3days/yesterday)	Average price of the hotels a customer clicked in the specified time period
	Browse_medprice_lastweek (last3das/yesterday)	Medium price of the hotels a customer clicked in the specified time period
	Browse_minprice_lastweek (last3das/yesterday)	Minimum price of the hotels a customer clicked in the specified time period
	Click_num_lastweek (last3das/yesterday)	Number of clicks

three groups of features, as shown in Table 1. These features mainly come from two sources: 1) Historical data and 2) Recent browsing data.

1) HISTORICAL DATA

Features that describe the users' historical orders can also be broadly categorized into two parts: a) simple statistics and b) user attributes.

Simple statics means that features are obtained simply by averaging historical data, such as avgprice_lastyear, avgprice_sameday_week_lastyear, avgprice_last6months, avgprice_sameday_week_last3months etc. Avgprice_lastyear denotes the average price of orders user placed in the past year. Differently, avgprice_sameday_week_lastyear represents the average price on the same day in a week during the past year. For example, supposing that the booking

date is Sunday, avgprice_sameday_week_lastyear represents the average price of the hotels reserved on Sunday in the past year. In a similar vein, avgprice_last6months and avgprice_sameday_week_last3months only change the length of the period from one year to six or three months. This group of features account for the largest proportion of extracted features.

On the other hand, user attributes require a relatively complex computation. For example, we calculate some user attributes from aspects of trip purpose, service requirements and degree of generosity, such as ratio_businesshotel, ratio_breakfast and so on. Ratio_businesshotel represents the proportion of business hotels in all hotels that the user has reserved before. And ratio_breakfast indicates the ratio of hotels serves breakfast.

2) RECENT BROWSING DATA

Before reserving a hotel, customers usually browse a lot of hotels, many of which share some similarity to the one that is finally booked. This implies that recent browsing information usually more accurately indicates what a user needs. We focus on some features related to price and clicks during different periods, such as browse_avgprice_lastweek, browse_maxprice_lastweek, click_num_yesterday and so on.

C. CUSTOMIZED OBJECTIVE FUNCTION

As illustrated in (4) and Fig.1, existing methods (such as quantile regression) for predicting intervals could lead to "central" intervals that sometimes may not appropriately account for the uneven shapes of estimation uncertainty. Essentially, such problems are due to the mismatch between the learning objectives of these methods (e.g., quantiles) and the quantities that we are truly interested in, namely the quality of the predicted intervals. In this section, we propose to tackle this problem by adopting a customized objective function that is more closely related to the interval quality.

As pointed out in [10], the quality of predicted interval $[\widetilde{l}_i, \widetilde{u}_i]$, $1 \le i \le n$ can be measured in two ways:

1) The accuracy of interval, namely how often does the true value y_i fall into the predicted interval $[\widetilde{l}_i, \widetilde{u}_i]$.
2) The average width of the interval, as narrower intervals indicate more precise localization of the true underlying value.

As wider intervals could increase the chance of including the true value, these two measures are essentially conflicting and trade-offs have to be made between them.

Motivated by the above considerations, we first consider an objective function defined as follows:

$$\begin{aligned}\ell(L,U) &= \sum_{i=1}^{n}(\underbrace{\mathbb{I}(\widetilde{l}_i \le y_i \text{ and } y_i \le \widetilde{u}_i)}_{\text{Interval accuracy term}} + \underbrace{\alpha(\widetilde{l}_i - \widetilde{u}_i)^2}_{\text{Interval width term}}) \\ &= \sum_{i=1}^{n}(\underbrace{\mathbb{I}(\widetilde{l}_i \le y_i)\cdot\mathbb{I}(y_i \le \widetilde{u}_i)}_{\text{Interval accuracy term}} + \underbrace{\alpha(\widetilde{l}_i - \widetilde{u}_i)^2}_{\text{Interval width term}}) \end{aligned} \quad (5)$$

which is a combination of 2 parts that respectively measure the accuracy and width of the inferred interval, with a parameter α that controls the trade-off between them. While being easy to understand conceptually, the interval accuracy term in (5) is non-convex and non-smooth, and difficult to optimize numerically. To make the computation more tractable, we use the convex hinge loss as a surrogate for the indicator function, and then modify (5) as:

$$\begin{aligned}\mathcal{L}(L,U) &= \sum_{i=1}^{n}\ell(\widetilde{l}_i, \widetilde{u}_i) \\ &= \sum_{i=1}^{n}(\underbrace{max(\widetilde{l}_i - y_i, 0) + \gamma max(y_i - \widetilde{u}_i, 0)}_{\text{Convex interval accuracy term}} \\ &\quad + \underbrace{\alpha(\widetilde{l}_i - \widetilde{u}_i)^2}_{\text{Interval width term}}) \end{aligned} \quad (6)$$

where the accuracy term in (5) is now replaced with the summation of two parts that respectively measure the loss of accuracy caused by inappropriate lower bound $\widetilde{l}_i$ and upper bound $\widetilde{u}_i$, with an additional hyper-parameter γ that balances the two parts.

