



## A COST-SENSITIVE LOGISTIC REGRESSION CREDIT SCORING MODEL BASED ON MULTI-OBJECTIVE OPTIMIZATION APPROACH

Feng SHEN<sup>\*</sup>, Run WANG, Yu SHEN

*School of Finance, Southwestern University of Finance and Economics, Chengdu 611130, PR China*

Received 18 February 2019; accepted 15 September 2019

**Abstract.** Credit scoring is an important process for peer-to-peer (P2P) lending companies as it determines whether loan applicants are likely to default. The aim of most credit scoring models is to minimize the classification error rate, which implies that all classification errors bear the same cost; however, in reality, there is a significant cost-sensitive problem in credit scoring methods. Therefore, in this paper, a new cost-sensitive logistic regression credit scoring model based on a multi-objective optimization approach is proposed that has two objectives in the cost-sensitive logistic regression process. The cost-sensitive logistic regression parameters are solved using a multiple objective particle swarm optimization (MOPSO) algorithm. In the empirical analysis, the proposed model was applied to the credit scoring of a Chinese famous P2P company, from which it was found that compared with other common credit scoring models, the proposed model was able to effectively reduce type II error rates and total classification error costs, and improve the AUC, the F1 values (reconciliation average of Recall and Precision), and the G-means. The proposed model was compared with other multi-objective optimization algorithms to further demonstrate that MOPSO is the best approach for cost-sensitive logistic regression credit scoring models.

**Keywords:** credit scoring, cost-sensitive, logistic regression, multi-objective optimization, P2P.

**JEL Classification:** C53, C61, E51, G32.

### Introduction

Due to the prevalence of financial frictions and credit constraints, it is difficult for individuals and SME (small to medium-sized enterprise) to obtain loans from banks (Rashid & Jabeen, 2018; Abraham, 2018). The peer-to-peer (P2P) lending market began to develop as an emerging lending channel. P2P lending market is a platform based portal that allows individual lenders to come into contact with borrowers seeking loans, for which the lender assumes full risk. P2P lending has become popular because of the reduced financing costs (Guo, Zhou, Luo, Liu, & Xiong, 2016). In the P2P lending market, the borrower submits a loan applica-

---

\*Corresponding author. E-mail: [shenfeng1213@gmail.com](mailto:shenfeng1213@gmail.com)

tion through the website platform, and the lender decides whether or not to lend the amount requested based on the relevant borrower information (Zhu, Li, Wu, Wang, & Liang, 2013). However, relaxed regulation has caused many credit risks to society as well as increased losses for investors (Chen, Li, Wu, & Luo, 2017). Therefore, as predicting whether a borrower is able to repay the loan in the requested or granted time frame is extremely important, algorithmic online credit scoring has become a vital tool for P2P lending companies (Verbraken, Bravo, Weber, & Baensens, 2014). In credit scoring models, the dependent variable is dichotomous, with a default being assigned a “0”, and a successful loan being assigned a “1” (Serrano-Cinca & Gutiérrez-Nieto, 2016). Credit scoring models estimate default probabilities (PD) based on applicant credit histories so that lending institutions can determine whether to approve or reject the loan application (Bequé, Coussement, Gayler, & Lessmann, 2017).

Traditional credit scoring models build a series of statistical models based on the premise that different classification errors have a consistent cost using classifiers such as logistic regression (LR) (Wiginton, 1980), support vector machines (SVM) (Min & Lee, 2005), decision trees (DT) (Huysmans, Dejaeger, Mues, Vanthienen, & Baensens, 2011), neural networks (NN) (Khashman, 2010), or ensemble approaches such as bagging (Yu, Yue, Wang, & Lai, 2010), and boosting (Wang, Ma, Huang, & Xu, 2012). However, in the credit scoring process, cost-sensitivity is more important as the cost associated with approving an application for someone who defaults on the loan is far greater than the cost associated with rejecting an application from a customer who may have successfully repaid the loan (Kao, C. C. Chiu, & F. Y. Chiu, 2012).

Therefore, cost-sensitive learning is a relatively new machine learning approach that trains classifiers to recognize the different costs associated with different classification errors. As shown in Figure 1, the orange circle indicates loan applicants who may default in the future and the green circle indicates applicants who may not default in the future. If it is assumed that the cost of misjudging a non-defaulter as a defaulter is 1, and the cost of judging a defaulter as a non-defaulter is 5, in Figure 1 left, there is a traditional classifier that minimizes this error rate; that is, the classification error cost of judging a defaulter as a non-defaulter is 5. The right side is the cost-sensitive learner that minimizes the cost of the classification errors by judging both borrowers as possible defaulters so that the resulting classification error cost is  $1 + 1 = 2$ . Therefore, the cost-sensitive classifier has less total cost and is more in line with the actual credit scoring situation (Günnemann & Pfeffer, 2017).

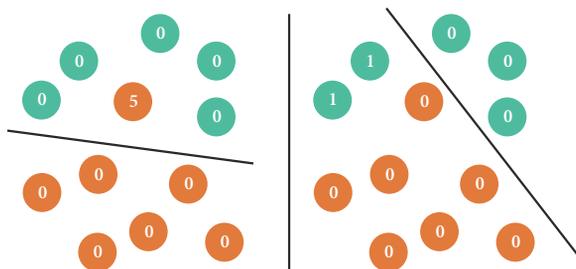


Figure 1. The result of a cost-insensitive classifier (left); the result of a cost-sensitive classifier (right)

Cost-sensitive learning algorithms can be divided into direct and indirect cost-sensitive methods (Ling & Sheng, 2011). The direct method builds a cost-sensitive learning algorithm by directly introducing different misclassification costs into the learning process, and the indirect cost-sensitive approach converts the cost-insensitive classifiers into cost-sensitive classifiers by preprocessing the training data through undersampling or oversampling or through postprocessing such as thresholding, which changes the cutoff value into a positive or negative when classifying a sample (Xia, C. Liu, & N. Liu, 2017). García, Marqués, and Sánchez (2012) examined whether resampling methods could improve classifier performance and found that for any classifier, the classifier performance after resampling was better than using unbalanced data. Marqués, García, and Sánchez (2013) then compared this credit scoring resampling method and found that it was able to balance cost-sensitive data and that using an oversampling method was better than using an undersampling method. As indirect cost-sensitive methods are only modeled at the data level and are therefore not studied with the model itself, direct cost-sensitive methods may have more value. Therefore, the proposed method in this paper is based on a direct cost-sensitive method.

