Data-Centric Computing at the Edge for Internet of Things - Research Article

International Journal of Distributed Sensor Networks

International Journal of Distributed Sensor Networks
2022, Vol. 18(6)

DOI: 10.1177/15501329221106935
journals.sagepub.com/home/dsn
SAGE

Research and application of XGBoost in imbalanced data

Ping Zhang, Yiqiao Jia and Youlin Shang

Abstract

As a new and efficient ensemble learning algorithm, XGBoost has been widely applied for its multitudinous advantages, but its classification effect in the case of data imbalance is often not ideal. Aiming at this problem, an attempt was made to optimize the regularization term of XGBoost, and a classification algorithm based on mixed sampling and ensemble learning is proposed. The main idea is to combine SVM-SMOTE over-sampling and EasyEnsemble under-sampling technologies for data processing, and then obtain the final model based on XGBoost by training and ensemble. At the same time, the optimal parameters are automatically searched and adjusted through the Bayesian optimization algorithm to realize classification prediction. In the experimental stage, the G-mean and area under the curve (AUC) values are used as evaluation indicators to compare and analyze the classification performance of different sampling methods and algorithm models. The experimental results on the public data set also verify the feasibility and effectiveness of the proposed algorithm.

Keywords

XGBoost, imbalanced data, sampling technology, ensemble method, machine learning

Date received: 26 August 2021; accepted: 11 May 2022

Handling Editor: Yanjiao Chen

Introduction

With the breakthrough and innovation of science and technology, we are now in the cloud era of massive data. In the face of complicated and expansive data, appropriate analysis methods must be adopted to explore its potential value. In recent years, research fields such as data mining, artificial intelligence, and machine learning have developed strongly. As one of the important contents, classification technology has also become the research focus of scholars.

Data imbalance refers to the uneven distribution of samples in each category in the data set. The data set can be divided into the majority class (negative class) and the minority class (positive class) according to the sample size.[1] The classification problem of imbalanced data exists in many aspects of life, such as medical diagnosis, information security, text mining and target detection, and so on.[2] Because the current traditional classification algorithms aim at maximizing the overall accuracy and are established based on the premise that the data distribution of each category is relatively balanced, the misclassification rate for minority samples is high, and they cannot be applied maturely and stably in the problem of data imbalance.[3] Therefore,

School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang, China

Corresponding author:
Ping Zhang, School of Mathematics and Statistics, Henan University of Science and Technology, No. 263 Kaiyuan Avenue, Luolong District, Luoyang 471023, Henan, China.
Email: zping@haust.edu.cn

how to obtain a more accurate and ideal classification effect on imbalanced data sets is an urgent problem to be solved, which has great practical significance and application value.

Comprehensively, the research work of scholars at home and abroad for the classification of imbalanced data over the years has mainly focused on three levels: data, algorithm and evaluation index.[4] The core of data-level research is to reconstruct and adjust the sample distribution of the original data set in various ways to reduce or eliminate its imbalance. The main methods include data resampling and feature selection.[5] The key of algorithm-level research is to improve the imbalanced limitations of traditional classification algorithms, which mainly including cost-sensitive learning and ensemble learning methods. When dealing with imbalanced data, cost-sensitive learning will complete the distinction by setting different misclassification costs for various samples, which is more flexible, but there is a risk of overfitting, and the learning cost is high. The ensemble learning method mainly combines traditional algorithms with other improved algorithms. For example, Chawla et al.[6] proposed the SMOTEBoost algorithm combining the classical over-sampling algorithm synthetic minority over-sampling technique (SMOTE) and AdaBoost algorithm, which improves the generalization ability. The evaluation index level is mainly aimed at exploring and optimizing the classification algorithm index. For example, Cheng et al.[7] tried to improve the *F*-value using the support vector machine (SVM) of cost-sensitive learning.

The extreme gradient boosting algorithm XGBoost[8] is an ensemble learning algorithm with the advantages of high flexibility, strong predictability, strong generalization ability, high scalability, high model training efficiency, and great robustness. The current research work on XGBoost mainly focuses on direct application,[9–14] integration with other algorithms,[15–18] and parameter optimization.[19–21] In terms of imbalanced data research, Jia[22] combined the improved SMOTE algorithm of clustering with XGBoost, and applied ensemble learning to realize the abnormal detection of bolt process. Cui[23] combined the EasyEnsemble under-sampling algorithm with XGBoost, and comprehensively used most classes of sample information for classification and prediction. However, due to the diversity of imbalanced data distribution, the effect of combining over-sampling and under-sampling methods with XGBoost is often not ideal.