D. DECISION TREE FITTING

Based on the defined objective function (6), we propose to learn $f_L(x)$ and $f_U(x)$, 2 nonlinear functions that respectively map the input instance to lower and upper bounds, in other words:

$$f_L(x_i) = \widetilde{l}_i, f_U(x_i) = \widetilde{u}_i, \quad 1 \le i \le n \quad (7)$$

A wide variety of nonlinear functions can be explored to infer $f_L(x_i)$ and $f_U(x_i)$. In this paper, we specifically consider gradient boosting decision trees (GDBT) [39], a powerful machine-learning technique that has a wide range of successful applications. GDBT is also particularly suitable for dealing with the problem properties discussed in Section II, as it can handle missing values elegantly and scale beyond billions of samples thanks to recent proposed boosting algorithms such as XGBoost [32], lightGBM,[1] and PV-Tree [40], while also allowing optimization of customized metrics that more suited to specific applications [41], such as (6).

Under the GDBT framework, $f_L(x)$ and $f_U(x)$ can be expressed as sums of regression tree, deonoted as $T(\cdot)$:

$$\begin{aligned} f_L(x) &= \sum_{i=1}^{K} T_{L,i}(x) \\ f_U(x) &= \sum_{i=1}^{K} T_{U,i}(x) \end{aligned} \quad (8)$$

As shown in (6), the learning of $f_L(x)$ and $f_U(x)$ is coupled and cannot be done independently. We deal with this problem via a hybrid learning framework that adopts alternating minimization method [42], [43] and boosting algorithm: as shown in Fig.3, in each iteration, $f_L(x)$ and $f_U(x)$ are updated in turn

[1] https://github.com/Microsoft/LightGBM

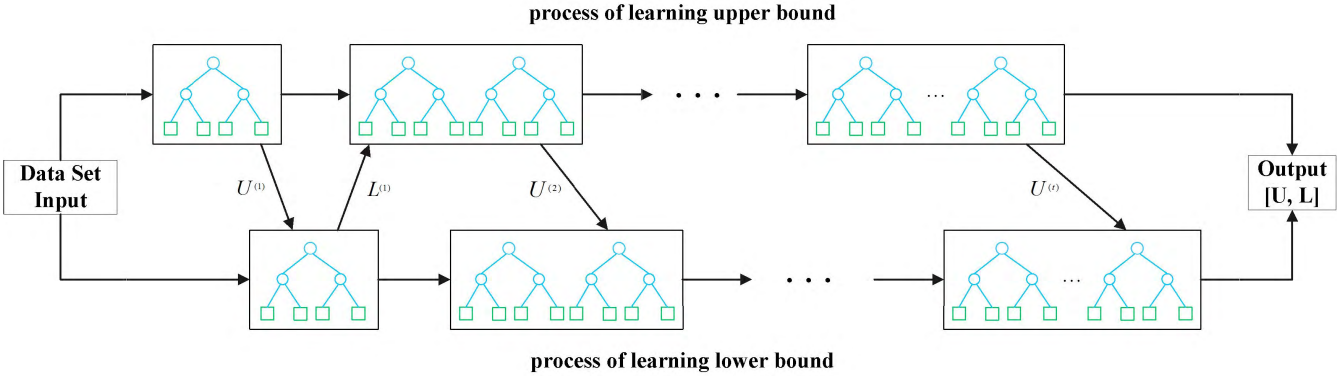

FIGURE 3. Framework of alternating iteration.

with the other one fixed, while gradient boosting is used to perform the updating.

