11-2020

Cost-sensitive deep forest for price prediction

Chao MA

Zhenbing LIU

Zhiguang CAO
Singapore Management University, zgcao@smu.edu.sg

Wen SONG

Jie ZHANG

See next page for additional authors

Citation

MA, Chao; LIU, Zhenbing; CAO, Zhiguang; SONG, Wen; ZHANG, Jie; and ZENG, Weiliang. Cost-sensitive deep forest for price prediction. (2020). *Pattern Recognition*. 107, 1-16.
Available at: https://ink.library.smu.edu.sg/sis_research/8122

Author

Chao MA, Zhenbing LIU, Zhiguang CAO, Wen SONG, Jie ZHANG, and Weiliang ZENG

Pattern Recognition 107 (2020) 107499

Contents lists available at ScienceDirect

Pattern Recognition

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

Cost-sensitive deep forest for price prediction

Chao Ma[a,b], Zhenbing Liu[b], Zhiguang Cao[c,*], Wen Song[d], Jie Zhang[e], Weiliang Zeng[f]

[a] *School of Informatics, Xiamen University, China*
[b] *School of Electronic Engineering and Automation, Guilin University of Electronic Technology, China*
[c] *Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore*
[d] *Institute of Marine Science and Technology, Shandong University, China*
[e] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[f] *School of Automation, Guangdong University of Technology, China*

ARTICLE INFO

Article history:
Received 25 March 2019
Revised 7 April 2020
Accepted 12 June 2020
Available online 15 June 2020

Keywords:
Cost-sensitive Deep Forest
Ensemble Deep Learning
Price Prediction
Modified K-means

ABSTRACT

For many real-world applications, predicting a price range is more practical and desirable than predicting a concrete value. In this case, price prediction can be regarded as a classification problem. Although deep forest is recognized as the best solution to many classification problems, a crucial issue limits its direct application to price prediction, i.e., it treated all the misclassifications equally no matter how far away they are from the real classes, since their impacts on the accuracy are the same. This is unreasonable to price prediction as the misclassification should be as close to the real price range as possible even if they have to be wrongly classified. To address this issue, we propose a cost-sensitive deep forest for price prediction, which maintains the high accuracy of deep forest, and propels the misclassifications to be closer to the real price range to reduce the cost of misclassifications. To make the classification more meaningful, we develop a discretization method to pre-define the classes of price, by modifying the conventional K-means method. The experimental results based on multiple real-world datasets (i.e., car sharing, house renting and real estate selling) show that, the cost-sensitive deep forest can significantly reduce the cost in comparison with the conventional deep forest and other baselines, while keeping satisfactory accuracy.

1. Introduction

Price prediction is an important task of machine learning and pattern recognition techniques [1]. It has been widely studied in many economic related areas, such as stock market [2], oil price [3], electricity load [4] and electricity price [5], real estates [6], and airfare [7], and plays a critical role in providing important decision support information. Based on the task, price prediction methods can be roughly classified into two types. The first type of methods is designed to forecast the trend of price in the time-series form, e.g. stock price and oil price prediction. The second type of methods focuses on predicting the price of individual items based on their features, e.g. the price of a house or air ticket. In this paper, we concentrate on the second type of price prediction task.

Most existing works of price prediction employ regression-based methods to predict a concrete value for the price. However, for many real-world applications, predicting a class (or range) of price is much more practical. This is typical for the sharing economy, which involves the sharing of a wide range of properties (e.g. cars, houses) and is rapidly expanding and becoming part of our life [8]. The P2P car sharing business, as a motivating example, is gaining more and more popularity in both research community and industry, as it not only facilitates the daily commute of people [9], but also has potential to alleviate traffic congestion and reduce pollution to the environment [10]. In general, the car fleet in P2P car sharing is decentralized, and owned by private individuals rather than a central operator. The principle role of the operator is to provide an online marketplace to connect car owners with prospective renters. Compared with other types, the P2P car sharing service is also characterized by the diverse options of cars [11], which can easily meet the specific requirements of renters, but may also cause problems. On one hand, new renter without much experience, may have no idea whether the price for the car he/she is interested in is reasonable or not. On the other hand, the new owner may not know whether the price he/she listed is proper enough, so that the car can be attractive and the owner can make desirable profit. Therefore, price prediction plays a significant role in the P2P car sharing business, which has the potential to make the whole process more efficient by saving the bargain between owners and renters. However, due to the characteristic of decentralization, predicting a class or a range which the price of the car

[*] Corresponding author.
E-mail address: isecaoz@nus.edu.sg (Z. Cao).

https://doi.org/10.1016/j.patcog.2020.107499
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

belongs to is much more practical than that of a concrete price value [12]. Because to both parties, it is always the cases that the deal will be made as long as the price falls in an acceptable range. In such scenarios which require predicting the class of price, the prediction task can be regarded as a classification problem rather than regression.

Although deep forest is considered as the best solution to many classification tasks [13], it cannot be directly apply it to the price prediction task. The traditional deep forest usually treats all the cost of misclassifications equally as their impacts on the classification accuracy are the same, e.g., misclassifying the true price class '500–599' as '400–499' does not make difference to the accuracy in comparison with '100-199'. However, taking into account the particular characteristics of price prediction, '400–499' is much more reasonable than '100-199' as the former is much closer to the true price class. In this case, the interested party (e.g. the owner or renter in P2P car sharing) will not miss the target too much regarding the price for the item, even if he/she made a wrong prediction. Thus, it is desirable to improve the conventional deep forest by considering the cost of each misclassification, such that the misclassifications will be forced to be closer to the true price classes, while keeping satisfactory accuracy.

To this end, in this paper we propose a cost-sensitive deep forest for price prediction. With deep forest, we expect to achieve higher accuracy than the conventional deep forest as the number of price classes goes up. With the cost-sensitive scheme that imposes higher cost to the misclassifications far away from the true price classes, we expect to reduce the cost of misclassifications and propel them to the true price ranges. In addition, to further improve the overall performance, we develop a discretization method to pre-define the classes of price, by modifying the conventional K-means method. The effectiveness of our method is verified by experimental results on multiple real-world datasets, including car sharing, house renting, and real estate selling. To summarize, we make the following contributions: (1) investigate how to apply deep forest to price prediction by considering it as a classification problem; (2) introduce the cost-sensitive scheme to deep forest; and (3) propose a modified K-means method for more meaningful discretization of the data.

The rest of this paper is organized as follows. Section 2 reviews the existing methods on discretization, classification and cost-sensitive learning. Section 3 elaborates the methods we proposed for the cost-sensitive price prediction. Experimental results and performance analysis are presented in Section 4. The paper ends with the conclusion and future works in Section 5.

2. Related work

In our price prediction problem, we need to firstly define the range or interval for each price class using a discretization method, and then perform the prediction using a classification method. Accordingly, we review the related works on discretization methods, followed by the classification methods, i.e., random forest and deep forest. Afterwards, we discuss the related works on the cost-sensitive learning.

2.1. Discretization for classification

Three typical discretization methods for classification were summarized by Torgo and Gama [14]: (1) Equally probable intervals (EPI), which divides continuous values into intervals with the same number of examples. (2) Equal width intervals (EWI), which divides the whole range of target values into intervals with the same span. (3) K-means clustering, which minimizes the total distance of each sample within an interval to its center given a fixed number of intervals. As an unsupervised method, K-means clustering is able to well explore the distribution of data, which has strong robustness to the data variance [15]. Inspired by the advantages, many subsequent discretization methods based on k-means have been developed [16]. Among them, the system RECLA is attractive, as it is able to transform a regression problem into a classification one by applying the K-means based discretization, thus the existing classification methods could be adopted to solve the new problem.

2.2. Classification methods: random forest and deep forest

For many classification problems, ensemble learning based methods can achieve better performance than the non-ensemble learning based ones [17]. Random forest (RF) is the most representative ensemble learning method, which yields many variants, such as Extra Tree (ET) [18], Rotation Forest (RoF) [19] and Oblique Decision Trees (ODT) [20]. RF classifier usually consists of many base classifiers, i.e., decision tree. For a classification problem, each base classifier makes a decision first, then all decisions are integrated and the category with most votes is considered as the target output. The overall performance of an ensemble classifier is usually influenced by the accuracy and diversity of base classifiers. Normally, the more accurate and diverse the base classifiers are, the better overall performance the ensemble classifier gets. Nevertheless, given the characteristic of the ensemble learning, there is no need to constantly increase the accuracy of each base classifier. On the other hand, it is more practical and rewarding to enhance the diversity of the base classifiers, which can be achieved through four strategies [21]: 1) Data sample manipulation, which samples different data to train different subclassifiers [22]. 2) Input feature manipulation, which generates different feature subspaces to train different subclassifiers [23]. 3) Learning parameter manipulation, which sets different parameter for different subclassifers [18]. 4) Output representation manipulation, which uses different output representations to generate different subclassifiers [24]. Meanwhile, multiple strategies can also be applied simultaneously.

Before deep forest (DF) is proposed, deep learning is approximately equal to deep neural network (DNN) [13]. Deep learning achieves great success in many tasks and most of its applications are based on DNN [25]. The success of DNN mainly benefits from three key characteristics [13]: 1) Layer-by-layer processing, by which different levels of features are extracted gradually. 2) In-model feature transformation, by which information extracted from the preceding layer is delivered to next layer as features. 3) Sufficient model complexity, which is necessary for exploiting large training data. However, the structure of DNN, especially the depth, needs to be designed and fixed before training, which is difficult to be pre-defined. Therefore, Zhou and Feng [13] proposed the deep forest (DF), the depth of which is self-adapting, thus opened a door to the non-neural-network style deep learning. DF is usually characterized by two components, i.e., multi-grained scanner and cascade forest. The former is used to extract information from raw data which may have spatial or temporal relationship, and the latter is used to construct the ensemble classifier with self-adapting depth. However, multi-grained scanner is not a necessity in all DF. In general, DF can achieve competitive performance compared to DNN in a broad range of tasks, with much less hyper-parameters and higher accuracy [26]. On one hand, DF is still regarded as an ensemble learning method based on decision trees, so it has most of the advantages of the general ensemble learning methods. On the other hand, DF also has the characteristics of DNN mentioned above to make the ensemble classifier deeper, which will potentially enhance diversity and improve the overall performance [27].

2.3. Cost-sensitive classification

Traditional classification methods are designed to achieve the lowest recognition errors and thus essentially assume same cost or loss for different misclassifications. However, this assumption may not be proper for many real-world applications. Therefore, cost-sensitive classification, assigning different cost to different misclassifications, is significantly desirable. Since [28] established the foundation of cost-sensitive classification, it has been introduced into Support Vector Machine (SVM) [29], Multilayer Perceptron (MLP) [30], RF [31] and RoF [32]. However, DF, which offers an alternative when DNN is not superior, still has not been incorporated with the cost-sensitive scheme. Meanwhile, different problems in real world have been widely addressed by cost-sensitive classification, such as, spam detection [33], face recognition [34], medical diagnosis [35], fraud detection [36], and class imbalance [37]. However, cost-sensitive classification has never been studied in the price prediction problem. Therefore, we aim to develop a comprehensive approach to predict the price by leveraging the deep forest and the cost-sensitive scheme in this paper.