In a direct cost-sensitive approach, J. Kim, Choi, G. Kim, and Suh (2012) compared a traditional classifier with a cost-sensitive decision tree and found that the lowest classification cost was incurred when a MetaCost approach was used and when the non-fraud data and fraud data were balanced. Bahnsen, Aouada, and Ottersten (2015) constructed an example-dependent cost-sensitive decision tree with a novel impurity measure and pruning criteria, which was found to outperform baseline models in terms of cost savings and training time. Of the various direct methods, cost-sensitive decision trees have been the most popular method because of their simple operability; however, as cost-sensitive decision trees are sensitive to data patterns, any small change in the training set can result in a completely different tree and a significant change in the predictions (James, Witten, Hastie, & Tibshirani, 2013). Logistic regression, however, is not sensitive to data, is suitable for continuous variables, and the regression results contain the parameters for each variable, with the size and symbol of the parameter indicating whether and how the variable affects the loan applicant's default probability; therefore, logistic regression is very explanatory and practical (Bequé et al., 2017).

Even though logistic regression has been found to have significant interpretability advantages, there have been few studies focused on the use of cost-sensitive logistic regression. Bahnsen, Aouada, and Ottersten (2014) developed an example-dependent cost-sensitive logistic regression model using two publicly available datasets, which achieved high-cost savings compared to the benchmarks; however, the loss function was linear, which caused weak differentiation between the false and correctly classified instances. Günnemann and Pfeffer (2017) introduced a new cost-sensitive prediction model based on a nonlinear loss function, which extended logistic regression and allowed for the different costs of misclassified instances, thereby obtaining prediction results with an overall lower cost. However, most of these cost-sensitive models have been based on a single optimization criterion. Therefore, in order to ensure the excellent performance of the classification model while identifying the defaulters who have the higher costs as accurately as possible, this paper seeks to synthesize

the two goals to construct a multi-objective optimization logistic regression model based on a maximum AUC and a minimum total classification cost, and then solves the proposed model using a MOPSO evolutionary algorithm.

In this paper, a logarithmic loss function and the AUC objective function are used to optimize the logistic regression parameters. The two objective functions are embedded in the MOPSO optimization algorithm to determine the optimal parameter set. This paper aims to construct a direct cost-sensitive logistic regression method based on multi-objectives to obtain a model that minimizes the total cost and maximizes the AUC. As the constructed model is better able to identify the sample categories with higher losses, it can therefore assist lenders identify possible defaulters, reduce default losses, and increase profits. The remainder of this paper is structured as follows. Section 1 introduces the preliminary knowledge used in this paper; logistic regression, the cost matrix, the AUC, and the multi-objective optimization algorithm. In Section 2, the multi-objective optimization cost-sensitive logistic regression is constructed, and in Section 3, the proposed model and algorithm are applied to a P2P dataset to evaluate its performance, and the results are compared with single-objective optimization and other multi-objective optimization algorithms to verify its effectiveness. Finally, conclusions and further research recommendations are given in the last section.

## 1. Preliminary knowledge

This section gives a brief introduction to the theoretical knowledge used in the following sections; the theory of logistic regression, the derivation of the AUC, and the meaning of the cost matrix.

### 1.1. Logistic regression

Logistic regression is a classification model that estimates the posterior probability of a positive class in a specific binary classification context. Logistic regression is one of the most widely used statistical models for deriving classification algorithms (Abdou, Tsafack, Ntim, & Baker, 2016). Given an instance  $X_i$ , the estimated probability of the positive class is evaluated as  $P(y=1|X_i)$ , and  $P(y=0|X_i)=1-P(y=1|X_i)$  respectively for the negative class occurrence. Here,  $f(g(x,\beta))$  is defined as the sigmoid function, which is known as the logistic (Desai, Crook, & Overstreet Jr, 1996):

$$p_i = P(y=1|X_i) = f(g(x,\beta)) = \frac{1}{1 + e^{-g(x,\beta)}}, \quad (1)$$

where  $g(x,\beta) = \beta_0 + \sum_{j=1}^m \beta_j x_j$  is a linear function of the logistic regression parameters  $\beta$  and the explanatory variables, and  $0 \leq p_i \leq 1$  refers to the hypothesis of  $i$  given the parameters  $\theta$ . The constructed logistic regression model is solved to ensure a consistent estimate with a true value, and is usually determined using the maximum likelihood estimation method (MLE), which assumes that each sample in the dataset is independent. When the true value  $y_i = 1$ ,  $p_i$  represents the probability that the sample is predicted to be 1; that is, the bigger the

$p_i$ , the smaller the difference between the predicted value and the true value. When the true value  $y_i = 0$ ,  $1 - p_i$  represents the probability that the sample is predicted to be 0; that is, the bigger the  $1 - p_i$ , the smaller the difference between the predicted value and the true value. The MLE principle ensures that the difference between the predicted value and the true value is the smallest; that is, when  $y_i = 1$ ,  $p_i$  is the largest and when  $y_i = 0$ ,  $1 - p_i$  is the largest. The likelihood function for MLE is derived as follows (Günnemann & Pfeffer, 2017).

$$L(x, \beta) = \prod_{i=1}^n f(g(x_i, \beta))^{y_i} \cdot (1 - f(g(x_i, \beta)))^{(1-y_i)} \tag{2}$$

As the  $\beta$  in logistic regression is obtained by maximizing  $L(x, \beta)$ , the maximum likelihood function solves the parameter by maximizing the difference between the predicted value and the true value; therefore, there is no cost-sensitive thinking in the MLE.