Here, we adopt the boosting framework implemented in [32]. Concretely, let $f_L^k(x)$ and $f_U^k(x)$ be the lower and upper bound functions obtained at the i-th iteration, the combined objective function for updating $f_U^k(x)$ to $f_U^{k+1}(x) = f_U^k(x) + T_{U,k+1}(x)$ is written as follows:

$$Obj(T_{U,k+1}) = \sum_{j=1}^{n} \ell(f_L^k(x_j), f_U^k(x_j) + T_{U,k+1}(x_j)) + \sum_{j=1}^{k+1} \Omega(T_{U,j}) \quad (9)$$

where the function $\Omega(\cdot)$ measures complexity of the learned trees. Then, the objective function is approximated using the second order Taylor expansion as follows:

$$Obj(T_{U,k+1}) = \sum_{j=1}^{n} (\ell(f_L^k(x_j),\ f_U^k(x_j)) + g_i T_{U,k+1}(x_j) + \frac{1}{2} h_j T_{U,k+1}^2(x_j)) + \Omega(T_{U,k+1}) + const \quad (10)$$

where $g_j = \partial_{f_U^k(x_j)} \ell(f_L^k(x_j), f_u^k(x_j)), h_j = \partial^2_{f_U^k(x_j)} \ell(f_L^k(x_j), f_U^k(x_j))$. Removing constant terms, then the objective function can be approximated as follows:

$$Obj(T_{U,k+1}) = (\sum_{j=1}^{n} \ell(f_L^k(x_j), f_U^k(x_j)) + g_i T_{U,k+1}(x_j) + \frac{1}{2} h_j T_{U,k+1}^2(x_j)) + \Omega(T_{U,k+1}) \quad (11)$$

After that, the objective function is transformed into a quadric function from which closed-form solutions can be obtained, details about which can be found in [32]. As shown in (11), the gradient and hessian of the new loss function is required for the optimization, and they can be computed as follows:

$$g_j = \frac{\partial \ell(f_L^k(x_j), f_U^k(x_j))}{\partial f_U^k(x_j)} = \begin{cases} \gamma + 2\alpha(f_U^k(x_j) - f_L^k(x_j)) & f_U^k(x_j) \le y_j \\ 2\alpha(f_U^k(x_j) - f_L^k(x_j)) & f_U^k(x_j) > y_j \end{cases} \quad (12)$$

$$h_j = \frac{\partial^2 \ell(f_L^k(x_j), f_U^k(x_j))}{\partial (f_U^k(x_j))^2} = 2\alpha \quad (13)$$

In a similar vein, $f_L^k(x_j)$ is updated to $f_L^{k+1}(x) = f_L^k(x) + T_{L,k+1}(x)$ by solving

$$Obj(T_{L,k+1}) = \sum_{j=1}^{n} (\ell(f_L^k(x_j), f_U^k(x_j)) + g_j T_{L,k+1}(x_j) + \frac{1}{2} h_j T_{x_j}^2) + \Omega(T_{L,k+1}) \quad (14)$$

where

$$g_j = \frac{\partial \ell(f_L^k(x_j), f_U^k(x_j))}{\partial f_L^k(x_j)} = \begin{cases} 1 + 2\alpha(f_L^k(x_j) - f_U^k(x_j)) & f_L^k(x_j) \le y_j \\ 2\alpha(f_L^k(x_j) - f_U^k(x_j)) & f_L^k(x_j) < y_j \end{cases} \quad (15)$$

$$h_j = \frac{\partial^2 \ell(f_L^k(x_j), f_U^k(x_j))}{\partial (f_U^k(x_j))^2} = 2\alpha \quad (16)$$

The entire learning process is listed in Algorithm 1.

Algorithm 1 The Interval Learning Process.

1. Input: The input data.
2. Initialization: $f_L^0(x_j) = 0, f_U^0(x_j) = 0$
3. For $k = 1, 2, \ldots, K$ do
4. //(i)The process of learning the upper bound
5. Calculate derivative functions according to formulas (12) and (13).
6. Update $f_U^k(x)$ to $f_U^{k+1}(x)$ by solving (11).
7. //(ii)The process of learning the lower bound
8. Calculate derivative functions according to (15) and (16).
9. Update $f_L^k(x)$ to $f_L^{k+1}(x)$ by solving (14).
10. End for
11. Output: $[\widetilde{l}_i, \widetilde{u}_i]$, $1 \le i \le n$ (calculated using (7)).

IV. EXPERIMENT

A. DATASETS

In order to evaluate the performance of the proposed method, we conduct experimental comparisons on several standard benchmark datasets and a real-world OTA dataset.

TABLE 2. Statistics of UCI datasets.