3. Methodology

In this section, we first propose modified k-means to discretize the price into multiple intervals, which are considered as the price classes in the subsequent classification. Afterwards, we elaborate the cost-sensitive deep forest method for price prediction.

3.1. Modified K-means for discretization

In statistics, coefficient of variation (CV) is widely used to measure the variational level of a distribution, which is defined as the standard deviation divided by mean. The bigger CV, the higher variational level. In discretization, two CV based metrics are usually adopted to evaluate this performance, i.e., coefficient of variation for the number of examples in interval (CVN) and coefficient of variation for the range of interval (CVR). Generally, lower CVN indicates less imbalanced class, and lower CVR indicates more uniform range of interval. Therefore, a discretization method with both lower CVN and CVR is more desirable. However, in price prediction, the distribution of price is often uneven, i.e., most of the price usually locate in a small range, and the remaining minority locate in a large range, which may significantly affect the performance of discretization [38]. Among various discretization methods, K-means often demonstrates comparatively better performance in terms of CVN and CVR, as it takes into account the variance and mean of the data when generating the clusters. Nevertheless, there is still much room to further improve K-means for discretization.

Inspired by the uniform effect of K-means [39] and isolation forest [40], we propose the modified K-means to further improve the discretization performance. Uniform effect of K-means refers to that the size of clusters generated by K-means tends to be a relatively uniform distribution [39]. Particularly, the 95%-confidence interval of CVN of K-means on datasets without excessive outliers is [0.09, 0.85], which means the probability that the CVN is between 0.09 and 0.85 is 95%. Related to target domain (i.e., the price) rather than feature domain, outlier here refers to the data points with sparse distribution and far from the high-density groups. If the dataset contains excessive outliers, increasing the number of intervals will decrease CVN but increase CVR. To alleviate this contradiction and get both low CVN and CVR with small number of intervals, we incorporate the intervals with excessive outliers into the adjacent ones, where an isolation forest is adopted to select such intervals. More specifically, we employ the isolation forest to find outliers rather than profiling normal points [40], based on two characteristics of outliers: 1) the outliers have fewer examples, 2) the attribute values of outliers are much different from the normal ones. Then, the isolation forest iteratively splits the data space into subspaces using random hyperplanes, until each subspace contains only one data point. Intuitively, point groups with high density and normal values will stop splitting after many iterations, while groups with low density and outliers will stop splitting relatively early.

To derive the modified K-means, we firstly define an abnormal interval as the one that does not change if the number of intervals is increased. The number of abnormal intervals may be larger than one, and in this case, only the most abnormal interval is modified each time. Then, we define three types of most abnormal intervals, i.e., the interval with the largest range, the smallest number of examples, and the smallest density, respectively. Consequently, the modified K-means based on them are termed as KMR, KMN and KMD, respectively, while the conventional K-means is termed as KM. Accordingly, the overall steps for implementing the modified K-means are stated as follows: 1) run conventional K-means by setting the number of intervals as smallest, 2) increase the number of intervals by one until the interval of most abnormal is found, 3) set all prices in the abnormal interval as the nearest price in the direction of the dense part, 4) repeat step 1, 2, 3 based on the new price obtained from step 3, until the number of intervals in step 2 is equal to the desired number, 5) the final intervals from step 4 are the output of the modified K-means. Details of the modified K-means are described in Algorithm 1. In our price prediction, we adopt the modified K-means to conduct discretization, followed by the cost-sensitive deep forest to predict the price range (or interval).

Algorithm 1: Modified K-means.

```
input : $P$, original price data; $K$, number of intervals for
        modified K-means;
        $s$, the smallest number of intervals.
output: $I^+$, $K$ intervals by modified K-means.
initialize former number of intervals, $N^-$, as $s$;
initialize former price, $P^- = P$;
get $N^-$ intervals, $I^-$, by K-means based on $P^-$;
while $N^- < K$ do
    get new number of intervals, $N^+ = N^- + 1$;
    get $N^+$ intervals, $I^+$, formed by K-means based on $P^-$;
    if $I^-$ and $I^+$ have same intervals then
        choose the most abnormal interval;
        set all price in this interval as the price towards the
        nearest dense part of all price to get new price, $P^+$;
        reinitialize, $P^- = P^+$;
        reinitialize, $N^- = s$;
        get $N^-$ intervals, $I^-$, formed by K-means based on $P^-$;
    end
    else
        $N^- = N^+$;
        $I^- = I^+$.
    end
end
```

3.2. Cost-sensitive deep forest

Deep forest (DF) is a type of ensemble learning in nature, which is also endorsed with the strength of deep neural network [41]. Inside a DF, the unit in each layer is a base classifier, and the more complex of the base classifier, the more diverse of DF. In general, sufficient diversity with reasonable accuracy of base classifiers will lead to satisfactory overall performance [42]. Moreover, DF can further enhance the diversity by automatically adjusting the depth, given that the depth of DF is self-adapting. Therefore, DF is able to achieve relatively superior performance of accuracy in classification even if the number of the classes is large.

Although DF is supposed to improve the accuracy over the conventional classification methods, it overlooks a critic issue, i.e., the cost of different misclassifications is evaluated equally. This type of misclassifications evaluation is undesirable in many applications because they should be as near the corresponding true classes as possible, even though they may not facilitate increasing the accuracy. It is especially crucial for price prediction, where the misclassifications should not be far from the real price range. To address this issue, we propose a cost-sensitive deep forest (CSDF), which assigns a specific cost for each misclassification, while preserving satisfactory accuracy.

To construct the CSDF, cost-sensitive base classifiers should be developed. To this end, we define a cost matrix first. Particularly, based on the discretization by K-means or its improved variants, the median value of an interval is chosen as the center. Then the distance of two centers is considered as the distance of these two classes, which is considered as the cost for misclassifying a sample of one class as the other. In view of this, we assume that c_{ij} is the cost for misclassifying the sample of class i as class j, and the cost matrix C is expressed as follows,

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}, \tag{1}$$

where $c_{ij} = c_{ji}$, and $c_{ii} = 0$. Given a sample with class label y, the predicted class label $\bar{y}$ is obtained by optimizing the objective function as follows,

$$\bar{y} = \mathop{\arg\min}_{\bar{y}_j \in \{I_1, \dots, I_n\}} loss(y, \bar{y}_j), \tag{2}$$

$$loss(y, \bar{y}_j) = \sum_{i=1}^{n} P(\bar{y}_j | y) c_{ij}, \tag{3}$$

where I_j refers to the case that the prediction is j while the true label is i; $P(\bar{y}_j|y)$ means the posterior probability of the prediction when the true label is given. Then the cost of a classification task is the average cost of classifying all samples. In general, the cost not only reflects the correctness of right classifications, but also the degree of wrong classifications. Therefore, it is desirable to incorporate the cost metric to the base classifier in DF for price prediction.

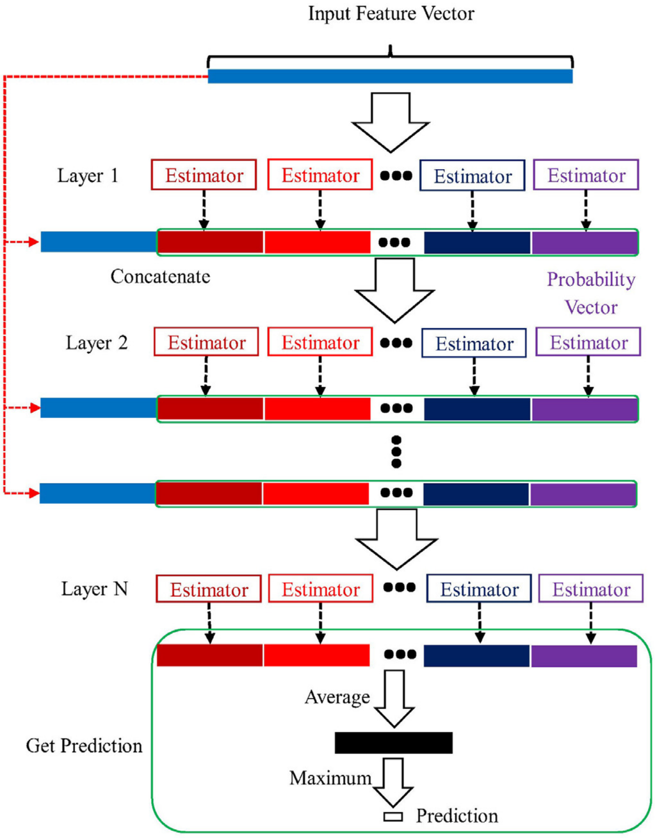


Fig. 1. The overall architecture of the proposed CSDF. (1) Different colors of estimators refer to different base classifiers, and different estimators will result in different probability vectors. The probability vectors will be concatenated with input feature vector to act as input to different layers. (2) The horizontal and vertical ellipsis symbols refer to the user-defined number of estimators within a layer and the flexible self-adapting number of layers, respectively. (3) The way to perform the prediction is shown in the green box, and the cost regarding the output of each layer can be obtained by referring to the cost matrix (i.e., Eq. (1)) based on the prediction and true label. Note that there is a green box in each layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Given the cost-sensitive base classifier, the overall architecture of CSDF is depicted in Fig. 1, which is of self-adapting depth with multiple layers. Meanwhile, each layer consists of multiple estimators, and each estimator is built with k-fold cross validation based on the cost-sensitive base classifier. More specifically, (1) regarding an estimator, the input is same with the input of that layer, and the output is a probability vector. Normally, an element with a higher value inside the probability vector refers to lower cost. The k-fold cross validation is applied in each estimator, and each base classier is cost-sensitive and trained on a fold. Moreover, the probability vector of an estimator is calculated as the average of the probability vectors of these k base classifiers, so an estimator is the ensemble of base classifiers, and the estimator details are described in Algorithm 2. (2) Regarding a layer, the input is the concatenation of raw feature vector and output of the preceding layer (if any), and the output is the probability vectors of all estimators of that layer. The type and amount of estimators among different layers should be the same, but could be different within the same layer. The details of a layer are described in Algorithm 3. (3) Regarding the CSDF, the input of the first layer is the raw feature vector, and the output of the layer is used to calculate the cost based on the cost matrix. If the cost decreases significantly, the

Algorithm 2: The estimator of a layer.

```
input  : Data $X = \{X_1, X_2 \cdots X_k\}$, where $X_k$ is a sub-dataset
         that equally and stratifiedly extracted from the raw
         feature input;
         label, $y = \{y_1, y_2 \cdots y_k\}$;cost matrix, $C$; classier, $Cl$.
output: probability vector on training set, $P_{tr}$;probability
         vector on validation set, $P_{cv}$;estimator, $E$
for $i = 1$ to $k$ do
    use all subdatasets expect $i$th subdataset to train a
    subclassifier, $Cl_i$, and get $P_{tr_i}$;
    use $i$th subdataset to test the subclassifier and get $P_{cv_i}$
end
integrate all subclassifiers to get $E$;
merge $P_{tr_k}$ to get $k-1$ probability vectors, $P'_{tr}$, for every
example;
average $P'_{tr}$ to get $P_{tr}$;
merge $P_{cv_k}$ to get $P_{cv}$.
```