In view of this, the primary goal of this article is to optimize and improve the classification performance of XGBoost in case of data imbalanced, and an XGBoost classification algorithm combining mixed sampling technology and ensemble learning is proposed. First, at the data level, SVM–SMOTE and EasyEnsemble are used to reduce the imbalance of data. Then, at the algorithm level, XGBoost is used to train the generative model, and the Bayesian optimization algorithm is used to automatically search the optimal parameters. By analyzing the experimental results, it can be seen that the classification model proposed in this article has a better effect than the other three representative classification models (RUSBoost,[24] CatBoost,[25] and LightGBM[26]) and the XGBoost algorithm based on mixed sampling designed by Yue.[27]

Related algorithms

SVM–SMOTE

As a classic over-sampling algorithm with universal applicability, SMOTE[28] realizes data synthesis based on random interpolation by selecting the nearest neighbor distance between samples, expands the feature decision area of minority samples, and can effectively balance data. However, it also has problems such as low quality of synthetic samples, fuzzy class boundary, and uneven distribution of a few samples.[29]

The subsequent improved algorithms mainly include Borderline-SMOTE, ADASYN, and SVM–SMOTE. Among them, Borderline-SMOTE and ADASYN mainly solve the quality problem of the generated samples, while SVM–SMOTE divides the minority samples into safe (more than half of the nearest neighbor samples belong to the minority class), dangerous (more than half of the nearest neighbor samples belong to the majority class), and noise (all the nearest neighbor samples belong to the majority class), and then training SVMs based on dangerous samples, using the support vectors found by SVM to generate new samples close to the boundary of most classes and a few classes, giving full play to the advantages of SVM algorithm in boundary decision-making, and can solve the problems of generated sample quality and class boundary ambiguity at the same time. It is an over-sampling method with good effect.

EasyEnsemble

Easyensemble[30] is a hybrid ensemble under-sampling algorithm. It uses Bagging to fuse random down sampling and AdaBoost algorithm, which makes up for the defect that the general under-sampling algorithm may lose important classification information. The AdaBoost algorithm selected by the base classifier also improves the classification accuracy and generalization ability. Different from the supervised combination of another representative algorithm, BalanceCascade, EasyEnsemble is based on unsupervised under-sampling. It has the advantages of low-time complexity and high utilization of data, which can greatly avoid the waste of limited data resources. It is an effective and

Table 1. Algorithm steps of EasyEnsemble.

Input: training set S, the number of weak learners of AdaBoost (M), the learning algorithm of AdaBoost (L)
1. Divide the training set S to obtain the minority sample set P and the majority sample set N, $|P|<|N|$, and select T subsets from N;
2. For $i = 1 : T$
Randomly select subset N_i from N so that $|N_i| = |P|$; let $S_i = N_i \cup P$, train an AdaBoost learner H_i constructed by M weak learners on S_i, record the weight $w_{i,j}$ of each weak learner $L_{i,j}$ and the ensemble decision threshold θ_i, that is

$$H_i(x) = \operatorname{sgn}\left(\sum_{j=1}^{M} w_{i,j}L_{i,j}(x) - \theta_i\right).$$

End

Output: the prediction result of the test sample is $H(x) = \operatorname{sgn}\left(\sum_{i=1}^{T}\sum_{j=1}^{M} w_{i,j}L_{i,j}(x) - \sum_{i=1}^{T}\theta_i\right)$

more expansive method. The principle steps of EasyEnsemble are shown in Table 1.

XGBoost

For a given training set with n examples and m features, $D = \{(x_i, y_i)\}_{i=1}^{n}$ $(|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R})$. XGBoost can be regarded as an additive model $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$ which composed of K CART trees. Among them, $f_k(x_i)$ represents the predicted value obtained after inputting the ith sample x_i into the kth tree, $\hat{y}_i$ represents the final prediction result of x_i, and F is the set space of all regression trees. The objective function of XGBoost can be defined as $Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$. Among them, y_i is the real result and $\sum_{i=1}^{n} l(y_i, \hat{y}_i)$ is the loss function, which can measure the prediction ability of the model; $\sum_{k=1}^{K} \Omega(f_k)$ is the regularization term of the model, used to control the complexity and avoid overfitting.

The modeling process of XGBoost is to leave the original model unchanged, and set the input of the next tree as the residual of $\hat{y}_i$ and y_i. The general steps are as follows:

Initialize $\hat{y}_i^{(0)} = 0$, we can get

$$\begin{aligned}
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)\\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)\\
&\cdots\\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

In the above formula, $\hat{y}_i^{(t)}$ and $\hat{y}_i^{(t-1)}$, respectively, represent the predicted value of the model during the tth and previous $t-1$ iterations of x_i, and $f_t(x_i)$ is the newly added prediction function in each round, so the objective function of the tth iteration is

$$\begin{aligned}
Obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k)\\
&= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{k=1}^{t-1} \Omega(f_k) + \Omega(f_t)
\end{aligned}$$

Since $\hat{y}_i^{(t-1)}$ and $l(y_i, \hat{y}_i^{(t-1)})$ are both constants, the second-order Taylor expansion can be used for the loss function to approximate the objective function. Let $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$, omitting the constant term, we have

$$Obj^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \qquad (1)$$

In XGBoost, the number of leaf nodes T and the weight w of the tree are considered to define the complexity of the tree, that is, $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$. Among them, γ and λ are the parameters controlling complexity. The larger the value, the more complex the structure of the tree.