Dataset	Sample Size (n)	No. Features (d)	Response variable
Forest Fires	517	12	Area: the burned area of the forest
Communities and Crime	1963	122	Violent Crimes per Population: total number of violent crimes per 100K popuation
Boston Housing	505	13	MEDV: median value of owner-occupied homes in $1000's
Online News Popularity	39644	58	Shares: number of shares in social networks (popularity)
Parkinson's Telemonitoring	5875	21	Motor_UPDRS: motor UPDRS scores

TABLE 3. Statistics of the OTA dataset.

Data	Time	Sample Size	No. Features(x)
Training set	2017-05-16	151249	155
Test set	2017-05-17	154053	155

1) STANDARD TEST DATASETS

The standard test datasets are obtained from the UCI machine learning repository, including Forest Fires,[2] Communities and Crime,[3] Boston Housing,[4] Online News Popularity[5] and Parkinson's Telemonitoring.[6] The sample sizes vary from $n = 505$ (Boston Housing) to $n = 39644$ (Online News Popularity), and number of features vary from $d = 13$ (Forest Fires) to $d = 122$ (Communities and Crime). For each of the dataset, about 80% of data points are randomly selected as the training set, with the rest as the test set. The statistics of the standard test data is shown in Table 2.

2) OTA DATASET

The real world data are obtained from a major Chinese OTA site, and we collect information of users who have reserved hotels from 2017-05-16 to 2017-05-17. In particular, we take samples of 2017-05-16 as the training set, and samples of the next day as the test set. About 7.1% of the data entries are missing. The statistics of the OTA dataset is shown in Table 3.

B. EVALUATION METRICS AND COMPARISON BASELINES

To measure the performance of various methods, we use the accuracy and interval width to analyze the results, these two metrics can be mathematically expressed as:

$$Accuracy = \frac{\sum_{i=1}^{n}(\mathbb{I}(\widetilde{l}_i \le y_i) \cdot \mathbb{I}(y_i \le \widetilde{u}_i))}{n} \tag{17}$$

$$IntervalWidth = \frac{\sum_{i=1}^{n} |\widetilde{u}_i - \widetilde{l}_i|}{n} \tag{18}$$

As discussed in Section III-C, (17) and (18) are essentially contradictory metrics and cannot attain the optimal value at the same time. Inspired by commonly used evaluation tools such as receiver operating characteristic (ROC) curve and precision-recall curve [44], for each method, we plot a curve that represents the evolution of Accuracy value as a function of the Interval Width, and the performance comparisons are conducted by inspecting and comparing these curves.

We use open source Boosting implementation XGBoost [32] through Python API to implement the proposed method. For the sake of comparison, we also use XGBoost to implement quantile regression as a baseline method. Meanwhile, linear quantile regression[7] and quantile regression forests[8] are considered as 2 additional baseline methods. To determine the hyper-parameters used in various methods, we selected another 20% of data instances from the training set, and chose the parameters that achieve the best results on such validation sets.

C. RESULTS ON UCI BENCHMARK DATA

The results of standard test data are shown in Fig.4, where the horizontal axis is the mean of the absolute value of predicted interval widths, and the vertical axis is accuracy. From it, we can draw the following observations. First, the curve of our approach is always higher than the others, which means the accuracy our approach attained is higher than the others when interval width is the same. Second, the improvement of the narrow interval is relatively higher throughout all intervals. Besides, the superiority of the proposed method is more obvious in the cases of Fig.4(b) (Communities and Crime) and Fig.4(d) (Online News Popularity), where the scale of the dataset or the number of features is large.

D. EXPERIMENTS ON OTA DATA

We apply our new method to the OTA data discussed in Section IV-A.2. Note that quantile regression forests is not evaluated here because it is too time-consuming for such large datasets (it did not finish after more than 3 days of computation), we only compare our approach with linear quantile regression and XGBoost-based quantile regression. Fig.5 shows that the performance of the new method is consistently better than XGBoost-based quantile regression throughout all intervals, especially when the width interval is narrow. For example, accuracy of XGBoost-based method is 0.712 with width 402, and accuracy of linear quantile regression is 0.708 with width 424, while our approach is up to 0.801 with width 373. These results indicate that the new method is also effective in dealing with OTA price prediction problems, which are always with large amount of data and features.