Algorithm 3: The layer of a CSDF.

```
input  : data, $X$;label, $y$;cost matrix, $C$; a list of classifiers,
         $Cl = \{Cl_1, Cl_2 \cdots Cl_l\}$
output: new features for next layer, $F$;
         cost of layer, $c$;
         layer, $L$
for $j = 1$ to $l$ do
    train and test $j$th estimator, $Cl_j$, by Algorithm~2 to get $F_j$,
    and probability vector on validation set, $P_{cv_j}$
end
concatenate $F_j$ from all estimators to get $F$;
average $P_{cv_j}$ then find maximum as prediction, $\bar{y}$;
compare $\bar{y}$ with $y$ to get $c$ based $C$;
compose $L$ by the trained list of classifiers, $Cl$.
```

raw feature vector is concatenated with the output of that layer as the input of the next layer. The next layer is trained and evaluated again as described above, and this process is iterated until the cost does not decrease significantly. The results of the layer, whose cost stops decreasing is regarded as the output of CSDF. Note that from the computational perspective, the major difference between DF and CSDF is Eq. (3), where the latter has an additional term, i.e., c_{ij}. Once the dataset is given and the intervals are fixed, c_{ij} will be considered as a constant. Therefore, there is basically no incremental computation in comparison with the original DF. The details of CSDF are described in Algorithm 4.

Algorithm 4: The CSDF.

```
input  : raw data, $X_{raw}$;label, $y$; cost matrix, $C$; a list of
         classifiers for layer, $Cl = \{Cl_1, Cl_2 \cdots Cl_n\}$; the number
         of folds, $k$; the number of iteration for stopping, $s$;
         the degree of improvement, $p$.
output: cost sensitive deep forest, $CSDF$.
initialize the number of layers, $n = 1$;
initialize the number of layers not improving significantly,
$m = 0$;
divide $X_{raw}$ into $k$ subdatasets equally and stratifiedly,
$X = \{X_1, X_2 \cdots X_k\}$
while do
    if $n > 1$ then
        concatenate $F$ with $X_{raw}$ to get new data, $X$
    end
    get $F$, $c$, $L$ by Algorithm~3
    if $n = 1$ or $c' - c > p * c'$ then
        $n' = n$;
        $c' = c$
    end
    else
        $m + +$;
    end
    if $m = s$ then
        break
    end
    $n + +$.
end
concatenate first $n'$ layers to composes CSDF.
```

4. Experiments

In this section, we conduct experimentation to evaluate the performance of our method for price prediction on three different datasets. Particularly, Section 4.1 elaborates the details of experimentation and performance analysis on a dataset of car sharing, and Sections 4.2 and 4.3 evaluate the performance on the datasets of house renting and real estate selling, respectively.

Table 1
Features in price prediction for car sharing.

Feature	Explanation
rents	the number of transactions completed for a car
response	the response time of owner
age	the age of car
mile	the distance travelled of car, i.e., $< 2, 2-4, \ldots, > 20$
displacement	the displacement of car
extra-fee	the premium per kilometer over 300 kilometers
seats	the number of seats of car
GPS	whether there is a GPS navigator in car
MP3	whether MP3 connection is available
f2f	whether transaction is face to face
transmission	the transmission of car, i.e., automatic or manual
recommendation	whether is recommended by platform
order	whether order is automatically accepted
experience	the driving years of renter
town	whether driving out of town is allowed
weekday	when transactions happened, i.e., 1,2,. . . ,7
city	where transactions happened, e.g., Beijing, Guangzhou, . . .
gender	i.e., male, female, unknown
country	the country of car brand, e.g., China, Germany, America, . . .

4.1. Evaluation on the dataset of car sharing

The price dataset of car sharing is collected from a P2P car sharing platform, i.e., START Car Life[1], which ranges from Oct-16-2017 to Oct-29-2017 (excluding public holidays). This dataset consists of 116,145 items of car information from three cities in China. We select 19 attributes of the car as features (shown in Table 1), including both numerical values and texts, and we employ one-hot-encoding method to convert the latter into numerical values. To hinder small values from being overwhelmed by large values, we also apply normalization to the features. Fig. 2 shows the price distribution, and we can see that most of price values are lower than 1,000, while price values higher than 1,000 are sparsely distributed over a large range. Given that we consider the price prediction as a classification task in this paper, we apply the discretization methods to divide the price into intervals, which are further adopted as the labels. Moreover, 70% of the data are randomly and stratifiedly selected as training set and the remaining are used for testing.

4.1.1. Evaluation of discretization methods

In view of the price distribution in Fig. 2, we empirically set the number of intervals from 4 to 19, which are implemented with all mentioned discretization methods. Fig. 3(a) shows CVN and CVR of EPI, EOH[2] and KM, where KM is better than EPI and EOH, since both lower CVN and CVR are more desirable for a discretization method. Fig. 3(b) further demonstrates CVN and CVR of K-means and modified K-means. We can see that CVN and CVR of all modified K-means decrease as the number of intervals increases, and they all are smaller than that of K-means in most cases, which verified the effectiveness of the modified K-means. More specifically, regarding KMR, CVN and CVR slightly decrease when the number of intervals is between 9 and 16, and drop quickly for most of other numbers. This shows that KMR has the capability

[1] http://www.startcarlife.com/.

[2] Every One Hundred (EOH) is a variant of EWI, in which each hundred units is defined as an interval. The last interval is exceptional as it contains the rest of price values. In price prediction, EWI will assign most of samples into few intervals, and the remaining intervals will have much less examples, which will cause serious issue of class imbalance.

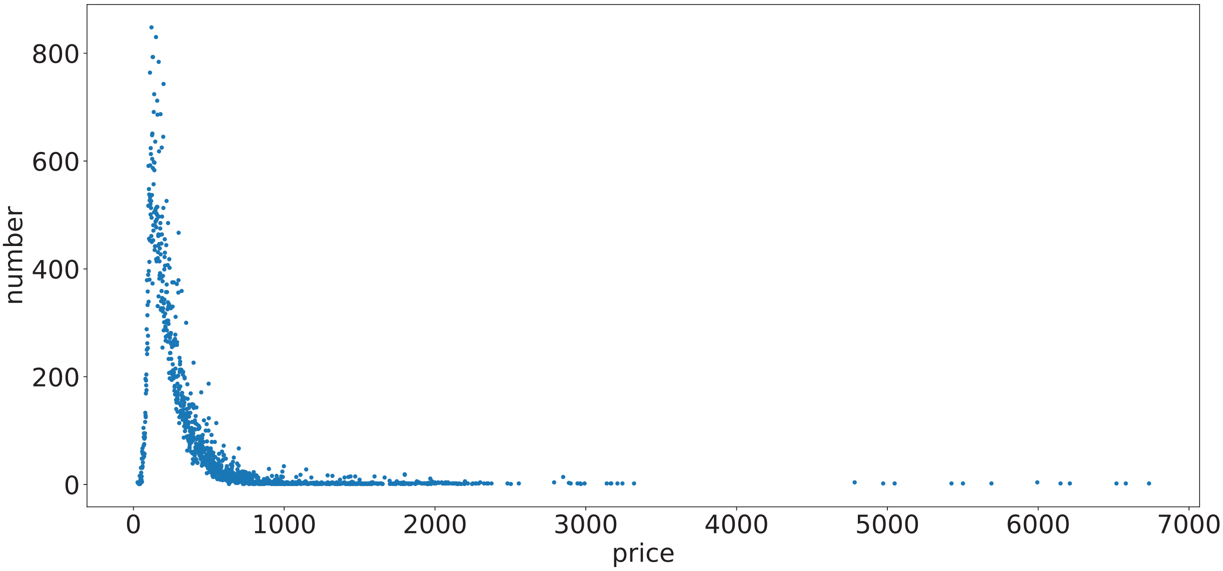


Fig. 2. Price distribution for dataset of car sharing, and *Y*-axis is the count of corresponding price.

of quickly reducing the variation levels for the number of samples and range of intervals, which further reduces the impact of outliers on K-means. And with the reduction of outliers, the uniform effect of K-means reduces CVN to be below 0.85 quickly. Regarding KMD, it achieves the best overall performance, especially with the number of intervals being 14. The improvement in KMD reduces the impact of outliers through the interval density, which relates to both the number of samples and range of interval. Regarding KMN, the trend of CVN and CVR is similar to that of KMD, but the values are slightly higher than that of KMD. In spite of slight discrepancy among the three modified K-means, CVN and CVR of them are significantly smaller than that of K-means in most cases, which also means that they are much superior to EPI and EOH.

4.1.2. Evaluation of traditional classification methods

Before verifying our approach, we first evaluate various types of baselines, including SVM, MLP and conventional random forest (RF). The price values are divided by six different discretization methods, i.e., {EOH, EPI, KM, KMR, KMN, KMD}, into 16 different numbers of classes (intervals), i.e., from 4 to 19. As such, there will be 96 classification tasks for each classification method. For SVM, we employed a linear kernel. For MLP, we adapted the hidden layers from 1 to 7, and finally chose 4 hidden layers and 100 units in each layer as they achieved the best results in most tasks. For RF, the number of trees is set as 10.