### 1.2. AUC

In traditional credit scoring models, the AUC (area under the receiver operating characteristic curve) is an effective measure for evaluating model performance. This curve is an ROC curve (Receiver Operating Characteristic). To predict the default probability of applicants  $p_i = P(y_i = 1 | X_i)$ , traditional credit scoring models construct a suitable classifier  $h_i(x)$  based on a historical dataset of the loan applicants, for which an appropriate threshold  $t$  is chosen, typically 0.5; when  $p_i < t$ ,  $h_i(x) = 1$ , the applicant’s loan is approved, and when  $p_i > t$ ,  $h_i(x) = 0$ , the applicant’s loan is declined. However, threshold  $t$  selection greatly affects the evaluation index value (Alaraj & Abbod, 2016). However, as the calculation of the AUC does not need to turn the prediction probabilities into categories, it is more useful for evaluating model performance.

There is a need to evaluate the classifier  $h_i(x)$  performance after it is estimated. In practice, AUC has been widely used to assess the performance of a credit scoring model. In a dichotomous problem, samples have two categories; positive and negative. When a classifier predicts samples, positive samples can be predicted as negative samples or positive samples, and negative samples can be predicted as negative samples or positive samples; that is, there are four circumstances. A sample that is truly positive and is also predicted to be positive is called a true positive (TP), a sample that is truly positive but predicted to be negative is called a false negative (FN), a sample that is truly negative and also predicted to be negative is called true negative (TN), and a sample that is truly negative but predicted to be positive is called a false positive (FP). These four cases constitute the classification matrix, as shown in Table 1.

Table 1. Classification matrix

	Actual positive $y_i = 1$	Actual negative $y_i = 0$
Predicted positive $h_i = 1$	TP	FP
Predicted negative $h_i = 0$	FN	TN

From the classification matrix, the true positive rate (TPR) and false positive rate (FPR) can be calculated.

$$TPR = \frac{TP}{TP + FN}; \tag{3}$$

$$FPR = \frac{FP}{FP + TN}. \tag{4}$$

The ROC curve is based on the false positive rate on the horizontal axis and the true positive rate on the vertical axis. First, the classification threshold is adjusted to the maximum, at which time the true and false positive rates are both 0, after which the classification threshold is set as the predicted value of each example in sequence so that each predicted value corresponds to a set of true positive rates and false positive rates. The ROC curve is obtained by plotting these true positive rates and false positive rates on a graph, with the area under the ROC curve being the AUC; the larger the AUC, the better the model performance.

This article uses the following formula to calculate the AUC (Fawcett, 2006):

$$AUC = \frac{\sum_{i \in \text{positive class}} rank_i - \frac{M(1+M)}{2}}{M * N}, \tag{5}$$

where  $M$  is the number of positive samples and  $N$  is the number of negative samples, and  $rank_i$  is the ranking of the forecast probability of the  $i$ th positive sample.

### 1.3. Cost matrix

In a cost-sensitive learning method, the different categories have the different classification error cost, and the cost-sensitive learning model performance evaluation is not limited to the AUC. The cost matrix is used to express the different classification error costs. As classification errors have different costs, different weights are assigned to different classifications, as shown in Table 2. The weights that classify a positive class as a positive class are  $c_{TP}$ , the weights that classify a positive class as a negative class are  $c_{FN}$ , the weights that classify a negative class as a positive class are  $c_{FP}$ , and the weights that classify a negative class as a negative class are  $c_{TN}$  (Günnemann & Pfeffer, 2017).

Table 2. Cost matrix

	Actual positive $y_i = 1$	Actual negative $y_i = 0$
Predicted positive $h_i = 1$	$c_{TP}$	$c_{FP}$
Predicted negative $h_i = 0$	$c_{FN}$	$c_{TN}$

In general, only the cost of misclassification is considered; that is  $c_{TP} = c_{TN} = 0$ . Therefore, for the classifier  $h_i(x)$ , the expected cost based on this cost matrix is

$$E(\text{cost}) = \sum_{i=1}^n (c_{FP} * (1 - y_i) * p_i + c_{FN} * y_i * (1 - p_i)), \tag{6}$$

where  $y_i \in \{0,1\}$  is the observed and  $h_i \in \{0,1\}$  is the predicted category for instance  $i$ , with  $n$  being the number of instances.

### 1.4. Multi-objective particle swarm optimization

In a single-objective optimization algorithm, the particle swarm optimization (PSO) algorithm, which is a novel optimization algorithm inspired by birds swarming, provides high-speed convergence. Compared to other evolutionary computing models, PSO is more popular (Kou, Chao, Peng, Alsaadi, & Herrera-Viedma, 2019). The multi-objective Particle Swarm Optimization (MOPSO) was developed based on the PSO. However, when applying a single-objective PSO algorithm to a multi-objective optimization problem, there are several problems: (1) selecting the leader particles for the updating process, (2) maintaining the non-dominated solution to obtain the optimal Pareto front, and (3) preventing the particles from converging to a local optimum and maintaining group diversity (Ding, Chen, Xin, & Pardalos, 2018). To resolve these three problems for multi-objective optimization problems, each particle may have a set of different leaders from which just one can be selected to update its position. As these leader sets are usually stored in a different place from the swarm, these are called an external archive, which is a repository in which the non-dominated solutions found so far are stored. The solutions in the external archive are used as the leaders when the positions of the swarm particles have to be updated, and the external archive contents are also usually reported as the final algorithmic output (Reyes-Sierra & Coello, 2006); therefore, updating the external archives is a core issue for the MOPSO.