In general, formula (1) can be rewritten as the following format

$$\begin{aligned}
Obj^{(t)} &\approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2\\
&= \sum_{j=1}^{T} \left[\left(\sum_{i\in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right) w_j^2\right] + \gamma T
\end{aligned} \qquad (2)$$

where $I_j = \{i | q(x_i) = j\}$ is the sample on the jth leaf node, and w_j is the weight of the jth node. Let $G_j = \sum_{i\in I_j} g_i$ and $H_j = \sum_{i\in I_j} h_i$, then there is

$$Obj^{(t)} \approx \sum_{j=1}^{T} \left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right) w_j^2\right] + \gamma T$$

Assuming that the tree structure has been determined, the optimal solution obtained by directly deriving w_j is $w_j^* = -(G_j/H_j + \lambda)$. Substituting this solution into equation (2), the optimal value of the objective function is

$$Obj^* = -\frac{1}{2}\sum_{j=1}^{T}\left(\frac{G_j^2}{H_j + \lambda}\right) + \gamma T$$

XGBoost calculates the gain value through the formula

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

and selects the feature with the largest corresponding gain value and the split point under this feature for splitting.

Among them, $(G_L^2/H_L + \lambda)$ and $(G_R^2/H_R + \lambda)$, respectively, represent the structure score of the left and right subtrees, and $((G_L + G_R)^2/H_L + H_R + \lambda)$ is the structure score when the current node is not split. The rest of the detailed algorithm flow can be found in the literature.[8]

Algorithm optimization and design

Regularization term optimization

Because the L_1 regularization term has stronger anti-noise ability and robustness, but there may be multiple optimal solutions, and the weight coefficients will be sparse, while the L_2 regularization term has lower computational complexity and faster speed. Therefore, this article refers to the idea of ElasticNet,[31] combines L_1 and L_2 regularization terms, and redefines the complexity of the tree model, which is

$$\Omega(f_t) = \frac{1}{2}\lambda\sum_{j=1}^{T}(w_j - 1)^2 = -\lambda\sum_{j=1}^{T}w_j + \frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 + \frac{1}{2}\lambda T$$

then formula (2) becomes

$$Obj^{(t)} \approx \sum_{j=1}^{T}\left[(G_j - \lambda)w_j + \frac{1}{2}(H_j + \lambda)w_j^2\right] + \frac{\lambda}{2}T \quad (3)$$

The optimal solution of the derivative solution is $w_j' = -(G_j - \lambda/H_j + \lambda)$, then the optimal value of the objective function is

$$Obj' = -\frac{1}{2}\sum_{j=1}^{T}\left(\frac{(G_j - \lambda)^2}{H_j + \lambda}\right) + \frac{\lambda}{2}T$$

and the gain calculation formula becomes

$$Gain' = \frac{1}{2}\left[\frac{(G_L - \lambda)^2}{H_L + \lambda} + \frac{(G_R - \lambda)^2}{H_R + \lambda} - \frac{(G_L + G_R - \lambda)^2}{H_L + H_R + \lambda}\right] - \frac{\lambda}{2} \quad (4)$$

It can be seen from the above formula that the new definition makes $|w_j'| < |w_j^*|$, that is, compared with the original regularization term, the parameters become smaller after the change, so overfitting can be avoided and the model variance can be reduced.

Classification algorithm

Since mixed sampling can take into account both under-sampling and over-sampling, it can often produce better results. Therefore, this article considers the integration of mixed sampling method and ensemble learning method, and designs an XGBoost imbalanced data classification algorithm which combines the SVM–SMOTE algorithm and the EasyEnsemble algorithm. Its strategy is as follows:

1. Divide the original data set D into training set and test set according to the preset proportion.
2. Use the SVM–SMOTE algorithm to oversample the minority samples P in the training set once to generate a sample set P' to increase the minority sample size.
3. Use the EasyEnsemble under-sampling algorithm to independently and randomly extract multiple subsets $N_i(i = 1:m)$ from the majority samples N in the training set, so that $|N_i| = |P'|$, and then combine N_i and P' into multiple balanced subsets and train multiple weak learners (apply XGBoost with better classification performance instead of AdaBoost as a weak learner), and then ensemble the results to obtain a strong learner model.
4. Use the original parameters and area under the curve (AUC) values of XGBoost as the input and output of the objective function in the Bayesian optimization search, adjust the best parameter combination in time, and perform K-fold cross-validation on the strong learner.
5. Based on the model obtained after tuning the parameters, the final prediction is completed on the test set.

The flow framework of the algorithm is shown in Figure 1.

Experiment and result analysis

Experimental platform and data introduction

The experimental platform environment of this article is Window 10×64 operating system, 8 GB memory, Intel(R)Core(TM)i7-7500 CPU@2.70 GHz; the experimental tools are mainly Python3.9, including xgboost 1.3.3, jupyter1.0.0, seaborn0.11.1, sklearn0.0, pandas 1.2.2, numpy1.20.1, matplotlib3.3.4, imblearn0.0, and other packages.