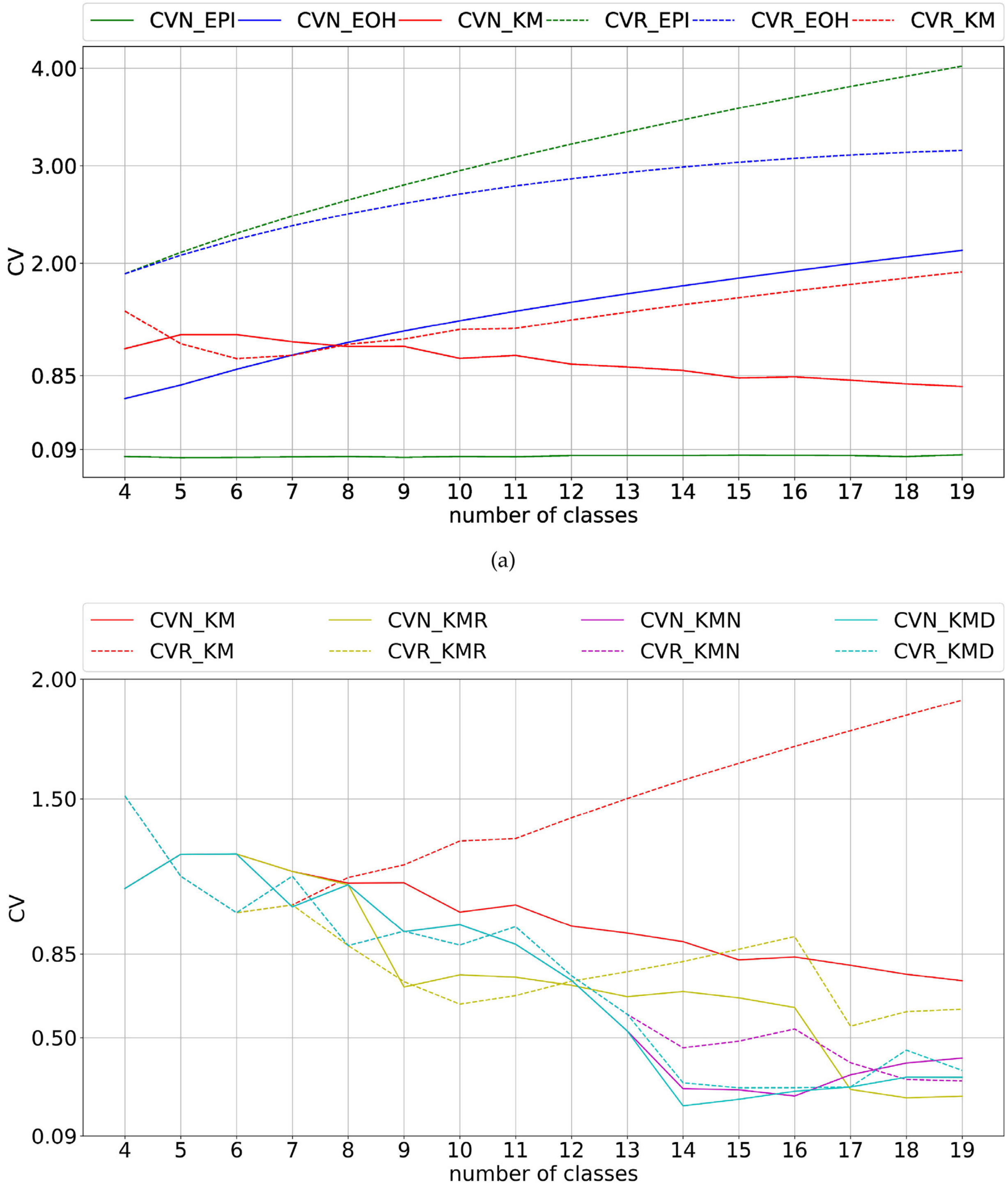
Fig. 4 shows the accuracy of SVM, MLP and RF with different discretization methods. We can see that for each discretization method, RF achieves the highest accuracy followed by MLP, and SVM yields the lowest accuracy. One of the reasons is that the dataset contains many categorical features, which can be more effectively handled by RF than others in general. Regarding different discretization methods, we can observe that EPI yields much lower accuracy compared with K-means, modified K-means and EOH. The reason is that EPI does not take into account the price distribution, which makes it problematic for the subsequent classification. Another observation is that EOH gets high accuracy when the number of intervals is large. This is because for EOH, most samples are discretized into the first few intervals, and the imbalanced classes will force the classifier to predict the sample as the interval with more samples, which results in high but meaningless accuracy. In spite of the discretization methods, RF always achieves the highest accuracy among the three types of baselines, which will be further adopted to compare with our approach. However, also given the analysis of CVN and CVR in Section 4.1.1, we will focus on the classification for K-means and modified K-means, rather than EPI and EOH.

4.1.3. Evaluation of deep forest and cost-sensitive deep forest

We evaluate our approach, i.e., cost-sensitive deep forest (CSDF), by comparing it with several strong baselines. Regarding deep forest (DF), the hyper-parameters include the type and number of base classifiers in each layer, the number of folds in the estimator, the degree of improvement, and the number of iterations for stopping. We adopted two random forests and two complete-random tree forests in each layer, and ten trees are employed in each forest, which is same as RF in Section 4.1.2. We use 5-fold cross validation in each estimator, and the degree of improvement is set as 0.01, which means that the improvement less than 1% is considered as insignificant. The number of iterations for stopping is set as three, meaning that if three successive layers have no significant improvement, DF will stop the iteration.

We first compare the strong baselines in Fig. 5, where we focus on K-means and modified K-means. Other than RF and DF, we also consider one more variant of RF, i.e., RoF (Random Rotation Forest) for the same dataset. From Fig. 5 we can observe that, the accuracy of DF and RoF are comparable in most cases. However, as the number of intervals increases from 13 to 19, DF with the three modified K-means is much better than RoF. Comparing with RF, DF is obviously superior. In particular, when the number of intervals is small, the accuracy of these two methods are similar. However, along with the increasing number of intervals, the gap between the accuracy of RF and DF becomes significantly larger. More specifically, the difference between the highest and lowest accuracy of RF is 21.3%, while that of DF is 15.7%. Based on these observations, we can see that DF can achieve superior performance, especially for larger number of classes. Since DF is principally adapted from RF rather than RoF, we will look into the performance comparison of DF and RF.

To further verify the improvement of accuracy achieved by DF over RF, we plot the improvements for different number of classes in Fig. 6(a). As shown, DF increases the accuracy of all K-means based discretization methods for all number of classes, with the largest improvement close to 10%. And the improvements are more



(a)

(b)

Fig. 3. (a) CVN and CVR (the lower, the better) of EPI, EOH and K-means; (b) CVN and CVR of K-means and modified K-means. In (b), some lines are overlapped with each other at the beginning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

obvious as the number of classes becomes larger. Comparing with RF in Fig. 5, we can observe that the trends of improvements in Fig. 6(a) roughly mirrored the accuracy drops of RF in Fig. 5. In other words, the lower accuracy of RF, the larger improvement of DF. Since RF is the base classifier of DF, the improvement achieved by DF might come from the diversity enhancement by ensemble learning, deep architecture and in-model feature transformation of deep learning. The diversity levels of different layers from the same DF are the same, so the difference in diversity levels comes from the number of layers in DF. Fig. 6(b) further shows the number of layers achieving the best results, which exhibits similar trends as those in Fig. 6(a), i.e., the more layers, the higher improvement. In addition, for the curves in Fig. 6(a) that have low improvements (roughly below 0.02), the corresponding numbers of layers in Fig. 6(b) are around 2. That happened might be due to the in-model feature transform, which is more helpful starting from the

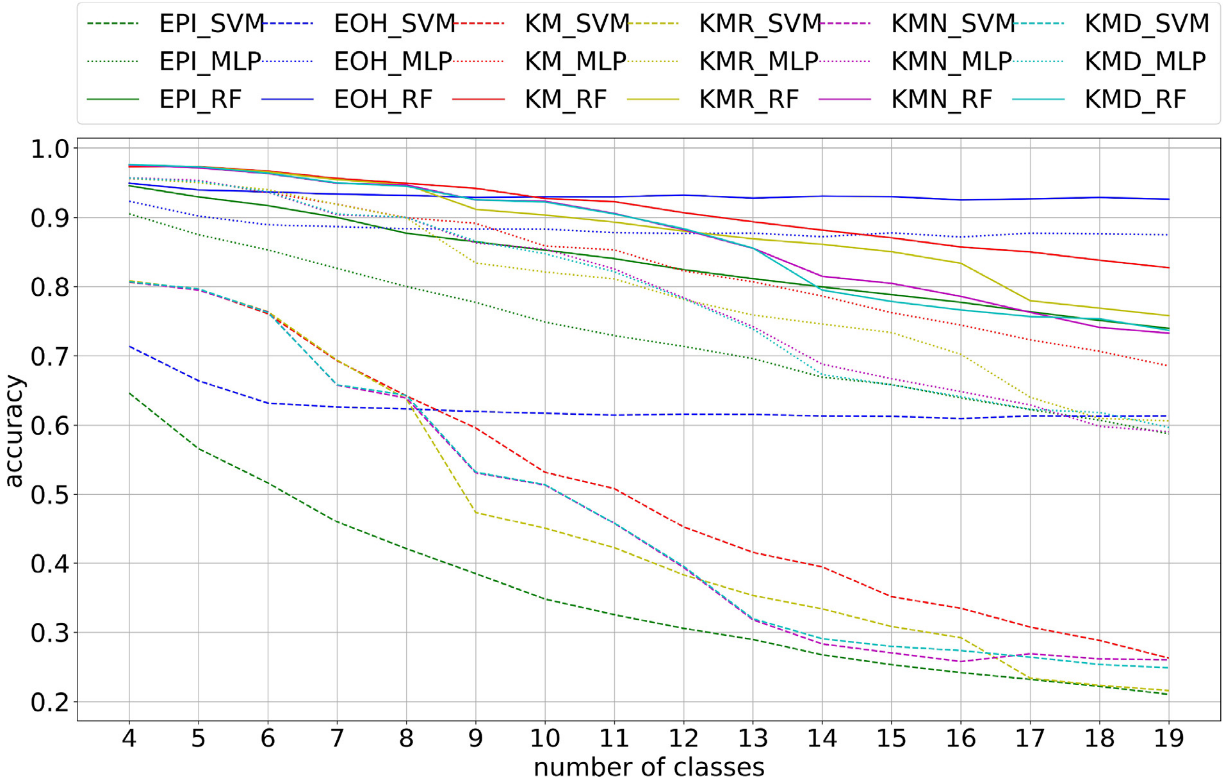


Fig. 4. Accuracy of SVM, MLP and RF with different discretization methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