The main advantage of single-target PSO is the use of  $\epsilon$ -dominance (Coello, Pulido, & Lechuga, 2004). In MOPSO, however, this method is primarily used to filter and update the external archives, as shown in Figure 2. In an optimization problem in which the objective function minimizes both  $f_1$  and  $f_2$ , the particles in the external archives are analyzed. First, particle 7 is inferior to all other particles because box  $(2\epsilon, 3\epsilon)$  is dominated by box  $(2\epsilon, 2\epsilon)$ . Particle 1 dominates Particle 2 because it is closer to the lower left corner of the point  $(\epsilon, 3\epsilon)$ . The same particle 3 is superior to particle 4, and particle 5 is superior to particle 6. Therefore, after each iteration, the external archive is filtered according to the  $\epsilon$ -dominance method. After the iteration, the optimized solution is randomly selected from among all externally archived particles.

The pseudo code for the MOPSO algorithm is shown in Figure 3 (Reyes-Sierra & Coello, 2006).

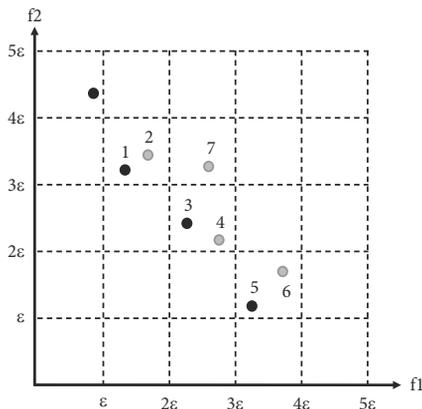


Figure 2.  $\epsilon$ -dominance MOPSO external archives

<p><b>Input:</b> Data set <math>D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}</math>;  The number of particle group <math>max</math>;  Training rounds <math>N</math>.  Initialize swarm  Initialize leader in an external archive  Quality (leaders)</p>
<p><b>Process:</b>  1: <b>for</b> <math>t = 1, 2, \dots, N</math>, do  2:   <b>for</b> <math>i = 1, 2, \dots, max</math> do  3:     Select leader  4:     Update Position  5:     Mutation  6:     Update pbest  9:   <b>end for</b>  10: Update leaders in the external archive  11: Quality (leaders)  12: <b>end for</b></p> <p><b>Output:</b> the optimal particle is randomly selected from the repository.</p>

Figure 3. Pseudocode for a general MOPSO algorithm

## 2. Proposed credit scoring model

The essence of the credit scoring model is the data mining and machine learning of large quantities of historical information to ensure the most accurate prediction of the loan applicant default probabilities. After the prediction result is obtained, the prediction value is compared with the true value to evaluate the performance of the credit scoring model using the AUC. The AUC indicator does not need to set the threshold in advance, which avoids the impact of the threshold on the classification result. It is the comprehensive indicator used to evaluate model performance. Therefore, this paper uses the AUC as one of the model optimization goals.

However, one of the AUC indicator assumptions is that different classification errors have consistent costs. This is unreasonable in some realistic credit scoring questions. In the credit scoring question, the cost of recognizing a possible defaulter as a non-defaulter is greater than recognizing a non-defaulter as a defaulter. If only the AUC is used as the optimization goal of the model, the proposed model has the same ability to recognize the defaulters and the non-defaulters, which would increase the total classification error cost of the model. However, in the realistic credit business, the total cost is closely related to the investor's revenue. In order to reduce the total classification error cost, the model would have a better ability to identify defaulters. Therefore, while considering the comprehensive performance of the classifier (evaluated in AUC), we cannot ignore the total classification error cost. Therefore, this paper uses the total classification error cost as one of the optimization goals. This paper describes how to construct a cost-sensitive loss function in the following sections.

From the introduction in section 2.1, logistic regression uses a maximum likelihood function to solve the parameter:

$$L(x, \beta) = \prod_{i=1}^n f(g(x_i, \beta))^{y_i} \cdot (1 - f(g(x_i, \beta)))^{(1-y_i)}. \quad (7)$$

The logarithm of the above formula can be obtained:

$$\ln L(x, \beta) = \sum_{i=1}^n (y_i \cdot \log(f(g(x_i, \beta))) + (1 - y_i) \cdot \log(1 - f(g(x_i, \beta)))) \tag{8}$$

If the deviation of the predicted value from the true value is defined as the loss, the parameter can also be solved from a loss minimization perspective. Then, the problem is to find the optimal parameters by minimizing the given cost function. In logistic regression, the cost function  $J(\theta)$  usually refers to the negative logarithm of the likelihood; therefore, maximizing  $\ln L(x, \beta)$  is equivalent to minimizing  $J(\theta)$ .

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N (-y_i \log(f(g(x_i, \beta))) - (1 - y_i) \log(1 - f(g(x_i, \beta)))) \tag{9}$$

Function  $J(\theta)$  constructed in this way then becomes a logarithmic loss function. The characteristic of  $J(\theta)$  is that when the actual sample  $y_i = 1$  and the predicted probability  $p_i = 1$ ,  $J(\theta) = 0$ ; however,  $J(\theta)$  increases as  $p_i$  decreases, as shown in Figure 4. If the threshold selected is 0.5, when  $p_i$  is greater than 0.5, the classification cost is  $F_{TP}$ , and when it is less than 0.5, the classification error cost is  $F_{FN}$ . When the actual sample  $y_i = 0$  and the predicted probability  $p_i = 0$ ,  $J(\theta) = 0$ ; however,  $J(\theta)$  increases as  $p_i$  increases, as shown in Figure 5. If the threshold selected is 0.5, when  $p_i$  is less than 0.5, the classification cost is  $F_{TN}$ , and when it is greater than 0.5, the classification error cost is  $F_{FP}$ .

If the threshold is not determined in advance, the logarithmic loss function can effectively measure the deviation between the predicted value and the true value; that is, the greater the deviation, the greater the loss function value. However, the above logarithmic loss function does not reflect cost-sensitive thinking.  $-y_i \log(p_i)$  is the cost of classifying the positive category into the negative category and  $-(1 - y_i) \log(1 - p_i)$  is the cost of classifying the negative category into the positive category, but it can be seen that the integrals of the two curves are the same. Therefore, the cost function indicates that the different classification errors costs are the same and needs to be improved. The different classification error costs need to be given different weights.