In order to prove the classification performance of the proposed algorithm in imbalanced data sets, two public imbalanced data sets are used for experiments. The first data set is the credit card data set in Taiwan publicly provided on UCI website. The data set records the history of bank customers' arrears, credit data, statistical characteristics, billing statements, and other information from April to September 2005. It has 24 characteristic attributes and one category identification, which can predict the customer's default situation. The specific meaning of features is shown in Table 2. The second data set is the credit fraud data set provided by ULB machine learning laboratories. The data set contains the credit card transactions of a bank in Europe in September 2013, including 492 frauds in 284,807 transactions, and the data categories are extremely imbalanced. For security reasons, the original data set has been desensitized and principal component analysis (PCA) dimensionality reduction, with a total of 29 feature attributes and 1 category identification. Among them, the features "V1"–"V28" are the principal components obtained by PCA, and the features not processed by PCA are "Time" and "Amount." "Time" represents the time interval between all transactions and the first transaction, and "Amount" represents the transaction amount. For the class label "Class," 1 represents the fraud and 0 represents the normal.

Data analysis

The first data set is mainly shown and analyzed below. The data in the second data set do not need cleaning, and there is no significant correlation between characteristic variables. The processing flow is similar to that in the first data set.

First, by loading and viewing the data set, and performing missing value testing and descriptive statistical analysis, it is found that the data have a total of 30,000 rows and 25 columns, all features have no missing values, and all the data are integers. Due to the large differences between the values, standardization is required. Part of the data and test results are shown in Figures 2 and 3.

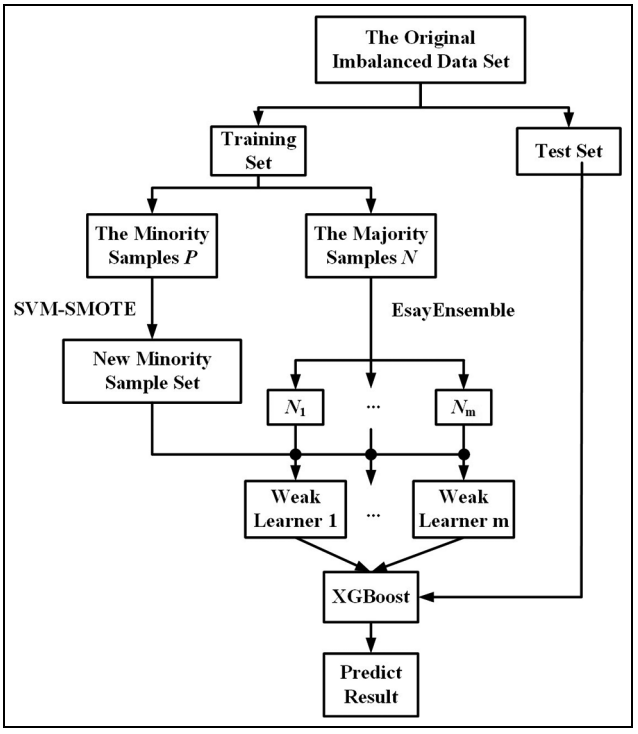

Figure 1. Framework diagram of algorithm flow.

Second, by looking at the sample distribution of each feature, the outliers were tested, and it was found that the features EDUCATION and MARRIAGE were abnormal. The value of DUCATION is 0, 5, and 6 more than in the introduction, and the number of outliers is 345, which is relatively small compared to the total sample size, so it is classified as one with 4. The value of MARRIAGE is 0 more than in the introduction, and the number of outliers is 54. It is classified as one with 3 for correction and filling.

At the same time, analysis shows that there will be 6636 customers defaulting next month, which is much lower than the number of non-defaulting customers (23,364), so the data are obviously imbalanced. The information of the two experimental data sets is summarized in Table 3.

Moreover, by analyzing the variables in the credit card data set, it can be seen that among all customers, the default ratio of males is 24.2%, and that of females is 20.8%. The number of customers who repay on time between 30 and 40 is the largest, and the older the age, the higher the default rate. The unmarried customer group has the largest number of repayments on time. The number of defaults is similar to that of married groups. High school customers have the highest default rate, and the higher the education level, the lower the default rate.

Table 2. Characteristic description.