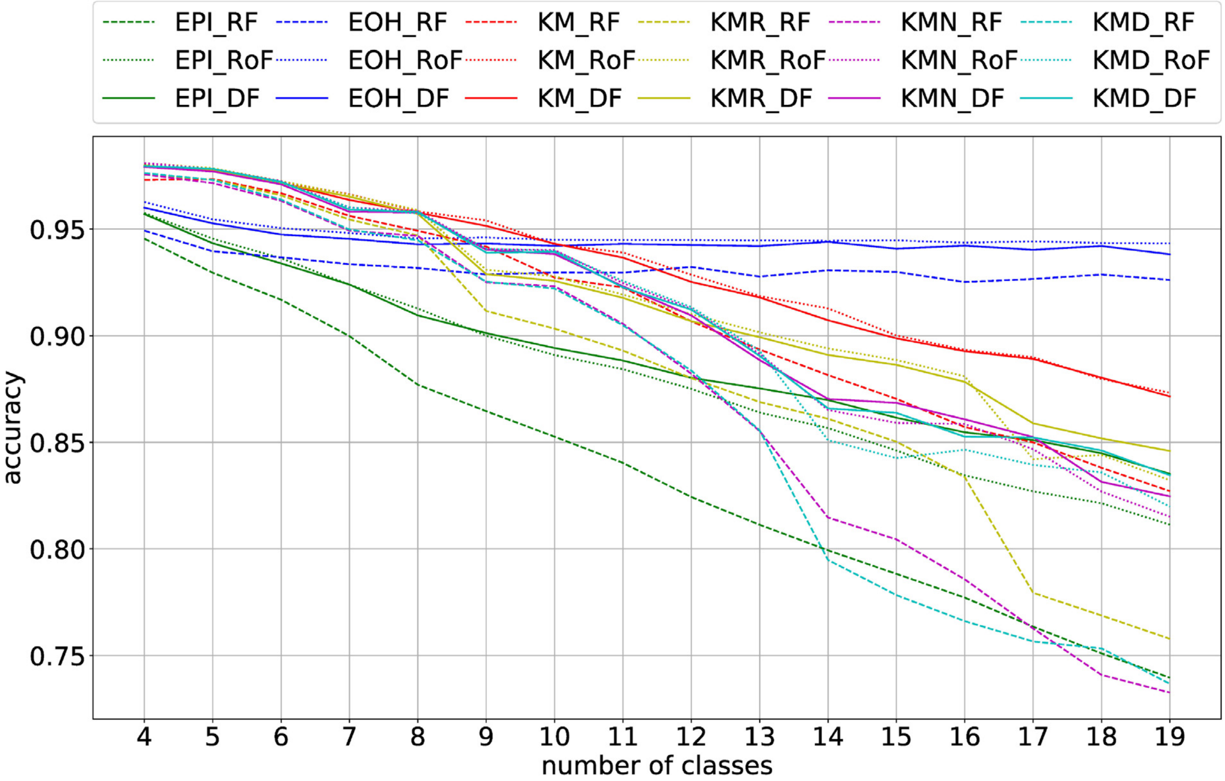
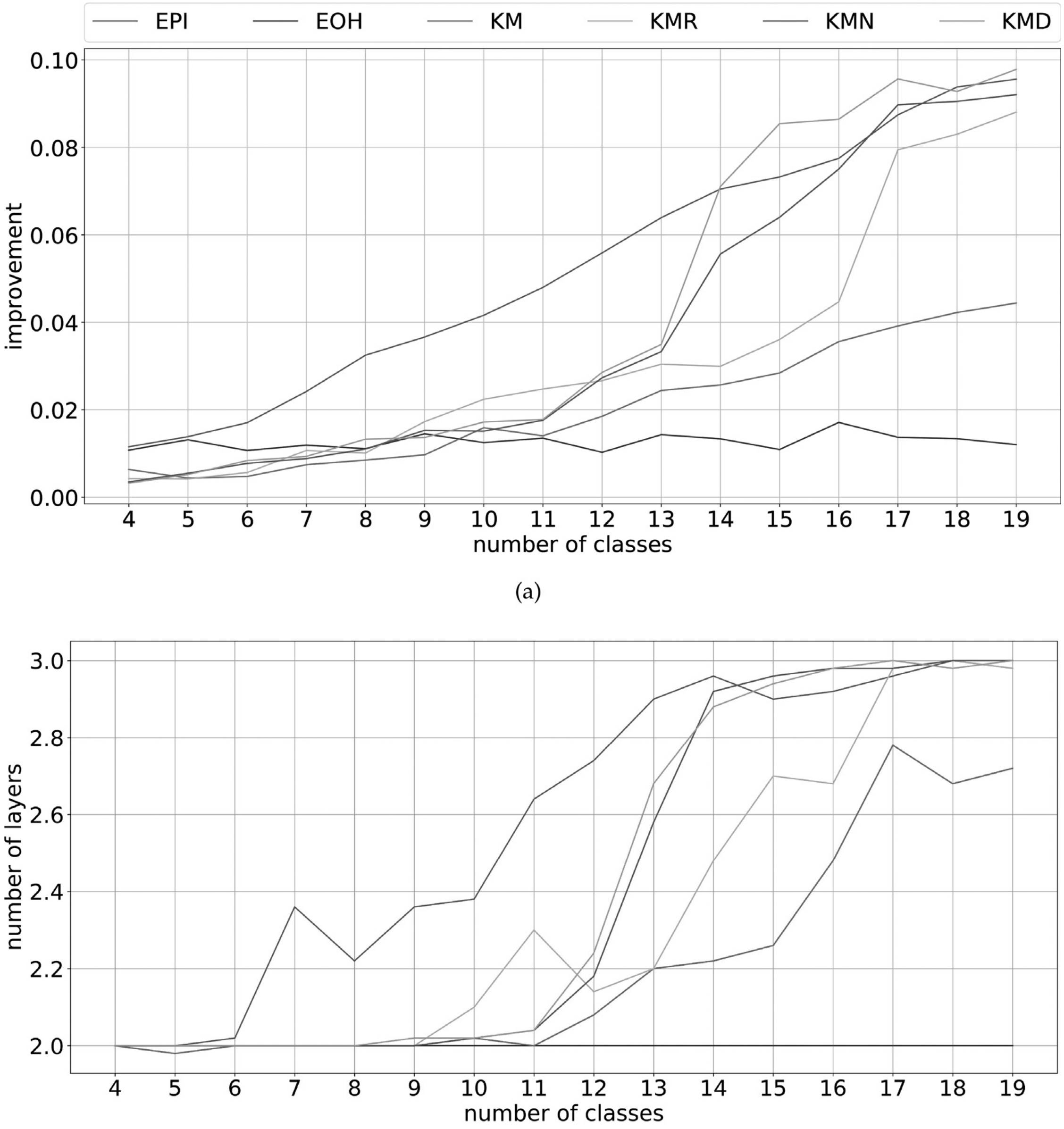


Fig. 5. Accuracy of RF, RoF and DF with different discretization methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

(b)

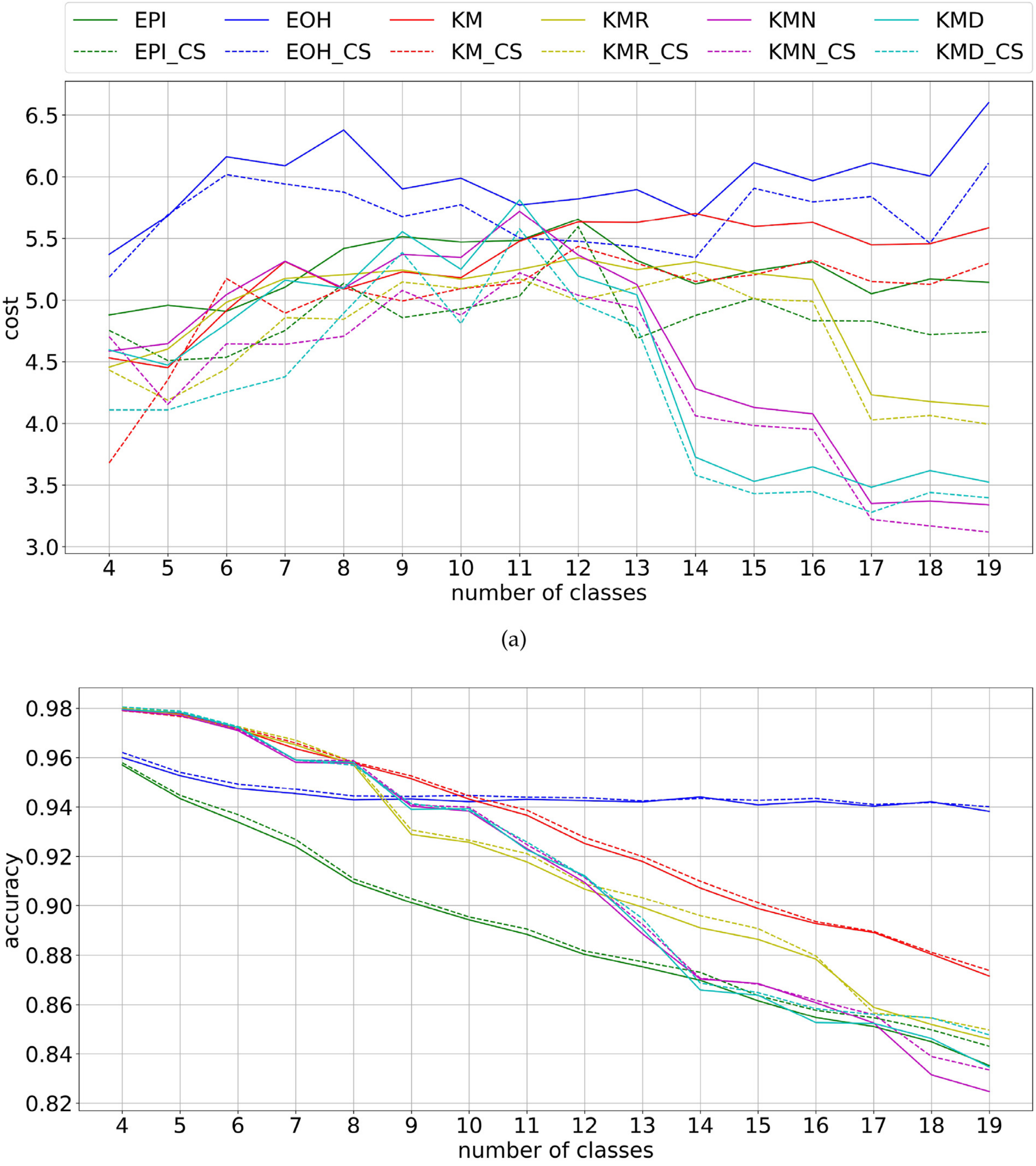
Fig. 6. (a) Improvement on accuracy of DF over RF; and (b) the number of layers achieving the best results in DF. Note that all methods are trained and tested for 30 times, and the average number of layers is not integer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
CVN and CVR of K-means and modified K-means on dataset of real estate selling.

	# classes	3	4	5	6	7	8	9	10	11	12
KM	CVN	0.706	0.724	0.657	0.735	0.813	0.700	0.737	0.650	0.618	0.526
	CVR	0.801	0.822	0.975	0.784	0.660	0.775	0.830	0.923	1.008	1.084
KMR	CVN	0.706	0.724	0.657	0.735	0.813	0.458	0.409	0.372	0.314	0.360
	CVR	0.801	0.822	0.975	0.784	0.660	0.357	0.388	0.438	0.481	0.491
KMN	CVN	0.706	0.724	0.657	0.735	0.813	0.458	0.409	0.372	0.314	0.360
	CVR	0.801	0.822	0.975	0.784	0.660	0.357	0.388	0.438	0.481	0.491
KMD	CVN	0.706	0.724	0.657	0.735	0.813	0.458	0.409	0.372	0.314	0.360
	CVR	0.801	0.822	0.975	0.784	0.660	0.357	0.388	0.438	0.481	0.491

second layer in the deep architectures. Based on these observations, we can basically conclude that the number of layers significantly influences the improvement.