The cost matrix shows that the cost of classifying a positive category into a negative category is  $c_{FN}$ , and the cost of classifying a negative category into a positive category is  $c_{FP}$ . For the loss function  $J(\theta)$ ,  $-y_i \log(p_i)$  is the cost of classifying a positive category into a negative category and  $-(1 - y_i) \log(1 - p_i)$  is the cost of classifying a negative category into a positive category. Therefore, the cost-sensitive loss function for the logistic regression is:

$$Cost_i(\theta) = -c_{FN} y_i \log(f(g(x_i, \beta))) - c_{FP} (1 - y_i) \log(1 - f(g(x_i, \beta))) \tag{10}$$

This method is equivalent to directly increasing the area under the loss function to increase the cost of the classification error; that is, instances with higher costs receive a higher average loss value. As the smaller the value of  $Cost_i(\theta)$ , the smaller the classification error cost, and the greater the profit, the cost-sensitive loss function has practical business significance. Therefore, minimizing the cost-sensitive loss function is taken as one of the optimization goals in this paper when constructing the multi-objective optimization cost-sensitive logistic regression.

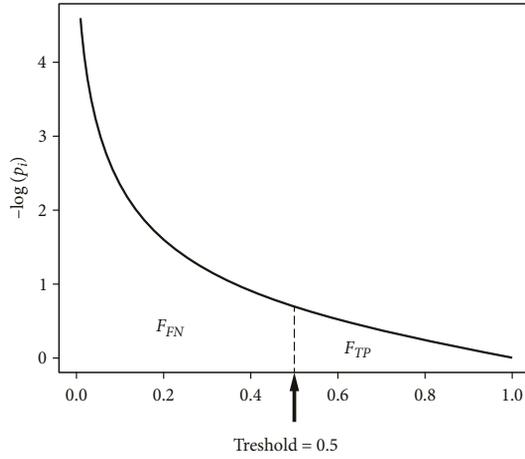


Figure 4. Cost loss function graph for  $y_i = 1$

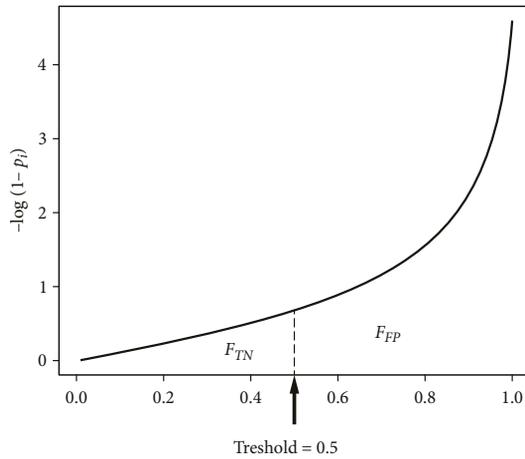


Figure 5. Cost loss function graph for  $y_i = 0$

Therefore, a cost-sensitive loss function is constructed based on the cost matrix and the logarithmic loss function, with the maximization of the AUC and the minimization of the cost-sensitive loss function being the two goals for determining the parameters for the proposed model. The constructed model is as follows:

$$\begin{cases} \max. AUC = \frac{\sum_{i \in \text{positive class}} \text{rank}(p_i) - \frac{M(1+M)}{2}}{M * N} , \\ \min. Cost_i(\theta) = -c_{FN} y_i \log(p_i) - c_{FP} (1 - y_i) \log(1 - p_i) \end{cases} \quad (11)$$

of which,  $p_i = P(y = 1|X_i) = \frac{1}{1 + e^{-g(x,\beta)}}$ .

The multi-objective optimization algorithm MOPSO is then used to solve the proposed model parameters. As mentioned, MOPSO uses an  $\varepsilon$ -dominance method to update the external archive when searching for the optimal parameter set. In the proposed model, the horizontal axis of the external archive is the total costs of the classification errors, and the vertical axis is the inverse AUC value. During each iteration, the mesh closest to the origin in the entire mesh and the particles closest to the lower left corner in each mesh are selected and stored in the external archive. The position and velocity of each particle are updated using the formula before proceeding to the subsequent iteration, after which the particles in the external archive are output, from which the optimal solution is randomly selected.

The algorithmic steps for solving the multi-objective optimization cost-sensitive logistic regression parameter values using MOPSO are as follows (Figure 6):