Number	Features	Description
1	ID	Customer's unique identification
2	LIMIT_BAL	Credit line (in NT), including personal and family credit lines
3	SEX	Customer's gender (1 = male; 2 = female)
4	EDUCATION	Customer's educational level (1 = master's degree and above; 2 = undergraduate; 3 = high school; 4 = others)
5	MARRIAGE	Customer's marital status (1 = married; 2 = unmarried; 3 = other)
6	AGE	Customer's age
7	PAY_0	Repayment status in September
8	PAY_2	Repayment status in August
9	PAY_3	Repayment status in July
10	PAY_4	Repayment status in June
11	PAY_5	Repayment status in May
12	PAY_6	Repayment status in April
13	BILL_AMT1	Bill amount in September
14	BILL_AMT2	Bill amount in August
15	BILL_AMT3	Bill amount in July
16	BILL_AMT4	Bill amount in June
17	BILL_AMT5	Bill amount in May
18	BILL_AMT6	Bill amount in April
19	PAY_AMT1	Payment amount in September
20	PAY_AMT2	Payment amount in August
21	PAY_AMT3	Payment amount in July
22	PAY_AMT4	Payment amount in June
23	PAY_AMT5	Payment amount in May
24	PAY_AMT6	Payment amount in April
25	default.payment.next.month	Default in the next month (1 = yes; 0 = no)

For PAY_0 and PAY_2-PAY_6, the value of −1 means that the customer has repaid on time, 1 means that the customer will postpone the repayment for 1 month, 2 means that the customer will postpone the repayment for 2 months, and so on. In addition, if the value of PAY_AMT is greater than or equal to the previous month's BILL_AMT value, it is deemed to be repaid on time. If the value is less than the previous month's BILL_AMT value but is greater than the lower limit of the repayment amount set by the bank, it will be regarded as delayed repayment, and less than the minimum repayment amount is default.

```
In [3]: data
```

Out[3]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689
1	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000
2	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500
3	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019
4	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29995	29996	220000	1	3	1	39	0	0	0	0	...	88004	31237	15980	8500	20000
29996	29997	150000	1	3	2	43	-1	-1	-1	-1	...	8979	5190	0	1837	3526
29997	29998	30000	1	2	2	37	4	3	2	-1	...	20878	20582	19357	0	0
29998	29999	80000	1	3	1	41	1	-1	0	0	...	52774	11855	48944	85900	3409
29999	30000	50000	1	2	1	46	0	0	0	0	...	36535	32428	15313	2078	1800

30000 rows × 25 columns

Figure 2. Partial data display.

```
In [5]: print("The dimension of the dataframe is: ", data.shape)
        data.info()

The dimension of the dataframe is:  (30000, 25)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null  int64
 1   LIMIT_BAL                   30000 non-null  int64
 2   SEX                         30000 non-null  int64
 3   EDUCATION                   30000 non-null  int64
 4   MARRIAGE                    30000 non-null  int64
 5   AGE                         30000 non-null  int64
 6   PAY_0                       30000 non-null  int64
 7   PAY_2                       30000 non-null  int64
 8   PAY_3                       30000 non-null  int64
 9   PAY_4                       30000 non-null  int64
 10  PAY_5                       30000 non-null  int64
 11  PAY_6                       30000 non-null  int64
 12  BILL_AMT1                   30000 non-null  int64
 13  BILL_AMT2                   30000 non-null  int64
 14  BILL_AMT3                   30000 non-null  int64
 15  BILL_AMT4                   30000 non-null  int64
 16  BILL_AMT5                   30000 non-null  int64
 17  BILL_AMT6                   30000 non-null  int64
 18  PAY_AMT1                    30000 non-null  int64
 19  PAY_AMT2                    30000 non-null  int64
 20  PAY_AMT3                    30000 non-null  int64
 21  PAY_AMT4                    30000 non-null  int64
 22  PAY_AMT5                    30000 non-null  int64
 23  PAY_AMT6                    30000 non-null  int64
 24  default.payment.next.month  30000 non-null  int64
dtypes: int64(25)
memory usage: 5.7 MB
```

Figure 3. Missing value test results.

Finally, through comprehensive feature analysis, it is found that there is only PAY_0 has the most complete information in the repayment status, and the two types of features BILL_AMT and PAY_AMT are highly correlated, so features with large correlation coefficient with default.payment.next.month can be selected for modeling, respectively. Therefore, delete ID, PAY_2–PAY_6, BILL_AMT2, BILL_AMT4–BILL_AMT6, PAY_AMT3–PAY_AMT 6, and default.payment.next.month. The specific feature correlation heatmap is shown in Figure 4.

Performance evaluation index

In traditional classification problems, single evaluation indexes such as recall rate and accuracy rate are often used to better reflect the performance of the algorithm.

However, in the face of the imbalanced classification problem of data tilt, only using single index no longer has good reference value. Therefore, this article selects the typical comprehensive evaluation index G-mean and AUC value to compare and analyze the prediction effect of the experiment.