Finally, we compare the performance of the proposed CSDF with DF. All hyper-parameters in CSDF are same as those in DF, and their cost is depicted in Fig. 7(a), respectively. As shown, most cost of CSDF is lower than that of DF for all discretization methods. More specifically, the average cost of CSDF is 5.6% lower than that of DF. We also observe that both methods get relatively small cost on modified K-means than on other discretization methods. Fig. 7(b) shows the accuracy of the proposed CSDF and DF, and we can see that the accuracy values of both methods drop as the



(a)

(b)

Fig. 7. (a) Cost (the lower, the better), and (b) accuracy of DF and CSDF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

number of classes increases. This is because as the range of intervals decreases, classifying an instance correctly may become relatively difficulty for any method. However, there is no significant difference of accuracy between CSDF and DF, both of which are superior, especially when comparing them with Figs. 4 and 5. Note that, taking Eq. (3) into account, the accuracy and cost may influence each other. Moreover, both accuracy and cost are also influenced by the discretization methods, so there might not be clear trend for the cost. Nevertheless, in spite of the trend, the observations demonstrate that CSDF can significantly reduce the classification cost of DF, and make the misclassification much closer to the true interval, while keeping almost same accuracy as DF.

4.2. Evaluation on the dataset of house renting

We continue to verify our approach on a price dataset of house renting, which is about the Airbnb house renting in United States[3]. The dataset we used includes 66,735 items of houses

[3] https://www.kaggle.com/stevezhenghp/airbnb-price-prediction.

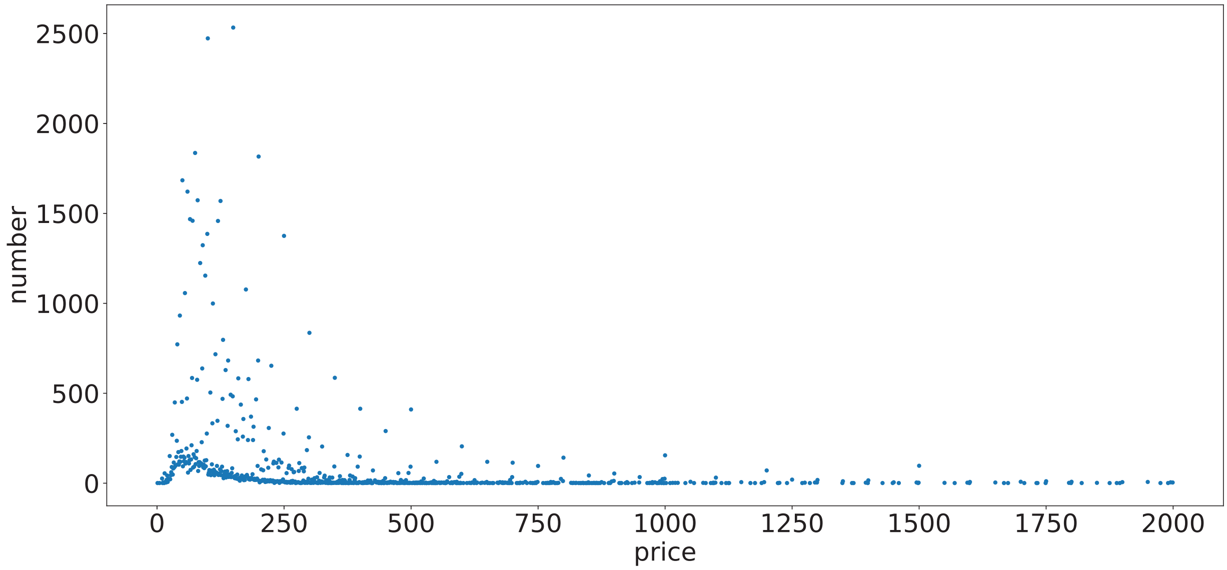


Fig. 8. Price distribution for dataset of house renting, and *Y*-axis is the count of corresponding price.

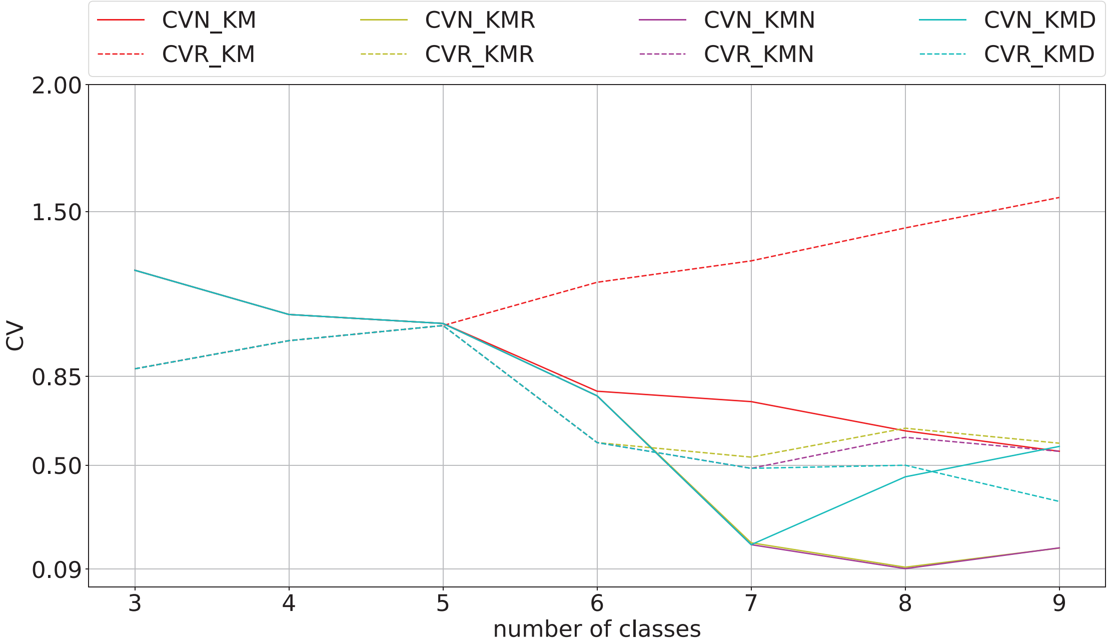


Fig. 9. CVN and CVR of K-means and modified K-means on dataset of house renting. Note: Some lines are overlapped with each other; and the smaller of the values, the better. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with 16 features, i.e., number of people that can be accommodated, number of bathrooms, whether cleaning fee applied, availability of profile pictures, whether host identity verified, whether instantly bookable, number of reviews, review scores rating, number of bedrooms, number of beds, city, cancellation policy, bed type, room type, property type, and neighbourhood. Same as previously, 70% of data items are randomly and stratifiedly selected as training set and the remaining 30% are used for testing. Fig. 8 shows the price distribution of the whole dataset, and we can observe that most price values are lower than 300, while price values higher than 300 are sparsely distributed over a large range.

Considering that the range of house price is about one third of the car sharing price, we empirically set the number of intervals (i.e., classes) of house renting price from 3 to 9. Firstly, we compare CVN and CVR of K-means and modified K-means, which are depicted in Fig. 9. From Fig. 9 we observe that, CVN and CVR of modified K-means are smaller than that of K-means, especially for number of intervals larger or equal to 6, which means that the modified K-means makes the intervals more uniform and balanced. Since CVR of K-means becomes significantly larger as the number of intervals increases, we will not adopt the conventional K-means as the discretization method for the subsequent classification. Secondly, we compare the accuracy of RF, RoF and DF as shown in Fig. 10. All accuracy values drop as the number of intervals increases, however, DF still gets the best performance with all modified K-means, especially for larger numbers of intervals. Finally, we continue to compare DF and CSDF regarding the cost and accuracy. As shown in Fig. 11(a), most of the cost of CSDF is lower than that of DF. Although this superiority is less significant on house renting data than that of car sharing data, the advantage regarding the cost might be more obvious as the number of intervals becomes

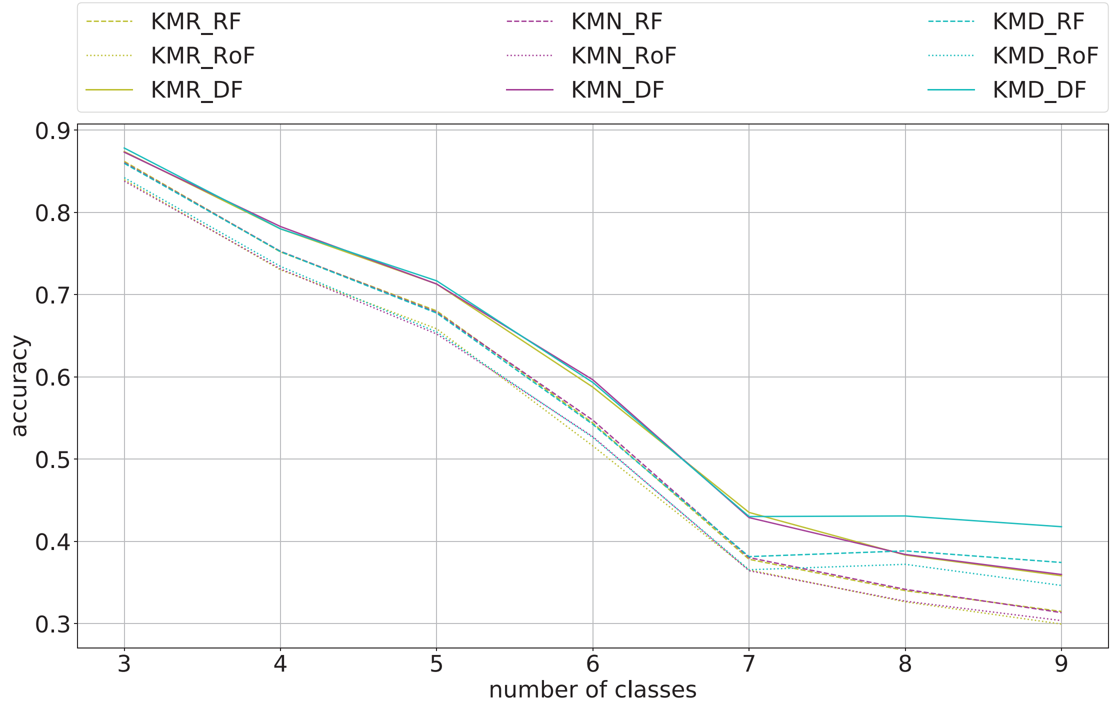


Fig. 10. Accuracy of RF, RoF and DF on dataset of house renting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

larger. However, we did not further increase the number of intervals up to the one in the car sharing data, because as mentioned previously, the price range is relatively smaller in the house renting data. Meanwhile, in Fig. 11(b), the accuracy of CSDF is almost the same as DF at the beginning, and becomes slightly higher than that of DF, as the number of intervals increases.

4.3. Evaluation on the dataset of real estate selling

We further evaluate our method on a price dataset of real estate selling, which includes 1,460 samples of real estates with 79 features[4]. Same as previously, 70% of the samples are randomly and stratifiedly selected as training set and the remaining 30% are used for testing. Fig. 12 shows the price distribution of the whole dataset, and we can observe that most price values are lower than 400,000. As we consider the price prediction as classification rather than regression, we empirically set the number of intervals (i.e., classes) of house selling data from 3 to 12.

We first evaluate K-means and modified K-means for discretization, the results of which are recorded in Table 2. We can observe that CVN and CVR of the four methods are same as the number of intervals is between 3 and 7. That is because almost no abnormal interval exists for small number of intervals, given the relative even distribution of the price. As the number of intervals continues to increase, CVN and CVR of modified K-means drop quickly, and the trend of K-means almost does not change. It implies the occurrence of abnormal intervals, while K-means is capability of capturing them. However, given the relative less intervals, CVN and CVR of the three modified K-means are the same for all cases. Therefore, we only choose one of them for the subsequent classification.