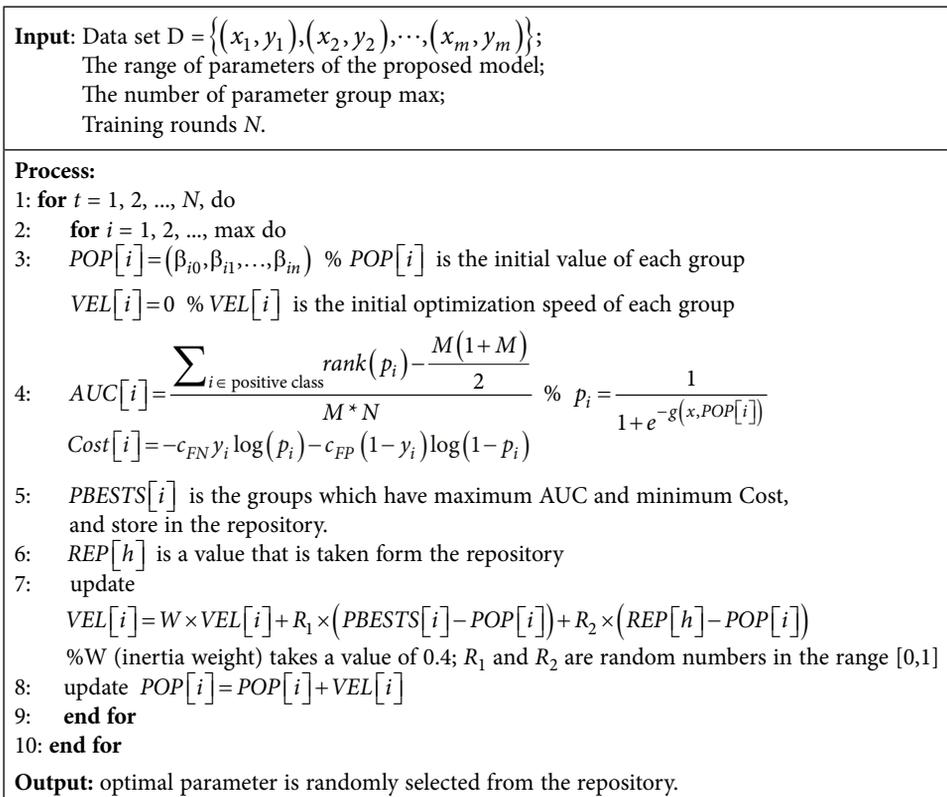


Figure 6. Pseudocode for the proposed model solved using the MOPSO algorithm

### 3. Empirical study

#### 3.1. Dataset description

The credit data set used in this paper was collected from a well-known P2P lending platform in China, for which the definition of a default loan is when the repayment is at least three times overdue (Greene, 1998); otherwise, the loans are classified as non-default. After delet-

ing the credit target data, 5000 defaults and 10000 non-default loans were randomly selected as the experimental data for this paper; the default rate was about 33.33% (5000/15000). Each loan instance in the credit data had 22 variables, which included 21 observation variables and the loan status labels. The input variables for each sample included the borrower's basic information, work information, credit certificate information, and asset information. Because there are decision trees and Bayes in the selected model, the selected variables were discretized in this paper. The definition for each variable is given in Table 3.

Table 3. Variable definitions

	Variable	Name	Definition
basic information	Gender	Gender	1 if an application is a man and 2 otherwise
	Age	Age	Age of the application in years
	education level	Edu	Education level of borrower, 1 = middle/high school, 2 = 3-year college, 3 = 4-year college, 4 = graduate school
	Marital status	Marriage	1 = married, 2 = unmarried, 3 = divorced, 4 = widowed
Loan information	Use of the loan	Purpose	1 = short-term turnover, 2 = personal consumption, 3 = buy a house, 4 = buy a house, 5 = otherwise
	Repayment type	type	1 = Equal principal and interest / monthly repayment, 2 = one-time debt service
	Loan amount (in RMB)	amount	The loan amount requested by the application
	Borrowing interest rate (in %)	interate	The rate that the application pays on the loan
	Loan period (in year)	term	Loan term requested by the application
Asset information	monthly income (in RMB)	income	Income level of the borrower, 1 = less than 1000, 2 = 1001–2000, 3 = 2001–5000, 4 = 5001–10000, 5 = 10001–20000, 6 = 20001–50000, 7 = more than 50000
	Real estate	house	1 if an application is a homeowner and 2 otherwise
	Mortgage	House_loan	1 if an application has a mortgage and 2 otherwise
	Car asset	car	1 if an application is a car owner and 2 otherwise
	Car loans	Car_loan	1 if an application has a car loan and 2 otherwise
Work information	Industry	work	Dummy variable
	Company type	Com_type	Dummy variable
	Company size	size	1 = less than 10 persons, 2 = 10–100, 3 = 100–500, 4 = more than 500
	Working time	length	1 = less than 1 year, 2 = 1–3 years, 3 = 3–5 years, 4 = more than 5 years
Approval Status	Credit certification	Credit_c	1 if application passed credit certification and 0 otherwise
	Identity certification	Identity_c	1 if application passed identity certification and 0 otherwise
	Work certification	Work_c	1 if application passed work certification and 0 otherwise
	Income certification	Income_c	1 if application passed income certification and 0 otherwise

The descriptive statistical analysis of these variables is shown in Table 4.

Table 4. Descriptive statistical analysis of variables

Variable	mode	P25	median	P75	Variable	mode	P25	median	P75
gender	1	1	1	1	car	1	1	2	3
age	35	31	36	42	car_loan	4	3	4	5
marriage	1	1	1	2	work	1	1	1	2
edu	2	2	2	3	com_type	2	2	2	2
purpose	1	1	2	3.25	size	2	1	2	2
type	2	2	2	2	length	2	2	2	2
amount	6	5	6	7	credit_c	1	1	1	1
interate	2	2	2	2	income_c	1	1	1	1
term	36	12	18	36	indentity_c	1	1	1	1
income	11	3	8	11	work_c	1	1	1	1
house	1	1	2	2	label	1	0	1	1
house_loan	4	2	4	4					

### 3.2. Expected loss

The cost matrix for the research dataset was developed before credit scoring. The cost matrix can describe the default risk in the lending process (Nayak & Misra, 2018). If credit agencies divide non-defaulters into possibly loan defaulters, it is possible that they could lose loan interest; however, if they divide possible loan defaulters into non-defaulters, they could lose both loan interest and principal, which is usually measured as the expected loss (EL) and is calculated as shown in the following formula (Altman & Sabato, 2005; Thomas, 2010):

$$EL = PD * EAD * LGD, \tag{12}$$

where PD is the possibility of a loan applicant defaulting, and EAD is the exposure at default or the total amount owed at default and is expressed in monetary units, and LGD is the loss given default, which indicates the borrower’s financial losses after default and is expressed as a percentage.

$$LGD\% = \frac{EAD - NPV(\text{Recoveries} - \text{costs})}{EAD}. \tag{13}$$

The loan amount and loan interest for a non-defaulter in the training set are used to calculate the possible interest income that could be lost when a non-defaulting applicant is judged as a possible defaulter. When the model recognizes that the non-defaulter as a defaulter, the interest income would be lost, corresponding to the value of  $c_{FN}$  in the cost matrix. The expected loss (EL) is calculated using the total default amount, the recovery amount, and the recovery costs of a defaulter sample in the training set. When the model recognizes that the defaulter as a non-defaulter, the loss is calculated from the EL, corresponding to the value of  $c_{FP}$  in the cost matrix. As the value of the EL is 3.22 times the interest loss, the cost of recognizing a possible defaulter as a non-defaulter is 3.22 times greater than recognizing a non-defaulter as a possible defaulter in the research dataset.