In the confusion matrix, TP, FP, FN, and TN, respectively, represent the sample sizes of true positive, false positive, false negative, and true negative cases. From this, the definitions of the above-mentioned types of evaluation indexes can be obtained

$$\begin{aligned}
\text{Recall} = \text{Sensitivity} = \text{TPR} &= \frac{\text{TP}}{(\text{TP} + \text{FN})} \\
\text{Precision} &= \frac{\text{TP}}{(\text{TP} + \text{FP})} \\
\text{Specificity} &= \frac{\text{TN}}{(\text{TN} + \text{FP})} \\
\text{FPR} &= \frac{\text{FP}}{(\text{TN} + \text{FP})} \\
\text{G} - \text{mean} &= \sqrt{\text{Sensitivity} \times \text{Specificity}}
\end{aligned} \tag{5}$$

Therefore, G-mean contains two single evaluation indexes Sensitivity and Specificity, which improve and perfect the overall accuracy and can better measure the classification effect. In addition, the receiver operating characteristic (ROC) curve is drawn with FPR and TPR as the horizontal and vertical axes, respectively. AUC is the coverage area below the ROC curve. The larger the value, the closer the ROC curve will be to the upper left corner, representing the more accurate the classification effect of the model.

Experiment settings and results

Experiment settings. Through data analysis and feature selection, we first divide the original imbalanced data set into training set and test set in the ratio of 7:3, and designs two groups of comparative experiments. The average value after 10-fold cross-validation is used as the experimental result to make it more objective.

Group 1: (compare the feasibility of the proposed algorithm at different levels)

1. The training set does not perform any operation, but directly constructs the XGBoost

Table 3. Experimental data set information.

Data set	Number of samples	Number of positive samples	Number of negative samples	Imbalance rate	Number of categories
Credit card	30,000	6636	23,364	4.52	2
Credit fraud	284,807	492	284,315	577.88	2

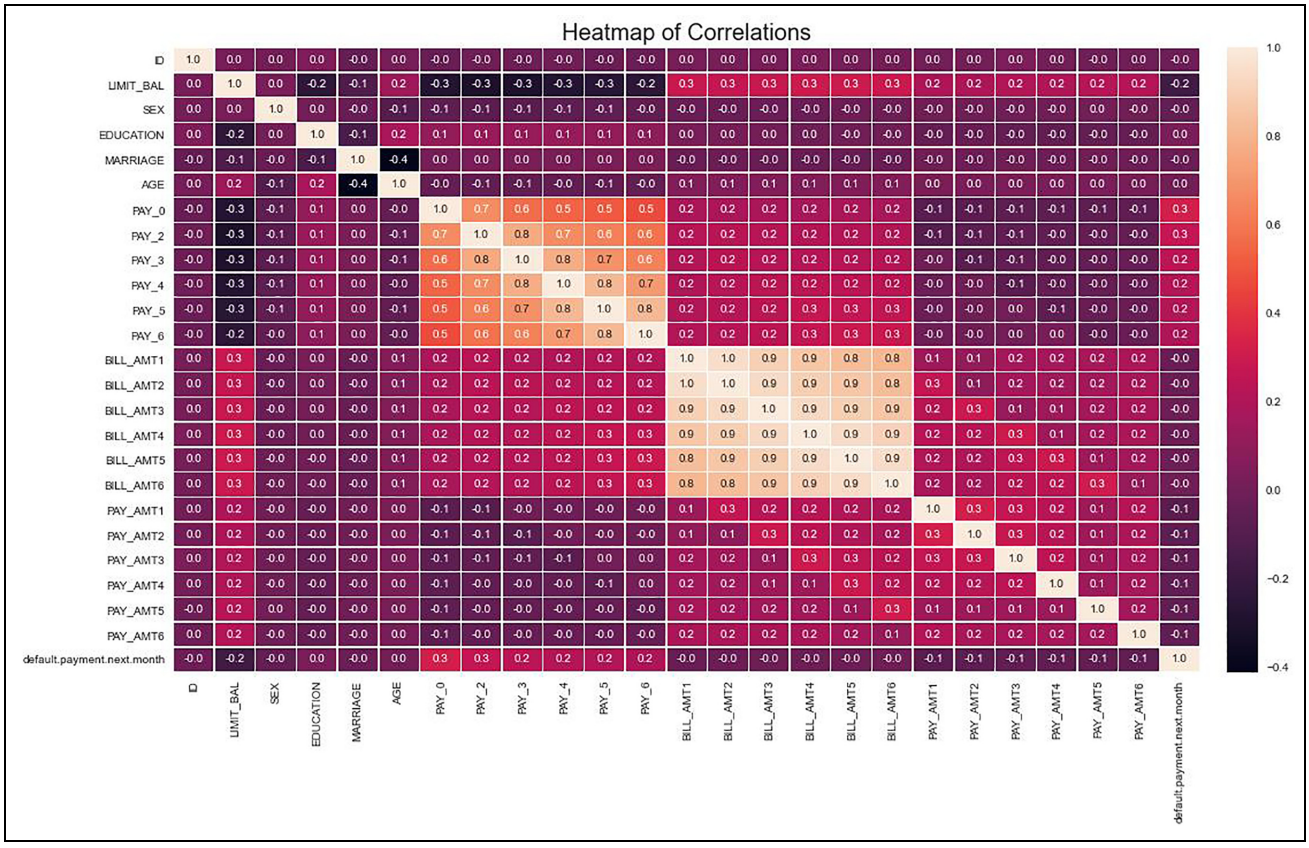

Figure 4. Feature correlation heatmap.