Then we compare the accuracy of RF, RoF, DF and CSDF for classification as shown in Fig. 13. All accuracy values drop as the number of intervals increases, however, DF and CSDF are comparable to each other, both of which achieve the best overall accuracy. Meanwhile, the accuracy values of RF and RoF are much lower than that of the two DF based methods, with RoF being slightly superior to RF, especially for larger numbers of intervals. Finally we verify the cost by comparing DF and CSDF. As shown in Table 3, expect for the number of intervals 11, all the cost of CSDF is lower than that of DF, which further justifies the effectiveness of CSDF in reducing the cost for price prediction.

Table 3
Cost of DF and CSDF.

#classes	3	4	5	6	7
DF	10894.5	12609.9	15199.0	14654.4	15882.2
CSDF	9634.4	11649.2	14446.0	14015.2	15512.4
#classes	8	9	10	11	12
DF	14976.6	15237.0	15084.1	15358.6	16158.0
CSDF	14945.3	15148.7	14906.6	15394.5	16030.9

To summarize, in comparison to the conventional methods, our CSDF achieved superior overall performance in price prediction for car sharing, house renting and real estate selling. With higher accuracy and lower cost for all the three tasks, the proposed CSDF also demonstrates strong capability of generalization.

5. Conclusion and future works

In this paper, we consider price prediction as a classification task in the sense of predicting the price interval of an item, which is practical for many applications. Traditional classification approaches only focus on the accuracy, and completely ignore the fact that different misclassifications have different costs, which makes them not suitable for price prediction problems. To resolve this issue, we proposed the cost-sensitive deep forest (CSDF), which explicitly considers the cost of misclassifications and there-

[4] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview.

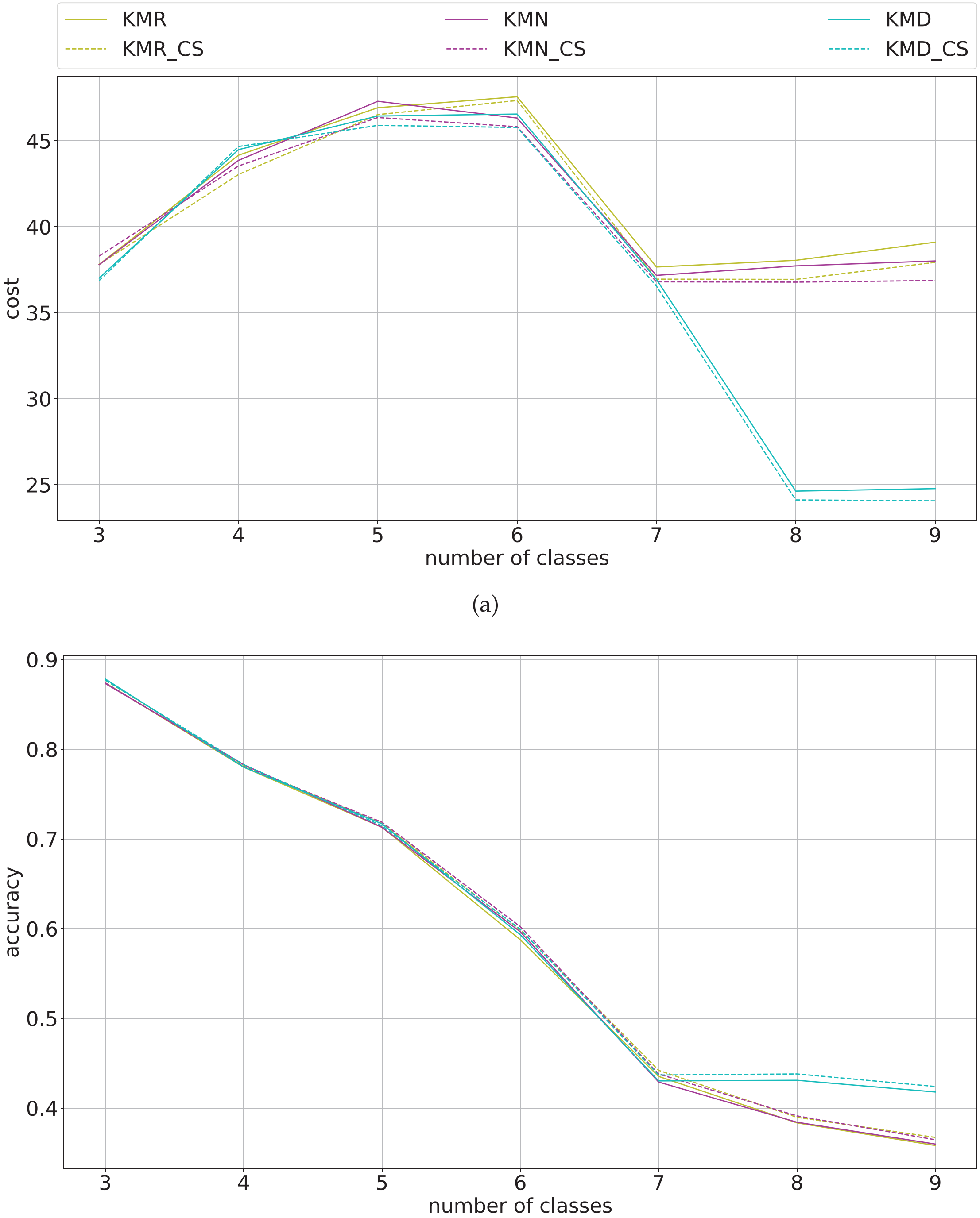


Fig. 11. (a) Cost (the lower, the better), and (b) accuracy of DF and CSDF on dataset of house renting. Note: some lines are overlapped with each other. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

fore can propel the wrong predictions to be closer to the corresponding real price intervals. We also proposed a modified K-means as the price discretization method, which explicitly considers the impact of outliers on the steadiness of prediction intervals compared with conventional K-means. In our experiments, owing to the cost-sensitive scheme we developed, CSDF results in less classification cost compared with the traditional deep forest on multiple price prediction tasks, while showing satisfying accuracy.

Besides the applications used in this paper (car sharing, house renting and real estate selling), CSDF also has the potential to be applied to other similar price prediction tasks, which we plan to demonstrate in the future by performing experiments on other datasets. We will also investigate other techniques to further enhance our method, for example using the intuitionistic fuzzy set [43] as a measurement to evaluate the decision making for price prediction.

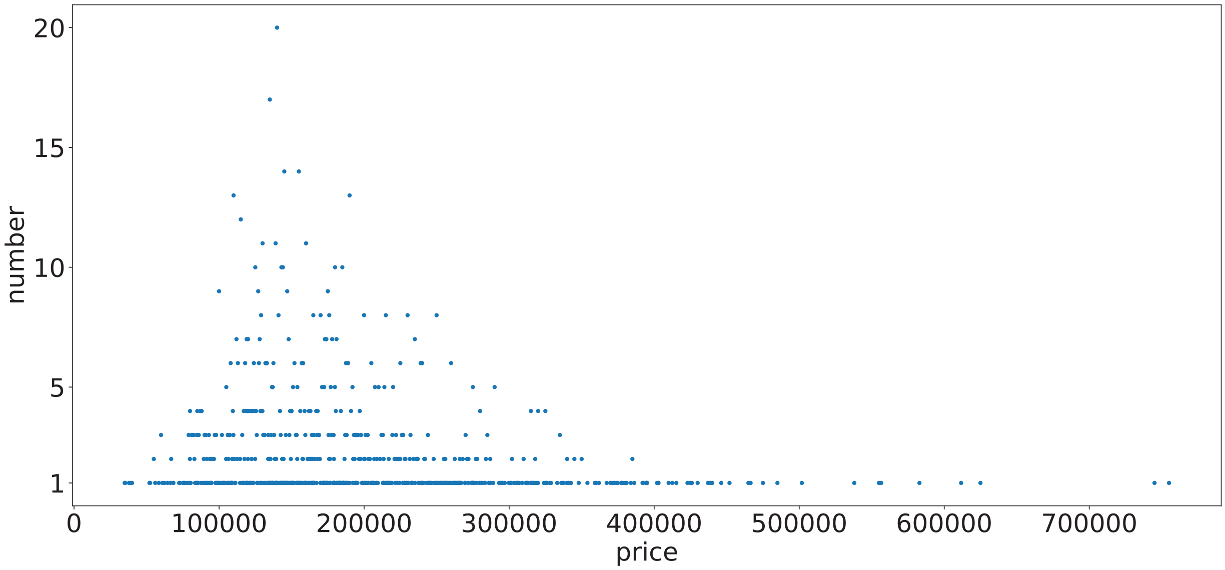


Fig. 12. Price distribution for dataset of real estate selling, and *Y*-axis is the count of corresponding price

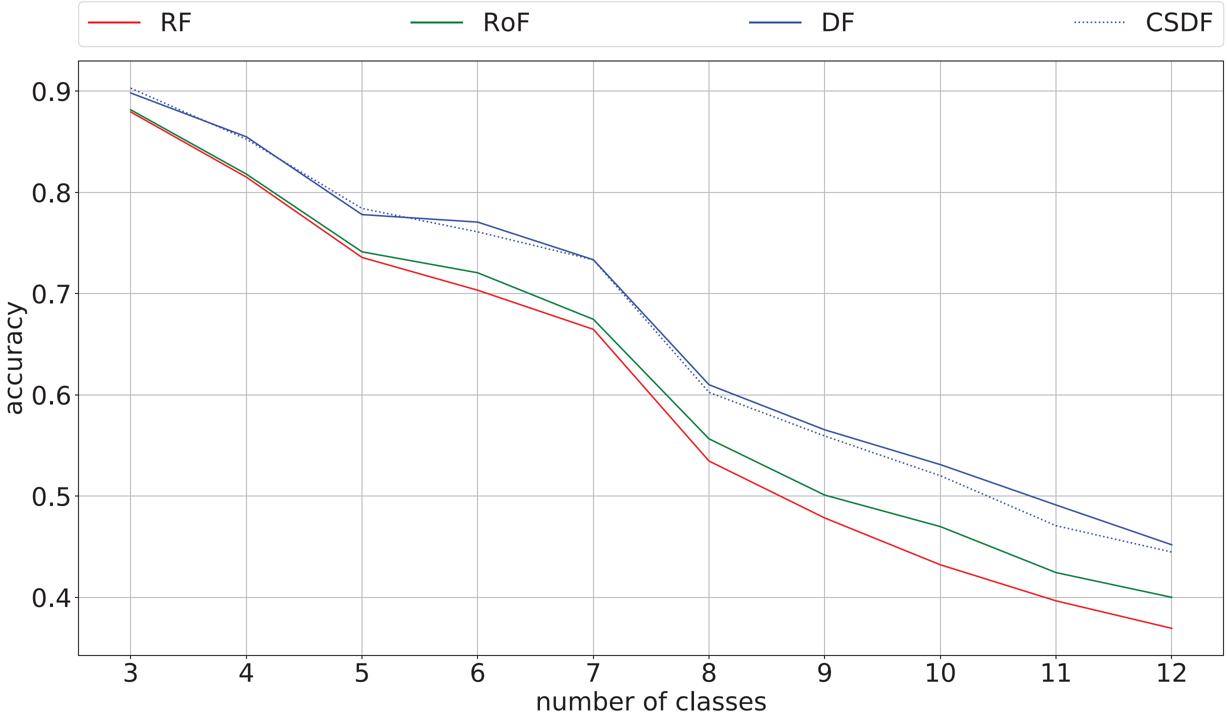


Fig. 13. Accuracy of RF, RoF, DF and CSDF with modified K-means. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Acknowledgement

This work is supported by the BMW@NTU Future Mobility Research Lab, Singapore; National Natural Science Foundation of China (61803104, 61803100, 61562013, 61866009, 61971369, U19B2031, 81671766); Open Fund of Science and Technology on Automatic Target Recognition Laboratory (6142503190202); Guangxi Natural Science Foundation (2017GXNFDA198025); the study abroad program for graduate student of Guilin University of Electronic Technology (GUET); and MOE Academic Research Fund of Singapore (R266000096133, R266000096731 and MOE2017-T2-2-153) and Singapore National Research Foundation (NRF-RSS2016004). Dr. Wen Song is supported by the Young Scholar Future Plan of Shandong University (62420089964188).