### 3.3. Experimental design

K-fold Cross-Validation was used to train the training set and ensure the stability of the experimental results as it makes full use of the dataset to test the algorithmic effect, thereby avoiding over-fitting. The basic idea is to randomly divide the dataset into k parts, one of which is used as a test set and the remaining k-1 used as the training set. The experimental process is shown in Figure 7.

This paper used a 10-fold cross-validation to train the model; that is, the dataset was divided into 10 parts, 10 tests were performed, and the average result of the ten tests used to evaluate the model performance.



Figure 7. Demonstration of K-fold cross-validation

### 3.4. Experimental results

The empirical results were comprehensively evaluated using type I error rate, type II error rate, F value, G-mean, accuracy, AUC, and total cost (TC) indicators. The definitions and calculation formulas for the above indicators are described in the following.

The type I error rate indicates the ratio of false negatives to all positive examples (Tsai, 2009).

$$\text{type I error rate} = \frac{FN}{FN + TP}. \tag{14}$$

The type II error rate indicates the ratio of false positives to all counterexamples (Tsai, 2009).

$$\text{type II error rate} = \frac{FP}{FP + TN}. \tag{15}$$

The total cost (TC) indicator is all error classification results calculated by combining the cost matrix.

$$\text{total cost}(TC) = c_{FN} * FN + c_{FP} * FP. \tag{16}$$

The F1 indicator is a harmonic mean with Recall (also known as TPR) and Precision, which is presented as Eq. (19). The Recall and Precision are respectively calculated using Eq. (17) and Eq. (18) (Baldi, Brunak, Chauvin, Andersen, & Nielsen, 2000).

$$\text{Recall}(TPR) = \frac{TP}{TP + FN}; \tag{17}$$

$$\text{Precision} = \frac{TP}{TP + FP}; \tag{18}$$

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{19}$$

The G-Mean (Geometric mean) indicator is a comprehensive evaluation method for im-balance dataset (Shen, Zhao, Z. Li, K. Li, & Meng, 2019), and is determined using Eq. (20). A higher G-Mean indicates that the balance between the classes is reasonable and has good performance in the binary classification model.

$$G - \text{mean} = \sqrt{\frac{TP}{TP + FN} * \frac{TN}{TN + FP}}. \tag{20}$$

From the collected data, 15000 samples were selected for training and prediction, and a 10-fold Cross-Validation used to prevent model overfitting. To assess the performance of the proposed multi-objective optimization cost-sensitive logistic regression, it was compared with other explanatory models; traditional logistic regression, decision tree, linear discrimi-nant, Bayes, SVM, Bagging, Adaboost, cost-sensitive decision tree (CS-DT). The experimen-tal results are shown in Table 5 and Table 6.

Table 5. Experimental results from the research dataset

Indicators	error I	error II	F1	G-mean	Accuracy	TPR	Precision
Traditional logistic	0.144	0.320	0.8488	0.7626	0.7969	0.856	0.842
Decision Tree	0.230	0.135	0.8381	0.8161	0.8017	0.770	0.920
Linear discrimination	0.151	0.298	0.8496	0.7717	0.7997	0.849	0.850
Bayes	0.698	0.003	0.4627	0.5482	0.5336	0.302	0.996
SVM	0.155	0.159	0.8778	0.8428	0.8433	0.845	0.914
Bagging	0.219	0.121	0.8480	0.8283	0.8135	0.781	0.928
Adaboost	0.153	0.178	0.8747	0.8341	0.8384	0.847	0.905
CS-DT	0.225	0.141	0.8398	0.8159	0.8030	0.775	0.917
Proposed model	0.279	0.036	0.8289	0.8335	0.8018	0.721	0.976

Table 6. Significance test for the AUC and cost

Indicators	AUC	paired t test	TC	paired t test
Traditional logistic	0.8842	-1.49	945.4	31.19 <sup>***</sup>
Decision Tree	0.8558	-9.10 <sup>***</sup>	567.4	15.58 <sup>***</sup>
Linear discrimination	0.8841	-1.29	897.6	29.98 <sup>***</sup>
Bayes	0.8528	-10.80 <sup>***</sup>	704.8	30.76 <sup>***</sup>
SVM	0.9116	14.3 <sup>***</sup>	553.0	12.74 <sup>***</sup>
Bagging	0.8627	-6.23 <sup>***</sup>	522.2	15.60 <sup>***</sup>
Adaboost	0.9119	12.42 <sup>***</sup>	598.8	24.28 <sup>***</sup>
CS-DT	0.8568	-11.02 <sup>***</sup>	576.7	15.18 <sup>***</sup>
Proposed model	0.8844	-	369.3	-

Note: For t test, Proposed model is used as baseline; \*\*\* Significant at 1% level; \*\* Significant at 5% level; \* Significant at 10% level.