[2]https://archive.ics.uci.edu/ml/datasets/forest+fires
[3]http://archive.ics.uci.edu/ml/datasets/communities+and+crime
[4]https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
[5]https://archive.ics.uci.edu/ml/datasets/online+news+popularity
[6]https://archive.ics.uci.edu/ml/datasets/parkinsons+telemonitoring
[7]https://cran.r-project.org/web/packages/quantreg/index.html
[8]https://cran.r-project.org/web/packages/quantregForest/index.html

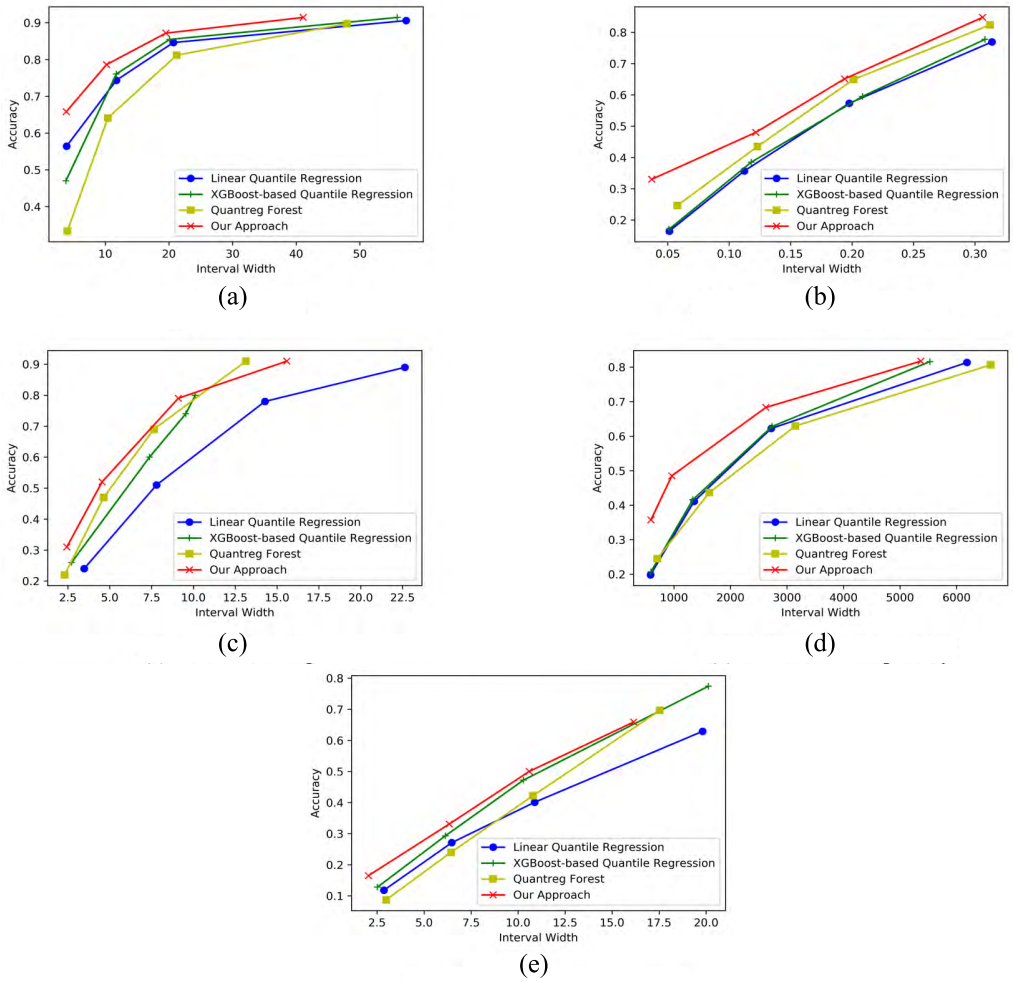

FIGURE 4. Comparison of performance on five standard test datasets. Results of all four methods are shown in figure a – e. (a) Forest fires. (b) Communication and crime. (c) Boston housing. (d) Online news popularity. (e) Parkinsons telemonitoring.

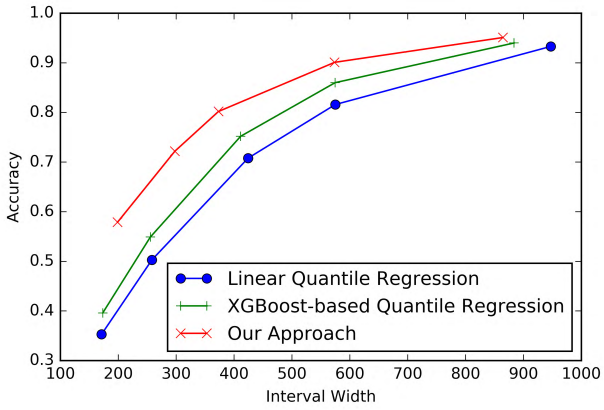

FIGURE 5. Comparison of performance on OTA dataset.