References

[1] J. Zhang, S. Cui, Y. Xu, Q. Li, T. Li, A novel data-driven stock price trend prediction system, Expert Syst. Appl. 97 (2018) 60–69.

[2] L. Zhang, C. Aggarwal, G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in: Proceedings of the 23rd ACM International Conference on Knowledge Discovery and Data Mining, 2017, pp. 2141–2149.

[3] Z. Cen, J. Wang, Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer, Energy 169 (2019) 160–171.

[4] M. Bicego, A. Farinelli, E. Grosso, D. Paolini, S.D. Ramchurn, On the distinctiveness of the electricity load profile, Pattern Recognit. 74 (2018) 317–325.

[5] X. Qiu, P.N. Suganthan, G.A.J. Amaratunga, Short-term electricity price forecasting with empirical mode decomposition based ensemble kernel machines, Procedia Comput. Sci. 108 (2017) 1308–1317.
[6] H. Crosby, P. Davis, T. Damoulas, S.A. Jarvis, A spatio-temporal, gaussian process regression, real-estate price predictor, in: Proceedings of the 24th ACM International Conference on Advances in Geographic Information Systems, 2016, pp. 1–4.
[7] K. Tziridis, T. Kalampokas, G.A. Papakostas, K.I. Diamantaras, Airfare prices prediction using machine learning techniques, in: Proceedings of the 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 1036–1039.
[8] I. Ballús-Armet, S. Shaheen, K. Clonts, D. Weinzimmer, Peer-to-peer carsharing: exploring public perception and market characteristics in the san francisco bay area, california, Transp. Res. Rec. 2416 (4) (2014) 27–36.
[9] Z. Cao, H. Guo, J. Zhang, D. Niyato, U. Fastenrath, Finding the shortest path in stochastic vehicle routing: a cardinality minimization approach, IEEE Trans. Intell. Transp. Syst. 17 (6) (2016) 1688–1702.
[10] Z. Cao, S. Jiang, J. Zhang, H. Guo, A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion, IEEE Trans. Intell. Transp. Syst. 18 (7) (2017) 1958–1973.
[11] S. Le Vine, A. Zolfaghari, J. Polak, Carsharing: Evolution, Challenges and Opportunities, Technical Report, European Automobile Manufactures Association, 2014.
[12] J. Huang, L. Zhu, B. Fan, Y. Chen, W. Jiang, S. Li, Large-scale price interval prediction at OTA sites, IEEE Access 6 (2018) 69807–69817.
[13] Z. Zhou, J. Feng, Deep forest: towards an alternative to deep neural networks, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 3553–3559.
[14] L. Torgo, J. Gama, Regression using classification algorithms, Intell. Data Anal. 1 (1–4) (1997) 275–292.
[15] L. Tian, F. Cao, L. Yu, An initial cluster center selection algorithm for categorical matrix data, Comput. Eng. 46 (5) (2020) 115–121.
[16] Y.Y. Webb, I. Geoffrey, Discretization for naive-Bayes learning: managing discretization bias and variance, Mach. Learn. 74 (1) (2009) 39–74.
[17] D.V. den Poel, Book review: ensemble methods: foundations and algorithms, IEEE Intell. Inform. Bull. 13 (1) (2012) 33–34.
[18] F.T. Liu, K.M. Ting, Y. Yu, Z.-H. Zhou, Spectrum of variable-random trees, J. Artif. Intell. Res. 32 (1) (2008) 355–384.
[19] L. Zhang, P.N. Suganthan, Benchmarking ensemble classifiers with novel co-trained kernel ridge regression and random vector functional link ensembles [research frontier], IEEE Comput. Intell. Magaz. 12 (4) (2017) 61–72.
[20] L. Zhang, P.N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE Trans. Cybern. 45 (10) (2014) 2165–2176.
[21] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman and Hall/CRC, 2012.
[22] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[23] I. Barandiaran, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach.Intell. 20 (8) (1998) 832–844.
[24] L. Breiman, Randomizing outputs to increase prediction accuracy, Mach. Learn. 40 (3) (2000) 229–242.
[25] Y. Lecun, Y. Bengio, G. Hinton, Deep learning., Nature 521 (7553) (2015) 436–444.
[26] X.Z. Wang, T. Zhang, R. Wang, Noniterative deep learning: Incorporating restricted Boltzmann machine into multilayer random weight neural networks, IEEE Trans. Syst. Man Cybern.Syst. 49 (7) (2017) 1299–1308.
[27] M. Zhou, X. Zeng, A. Chen, Deep forest hashing for image retrieval, Pattern Recognit. 95 (2019) 114–127.
[28] C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2001, pp. 973–978.
[29] P. Cao, D. Zhao, O. Zaiane, An optimized cost-sensitive SVM for imbalanced data learning, in: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2013, pp. 280–292.
[30] C.L. Castro, A.P. Braga, Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data, IEEE Trans. Neural Netw. Learn.Syst. 24 (6) (2013) 888–899.
[31] F. Yang, H.-z. Wang, H. Mi, W.-w. Cai, et al., Using random forest for reliable classification and cost-sensitive learning for medical diagnosis, BMC Bioinform. 10 (1) (2009) S22.
[32] H. Lu, L. Yang, K. Yan, Y. Xue, Z. Gao, A cost-sensitive rotation forest algorithm for gene expression data classification, Neurocomputing 228 (228) (2017) 270–276.
[33] J.M. Gómez Hidalgo, M.M. López, E.P. Sanz, Combining text and heuristics for cost-sensitive spam filtering, in: Proceedings of the 4th Conference on Computational Natural Language Learning, 2000, pp. 99–102.
[34] Y. Zhang, Z.-H. Zhou, Cost-sensitive face recognition, IEEE Trans. Pattern Anal. Mach.Intell. 32 (10) (2010) 1758–1769.
[35] Y. Li, A. Santorelli, O. Laforest, M. Coates, Cost-sensitive ensemble classifiers for microwave breast cancer detection, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 952–956.
[36] A.C. Bahnsen, D. Aouada, B. Ottersten, Example-dependent cost-sensitive decision trees, Expert Syst. Appl. 42 (19) (2015) 6609–6619.
[37] Z. Liu, C. Ma, C. Gao, H. Yang, R. Lan, X. Luo, Cost-sensitive collaborative representation based classification via probability estimation with addressing the class imbalance, Multimed. Tools Appl. 77 (9) (2018) 10835–10851.
[38] P.-Y. Yang, Z.-R. Lai, X. Wu, L. Fang, Trend representation based log-density regularization system for portfolio optimization, Pattern Recognit. 76 (2018) 14–24.
[39] J. Wu, H. Xiong, J. Chen, W. Zhou, A generalization of proximity functions for k-means, in: Proceedings of the 7th IEEE International Conference on Data Mining (ICDM), 2007, pp. 361–370.
[40] F.T. Liu, K.M. Ting, Z. Zhou, Isolation forest, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), 2008, pp. 413–422.
[41] H. Mhaskar, Q. Liao, T.A. Poggio, When and why are deep networks better than shallow ones? in: Proceedings of Conference on Association for the Advancement of Artificial Intelligence (AAAI), 2017, pp. 2343–2349.
[42] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in: Advances in Neural Information Processing Systems (NIPS), 1995, pp. 231–238.
[43] G.A. Papakostas, A.G. Hatzimichailidis, V.G. Kaburlasos, Distance and similarity measures between intuitionistic fuzzy sets: a comparative analysis from a pattern recognition point of view, Pattern Recognit. Lett. 34 (14) (2013) 1609–1622.

Chao Ma is currently a graduate student in Guilin University of Electronic Technology, China. He received the B.Eng degree in Henan University of Science and Technology, China, in 2015, and he was a visiting scholar with BMW@NTU Future Mobility Research Lab, Singapore, in 2018. His research interests include machine learning and intelligent systems.

Zhenbing Liu received Ph.D. in Institute of Image Recognition and Artificial Intelligence at Huazhong University of Science and Technology in 2010 and was a visiting scholar in the University of Pennsylvania. Now he is professor in Guilin University of Electronic Technology. His research interests include machine learning and pattern recognition.

Zhiguang Cao received Ph.D from Interdisciplinary Graduate School, Nanyang Technological University, Singapore, 2016. He is currently a Research Assistant Professor with the Department of Industrial Systems Engineering and Management, National University of Singapore. His research interests include the application of AI and Optimization to transportation optimization.

Wen Song received the B.S. degree in automation and the M.S. degree in control science and engineering from Shandong University, China, in 2011 and 2014, respectively, and the Ph.D. degree in computer science from the Nanyang Technological University, Singapore, in 2018. He hold a position of postdoc researcher in the Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU) after Ph.D graduation. He is currently an Associate Research Fellow with the Institute of Marine Science and Technology, Shandong University, China. His research interests include artificial intelligence, planning and scheduling, multi-agent systems, and operations research.

Jie Zhang is currently an Associate Professor of the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He obtained Ph.D. in Cheriton School of Computer Science from University of Waterloo, Canada, in 2009. His research interests include Artificial Intelligence, Multi-Agent System, User Modeling and Trust, Preference and Incentive.

Weiliang Zeng received the B.Sc. degree and the M.Sc. degree in 2009 and 2012 from School of Engineering, Sun Yat-sen University, China, and received the Ph.D. degree in Department of Civil Engineering, Nagoya University, Japan. Now he is an associate professor in Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, China. His research interests include sharable autonomous vehicle, route planning, machine learning, traffic simulation, and transportation dispatch system.