Table 4. Results of Experiment I.

Data set	Evaluation index	Models				
		XGBoost	SS-XGB	EE-XGB	SE-XGB	SEB-XGB
Credit card	AUC	0.7545	0.7445	0.7645	0.7649	0.7728
	G-mean	0.5763	0.6542	0.6933	0.6930	0.7003
Credit fraud	AUC	0.9654	0.9683	0.9709	0.9740	0.9849
	G-mean	0.8853	0.9078	0.9238	0.9364	0.9502

AUC: area under the curve.

classifier for experimentation, and the parameters are all default.

2. Select SVM–SMOTE over-sampling and EasyEnsemble under-sampling at the data level, respectively, and use XGBoost modeling experiment to compare the classification effect after adding sampling (the model is abbreviated as SS-XGB and EE-XGB).
3. Use the SVM–SMOTE + EasyEnsemble + XGBoost (SE-XGB) model, where the sampling proportion of SVM–SMOTE is set to 30% of the majority class.
4. Use the SVM–SMOTE + EasyEnsemble + Bayesian search tuning + XGBoost (SEB-XGB) model.

Group 2: (comparison between SEB-XGB and other imbalanced classification models). In order to prove the effectiveness of the algorithm model proposed in this article, it is compared with the classic improved algorithm RUSBoost in the field of imbalanced data classification, the recently popular improved algorithms CatBoost, LightGBM, and the EBB-XGBoost algorithm proposed in Yue.[27]

Table 5. Results of Experiment II.

Data set	Evaluation index	Models				
		RUSBoost	CatBoost	LightGBM	EBB-XGBoost	SEB-XGB
Credit card	AUC	0.7623	0.7695	0.7687	0.7683	0.7728
	G-mean	0.6874	0.5659	0.5674	0.7003	0.7003
Credit fraud	AUC	0.9533	0.9685	0.9694	0.9821	0.9849
	G-mean	0.9147	0.8738	0.8621	0.9344	0.9502

AUC: area under the curve.

Table 6. SEB-XGB parameter combination.

Data set	Parameter combination
Credit card	0.1016, 5, 0.7701, 2.094, 0.9995, 90, 2.317, 2.187
Credit fraud	0.2999, 5, 1.5252, 0.5318, 0.8, 85, 0.0865, 1.5804

Result analysis. The comparison results of the Group 1 and the Group 2 of experiments on the data set are shown in Tables 4 and 5, respectively.

Table 6 shows the optimal parameter combination results of Bayesian automatic search. The parameter combinations are as follows: learning_rate, max_depth, min_child_weight, colsample_bytree, subsample, n_estimators, lamda, and alpha.

Among them, for credit card data set, the AUC value of SEB-XGB after 10-fold cross-validation can reach 0.7796, while for credit fraud data set, the AUC value after 10-fold cross-validation can reach 0.9998.

It can be seen from Table 4 that the classification performance of SEB-XGB model has been improved by gradually adding data-level sampling processing, using the model combining mixed sampling and ensemble learning, and finally adding Bayesian parameter tuning. Compared with a single XGBoost, SEB-XGB increases the G-mean and AUC values by 12.4% and 2.51%, respectively, in the first data set, and 6.49% and 4.36%, respectively, in the second data set, which proves the feasibility of the proposed algorithm.

As can be seen from Table 5, on the two data sets, compared with other improved classification models, the G-mean and AUC values of SEB-XGB are the best on the whole, indicating that the algorithm has higher recognition rate and better classification prediction effect.

Conclusion

In order to improve the classification performance of XGBoost when the data are imbalanced, this article proposes an SEB-XGB algorithm combining sampling technology and ensemble learning from the two aspects of algorithm principle and imbalanced data processing. This algorithm first uses SVM–SMOTE over-sampling at the data level to generate minority supplementary samples, and then uses EasyEnsemble under-sampling to balance the data categories. Then, at the algorithm level, XGBoost is used as the base learner for training and ensemble, and the final model is obtained by Bayesian automatic search and optimization parameters. The results of two groups of comparative experiments show that the proposed algorithm is feasible and has a better effect than the original single XGBoost algorithm and other improved classification algorithms.

However, this article only studies the imbalanced binary classification problem, which also has certain limitations. As the research continues to deepen, we will try to explore multi-classification problems in the future.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (nos 12071112 and 11471102) and the Key Scientific Research Projects of Colleges and Universities in Henan Province (no. 20A520012).