To further analyze the results, we divide users according to their own uncertainty, which is measured by the price difference between orders of the same user. Firstly, we select users who have placed multiple orders on the test day (2017-05-17), and calculate the largest price difference between orders as uncertainty for each of them. Then we divide these users into three equal groups according to the distribution of their uncertainty, and recount accuracy and interval width for each group respectively. The results are shown in Table 4. Differently, Table 5 is obtained by dividing users according to their uncertainty qualified via historical orders. The uncertainty of each user is measured by the largest price difference between orders placed in the last year, and users who have less than 2 orders in that time period are not considered here. To facilitate the comparison of results, we set the average interval width of three methods to the same value. Besides, we also name these three parts as first, second, and third group respectively, according to their uncertainty from small to large.

Table 4 and Table 5 both show that for all three groups, the prediction of our approach is much better than the other two methods. Meanwhile, for all compared methods, the prediction intervals widen as the user uncertainties become higher, illustrating the usefulness of interval

TABLE 4. Results of dividing the users by their uncertainty on the tested day.

	Linear Quantile Regression		XGBoost-Based Quantile Regression		Our Approach	
	Accuracy	Interval Width	Accuracy	Interval Width	Accuracy	Interval Width
First Group	0.689	330	0.739	307.3	0.87	334.3
Second Group	0.722	400.8	0.721	383.1	0.813	382.5
Third Group	0.597	580	0.611	569.5	0.607	483.2

TABLE 5. Results of dividing the users by their uncertainty in the last year.

	Linear Quantile Regression		XGBoost-Based Quantile Regression		Our Approach	
	Accuracy	Interval Width	Accuracy	Interval Width	Accuracy	Interval Width
First Group	0.657	235.9	0.712	221.9	0.860	280.5
Second Group	0.691	346.9	0.724	342.9	0.810	370.6
Third Group	0.693	642.0	0.713	645.6	0.715	524.6

TABLE 6. Top ten important features learned by quantile regression and our approach.

	XGBoost-based Quantile Regression		Our Approach	
	Lower bound	Upper bound	Lower bound	Upper bound
Features	Avgprice_dif_stars	Avgprice_dif_stars	Avgprice_dif_stars	Prefer_star
	Order_num_yesterday	Browse_avgprice_lastweek	Avgprice_weekend	Browse_avgprice_yesterday
	Avgprice_weekday	Avgprice_weekday	Avgprice_weekday	Consumption_level
	Avgprice_weekend	Avgprice_weekend	Price_lastorder	Avgprice_dif_stars
	Price_lastorder	Consumption_level	Minprice	Browse_avgprice_lastweek
	Minprice	Price_lastorder	Is_hotel_lowest_price_lastorder	Browse_avgprice_last3days
	Is_hotel_lowest_price_lastorder	Is_hotel_lowest_price_lastorder	Avgpremium_hotel_price	Avgprice_lastyear
	Browse_avgprice_lastweek	Ratio_cancel_order	Avgprice_lastyear	Avgprice
	Browse_maxprice_lastweek	Browse_maxprice_lastweek	Stdprice	Browse_medprice_lastweek
	Browse_avgprice_yesterday	Browse_avgprice_yesterday	Ratio_avgpremium_hotel_price	Browse_medprice_yesterday

prediction for quantifying uncertainties. However, the accuracy results of linear quantile regression and XGBoost-based quantile regression do not have any regularity: there is no significant difference in accuracy among the three groups. However, the accuracy of our approach is much better than the other two methods for users with more regular behavior, which means that our method can better make use of the reduced data uncertainties to improve the accuracy.

Finally, for XGBoost-based Quantile Regression and our approach, we report the most important variables in their learned models for predicting the upper and lower bounds of the interval. The variable importance is assessed using the built-in function ''importance'' of XGBoost, which scores each feature based on the improvement of the objective function value by using it as the splitting variable [39]

From the first two columns in Table 6, it is easy to know that the features of historical orders play an important role in the learning process of XGBoost-based quantile regression. And its top ten features of two bounds are almost the same. In contrast, the results of our approach are significantly different. The important features of the model for learning lower bound are mainly from historical data, which are consistent with the results of XGBoost-based method, while the model for learning upper bound tends to take full use of recent browsing information, such as browse_avgprice_yesterday, browse_avgprice_lastweek and so on. Compared to XGBoost-based method, our approach is better at learning difference between upper and lower bounds.

V. CONCLUSION

This paper proposes a novel interval forecasting solution to improve the accuracy of predicting users' price preferences in OTA sites. It combines alternating direction optimization and the gradient boosting framework to efficiently aggregate weak individual predictors to optimize the introduced learning objective. Empirical comparisons conducted on a large-scale dataset shared with us by a major Chinese OTA site demonstrate the effectiveness of the proposed approach.

This work can be extended in several directions. Firstly, better joint learning method may be designed for inferring the two regression functions that predict the upper and lower bounds. Secondly, explicit trade-off between accuracy and interval width still need to be specified in our approach, it would be preferable to design an improved method that can find all trade-offs between intervals at once, while still keeping the computation tractable. Our future research will focus on addressing these issues.

REFERENCES

[1] H. A. Lee, B. D. Guillet, and R. Law, ''An examination of the relationship between online travel agents and hotels: A case study of choice hotels international and expedia.com,'' *Cornell Hospitality Quart.*, vol. 54, no. 1, pp. 95–107, 2013.

[2] L. Ling, X. Guo, and C. Yang, ''Opening the online marketplace: An examination of hotel pricing and travel agency on-line distribution of rooms,'' *Tourism Manage.*, vol. 45, pp. 234–243, Dec. 2014.

[3] R. S. Toh, C. F. DeKay, and P. Raven, ''Travel planning: Searching for and booking hotels on the Internet,'' *Cornell Hospitality Quart.*, vol. 52, pp. 388–398, Nov. 2011.

[4] S. Perret and J. Barthel. *OTAs—A Hotel's Friend Or Foe?* Accessed: Jul. 8, 2015. [Online]. Available: https://www.hvs.com/article/7408/otas-%E2%80%93-a-hotels-friend-or-foe/

[5] *The Hotel Distribution Report 2016*. Accessed: Oct. 2016. [Online]. Available: http://hotelanalyst.co.uk/wp-content/uploads/sites/2/2016/10/2016-HA-Hotel-Distribution-report-final-sample.pdf

[6] *Chinese Tourists Spend More Via Online Travel Agencies: Report*, Xinhua, Beijing, China, 2017.