ORCID iD

Ping Zhang https://orcid.org/0000-0002-8500-8727

References

1. Song LL, Wang SH, Yang C, et al. Application research of improved XGBoost in imbalanced data processing. *Comput Sci* 2020; 47(6): 98–103.
2. Liu DX, Qiao SJ, Zhang YQ, et al. A survey on data sampling methods in imbalance classification. *J Chongqing Univ Technol Nat Sci* 2019; 33(7): 102–112.
3. Fan XN. *Research on imbalanced dataset classification*. Hefei, China: University of Science and Technology of China, 2011.
4. Wan ZC. *Research on imbalanced classification method based on XGBoost*. Hefei, China: Anhui University, 2018.
5. Xu LL and Chi DX. Machine learning classification strategy for imbalanced data sets. *Comput Eng Appl* 2020; 56(24): 12–27.
6. Chawla NV, Lazarevic A, Hall LO, et al. SMOTEBoost: improving prediction of the minority class in boosting. In: *Proceedings of the European conference on principles of data mining and knowledge discovery*, Dubrovnik, 22–26 September 2003, pp.107–119. Berlin: Springer.
7. Cheng F, Zhou Y, Gao J, et al. Efficient optimization of F-measure with cost-sensitive SVM. *Math Probl Eng* 2016; 2016: 5873769.
8. Chen T and Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, CA, 13–17 August 2016, pp.785–794. New York: ACM.
9. Qu WL, Li YQ and Zhou L. Application of XGBoost algorithm in diabetic blood glucose prediction. *J Jilin Norm Univ Nat Sci Ed* 2019; 40(4): 118–125.
10. Yu GL, Zhao HS and Mi ZQ. Application of XGBoost algorithm in prediction of wind motor main bearing fault. *Electr Pow Autom Equip* 2019; 39(1): 73–77, 83.
11. Ma QQ, Sun DX, Shi JM, et al. Risk prediction of cancer in adult population based on support vector machine versus XGBoost. *Chin Gen Pract* 2020; 23(12): 1486–1491.
12. Yuan LX, Gu YJ and Zhao DP. Research on abnormal user detection technology in social network based on XGBoost method. *Appl Res Comput* 2020; 37(3): 814–817.
13. Romeo L and Frontoni E. A unified hierarchical XGBoost model for classifying priorities for COVID-19 vaccination campaign. *Pattern Recogn* 2022; 121: 108197.
14. Batunacun, Wieland R, Lakes T, et al. Using shapley additive explanations to interpret extreme gradient boosting predictions of grassland degradation in Xilingol, China. *Geosci Model Develop* 2021; 14(3): 1493–1510.
15. Zhang XL, Xiu CD, Wang YZ, et al. High-precision WiFi indoor localization algorithm based on CSI-XGBoost. *J Beijing Univ Aeronaut Astronaut* 2018; 44(12): 2536–2544.
16. Liu Y and Qiao M. Heart disease prediction based on clustering and XGBoost algorithm. *Comput Syst Appl* 2019; 28(1): 228–232.
17. Du XD, Li W, Ruan S, et al. CUS-heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection. *Appl Soft Comput* 2020; 97(Part A): 106758.
18. Lin MY, Zhu XF, Hua T, et al. Detection of ionospheric scintillation based on XGBoost model improved by SMOTE-ENN technique. *Rem Sens* 2021; 13(13): 2577.
19. Li YZ, Wang ZY, Zhou YL, et al. The improvement and application of XGBoost method based on the Bayesian optimization. *J Guangdong Univ Technol* 2018; 35(1): 23–28.
20. Wang Y and Guo YK. Application of improved XGBoost model in stock forecasting. *Comput Eng Appl* 2019; 55(20): 202–207.
21. Zhang CF, Wang S, Wu YD, et al. Diabetes risk prediction based on GA_XGBoost model. *Comput Eng* 2020; 46(3): 315–320.
22. Jia QC. *Anomaly detection of bolt tightening process for imbalanced data sets*. Jinan, China: Shandong University, 2018.
23. Cui LS. *Application of hybrid XGBoost model in unbalanced dataset classification predication*. Lanzhou, China: Lanzhou University, 2018.
24. Seiffert C, Khoshgoftaar TM, Van HJ, et al. RUSBoost: a hybrid approach to alleviating class imbalance. *IEEE Trans Syst Man Cybern Part A Syst Hum* 2010; 40(1): 185–197.
25. Prokhorenkova L, Gusev G, Vorobev A, et al. CatBoost: unbiased boosting with categorical features, 2017, https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf
26. Ke G, Meng Q, Finley T, et al. LightGBM: a highly efficient gradient boosting decision tree, 2017, https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
27. Yue QS. *Research on XGBoost performance optimization based on imbalanced data*. Lanzhou, China: Lanzhou Jiaotong University, 2019.
28. Chawla NV, Bowyer KW, Hall LO, et al. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 2002; 16(1): 321–357.
29. Shi H, Chen Y and Chen X. Summary of research on SMOTE oversampling and its improved algorithms. *CAAI Trans Intell Syst* 2019; 14(6): 1073–1083.
30. Liu XY, Wu J and Zhou ZH. Exploratory undersampling for class-imbalance learning. *IEEE Trans Syst Man Cybern Part B* 2009; 39(2): 539–550.
31. Zou H and Hastie T. Regularization and variable selection via the elastic net. *J Roy Stat Soc* 2005; 67(5): 301–320.