[7] J. Kiseleva *et al.*, "Where to go on your next trip?: Optimizing travel destinations based on user preferences," in *Proc. SIGIR*, 2015, pp. 1097–1100.
[8] B. Pan, L. Zhang, and R. Law, "The complex matter of online hotel choice," *Cornell Hospitality Quart.*, vol. 54, pp. 74–83, Feb. 2013.
[9] N. Meinshausen, "Quantile regression forests," *J. Mach. Learn. Res.*, vol. 7, pp. 983–999, Jun. 2006.
[10] N. A. Shrivastava, A. Khosravi, and B. K. Panigrahi, "Prediction interval estimation of electricity prices using PSO-tuned support vector machines," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 322–331, Apr. 2015.
[11] Y. Liu, A. Gelman, and T. Zheng, "Simulation-efficient shortest probability intervals," *Statist. Comput.*, vol. 25, pp. 809–819, Jul. 2015.
[12] T. Heskes, "Practical confidence and prediction intervals," in *Proc. NIPS*, 1997, pp. 176–182.
[13] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sep. 1992.
[14] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.
[15] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. New York, NY, USA: Springer, 2012.
[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2006.
[17] R. A. Stine, "Bootstrap prediction intervals for regression," *J. Amer. Stat. Assoc.*, vol. 80, no. 392, pp. 1026–1031, 1985.
[18] R. L. Schmoyer, "Asymptotically valid prediction intervals for linear models," *Technometrics*, vol. 34, pp. 399–408, Nov. 1992.
[19] R. Koenker, *Quantile Regression*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[20] X. Peng, "TSVR: An efficient twin support vector machine for regression," *Neural Netw.*, vol. 23, no. 3, pp. 365–372, 2010.
[21] H. A. Nielsen, H. Madsen, and T. S. Nielsen, "Using quantile regression to extend an existing wind power forecasting system with probabilistic forecasts," *Wind Energy*, vol. 9, pp. 95–108, Jan. 2006.
[22] J. B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy*, vol. 7, no. 1, pp. 47–54, 2004.
[23] H. Lee and H. Tanaka, "Upper and lower approximation models in interval regression using regression quantile techniques," *Eur. J. Oper. Res.*, vol. 116, pp. 653–666, Aug. 1999.
[24] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Netw.*, vol. 19, no. 2, pp. 225–235, 2006.
[25] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun, "A general boosting method and its application to learning ranking functions for Web search," in *Proc. NIPS*, 2008, pp. 1697–1704.
[26] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng, "Boosted multi-task learning," *Mach. Learn.*, vol. 85, nos. 1–2, pp. 149–173, 2011.
[27] C. J. Becker, C. M. Christoudias, and P. Fua, "Non-linear domain adaptation with boosting," in *Proc. NIPS*, 2013, pp. 485–493.
[28] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[29] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. NIPS*, 1997, pp. 155–161.
[30] I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola, "Nonparametric quantile estimation," *J. Mach. Learn. Res.*, vol. 7, pp. 1231–1264, Jul. 2006.
[31] C. Chen and Y. Wei, "Computational issues for quantile regression," *Sankhyā, Indian J. Statist.*, vol. 67, no. 2, pp. 399–417, 2005.
[32] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. KDD*, 2016, pp. 785–794.
[33] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[34] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[35] Y. Lin and Y. Jeon, "Random forests and adaptive nearest neighbors," *J. Amer. Statist. Assoc.*, vol. 101, no. 474, pp. 578–590, Jun. 2002.
[36] L. Bernardi, J. Kamps, J. Kiseleva, and M. J. Müeller. (2015). "The continuous cold start problem in e-commerce recommender systems." [Online]. Available: https://arxiv.org/abs/1508.01177
[37] G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas, "Interpretable predictions of tree-based ensembles via actionable feature tweaking," in *Proc. KDD*, 2017, pp. 465–474.
[38] Y. Huang *et al.*, "Telco churn prediction with big data," in *Proc. SIGMOD*, 2015, pp. 607–618.
[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2009.
[40] Q. Meng *et al.*, "A communication-efficient parallel algorithm for decision tree," in *Proc. NIPS*, 2016, pp. 1279–1287.
[41] Q. Wu, C. J. C. Burges, K. M. Svore, and J. Gao, "Adapting bboosting for information retrieval measures," *Inf. Retr.*, vol. 13, pp. 254–270, Jun. 2010.
[42] U. Niesen, D. Shah, and G. W. Wornell, "Adaptive alternating minimization algorithms," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1423–1429, Mar. 2009.
[43] I. Csiszar, "Information geometry and alternating minimization procedures," *Statist. Decisions*, no. 1, pp. 205–237, 1984.
[44] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. ICML*, 2006, pp. 233–240.

JINCHAO HUANG received the B.S. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Hubei, China, in 2015. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University. Her research interests include dimensionality reduction, machine learning, and recommendation systems.

LIN ZHU received the Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2013. He is currently an Algorithm Engineer with Ctrip Travel Network Technology (Shanghai) Co., Ltd. His research interests include latent feature learning, dimensionality reduction, and large-scale learning.

BO FAN received the Ph.D. degree in management information system from the Harbin Institute of Technology. He is currently a Professor with the School of International and Public Affairs, Shanghai Jiao Tong University. He has published in *Decision Support Systems*, the *European Journal of Operational Research*, *Expert Systems with Applications*, and so on. His research interests include E-government, big data analysis, and emergency management.

YIHONG CHEN received the Ph.D. degree from the School of Telecommunications, Tongji University. He is currently with Ctrip Travel Network Technology (Shanghai) Co., Ltd. His research interests include machine learning and artificial intelligence.

WEN JIANG received the Ph.D. degree from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2016. He is currently with Ctrip Travel Network Technology (Shanghai) Co., Ltd. His research interests include learning automata and their applications, time series analysis, pattern recognition, data mining (big data), machine learning, and hybrid intelligent systems.

SHENGHONG LI received the Ph.D. degree in radio engineering from the Beijing University of Posts and Telecommunications in 1999. Since 1999, he has been with Shanghai Jiao Tong University, as a Research Fellow, an Associate Professor, and a Professor, successively. In 2010, he was a Visiting Scholar with Nanyang Technological University, Singapore. His research interests include information security, signal and information processing, and artificial intelligence.