As can be seen from the experimental results, compared to the other classifiers, the proposed model had the smallest total classification error costs and a large AUC. Figures 8 and 9 show the box plots for the total costs and AUC obtained from the 10-fold cross-validation by the proposed model and the comparative models. The boxplots have six data nodes, which arrange the data from the largest to the smallest and then calculate the maximum, the upper quartile, the median, the lower quartile, the minimum, and the outliers; if there are no outliers, there are only five data nodes. As can be seen from Figures 8 and 9, the total cost distribution for the proposed model was lower in all experiments with the 10-fold cross-validation, and the maximum total cost in all experiments was lower than the minimum total cost of the other classifiers. Therefore, the proposed model performance clearly demonstrated that

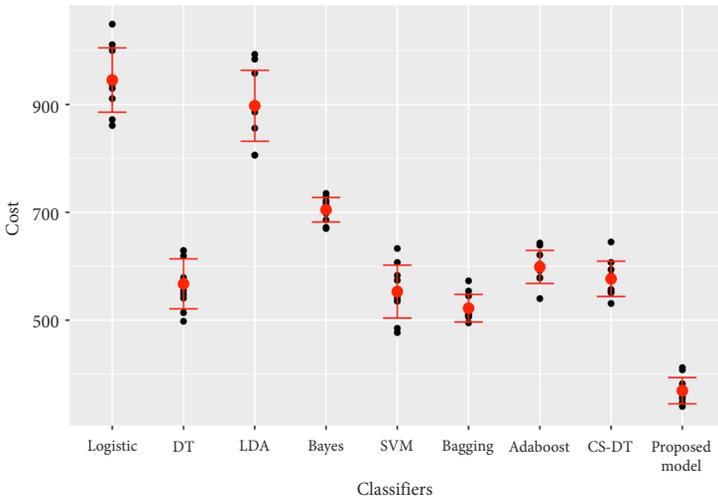


Figure 8. Total cost box plot

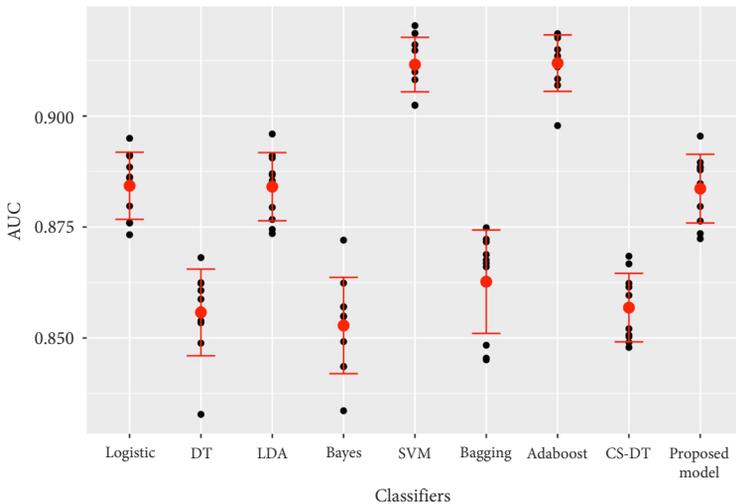


Figure 9. AUC box plot

it was effective in dealing with cost-sensitive issues. At the same time, the proposed model in this paper was found to have a large AUC, with the value being only lower than the SVM and Adaboost models. As the logistic regression used in this paper was a simple classifier and the data dimension was small, it was acceptable that the AUC for the proposed model was slightly lower than these two models.

To further verify that the AUC and cost indicators were better than the other comparison models, a paired t-test was performed, the results for which are shown in Table 6. As can be seen, the AUC for the proposed model was significantly higher than the Decision Tree, Bayes, Bagging, and the CS-DT, and the total cost was significantly lower than all other comparison models; therefore, the proposed model was found to be effective in dealing with cost-sensitive issues.

As can be seen from the Table 5, the empirical results showed that the type II error rate for the proposed model was significantly lower, which indicated that the model was better able to identify possible future loan defaulters. As this article defined non-defaulters as 1, FP indicated that the defaulters were classified as non-defaulters; therefore,  $\text{Precision} = \frac{TP}{TP + FP}$  was able to measure the ability and better identify defaulters. As can be seen from the Table 5, except for the Bayes model, the precision of the proposed model in this paper was the highest. However, the F1, G-mean, and Recall of Bayes were low, indicating that while the model developed in this article was better at identifying the defaulters, the ability to identify non-defaulters was not reduced. Figure 10 shows about 100 defaulted samples that the proposed model sample classifications were correct but that the traditional logistic regression classifications were incorrect. The black dots show the traditional logistic regression prediction results and the blue dots show the proposed model prediction results. The true label for these samples was 0 (loan defaulters). When traditional logistic regression was used to classify these samples, as the prediction probability was greater than the 0.5 threshold, these samples were judged as non-defaulters. However, when the proposed model was used to classify these samples, the prediction probability was less than 0.5, and the samples were judged as probable future loan defaulters. Therefore, the proposed model classifications were correct. If the

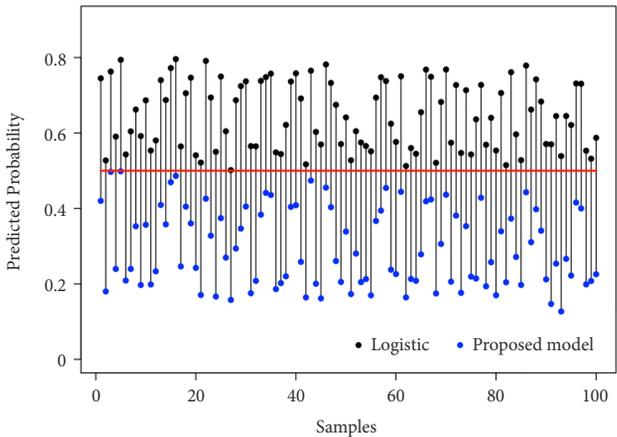


Figure 10. Some examples of the prediction probabilities for the two models

credit scoring model can better identify loan defaulters, then credit institutions would be able to reduce their default loans and improve their overall profit performances.

It can be seen from the F value, G-mean, and accuracy that the proposed model had a large G-mean and accuracy and a large F1 value, as shown in the line graph in Figure 11. As can be seen, the G-mean value of the proposed model is slightly lower than SVM and Adaboost. As the G-mean is a key indicator for the evaluation of unbalanced data performance, a high G-mean indicates that the model performed better on the unbalanced data sets.

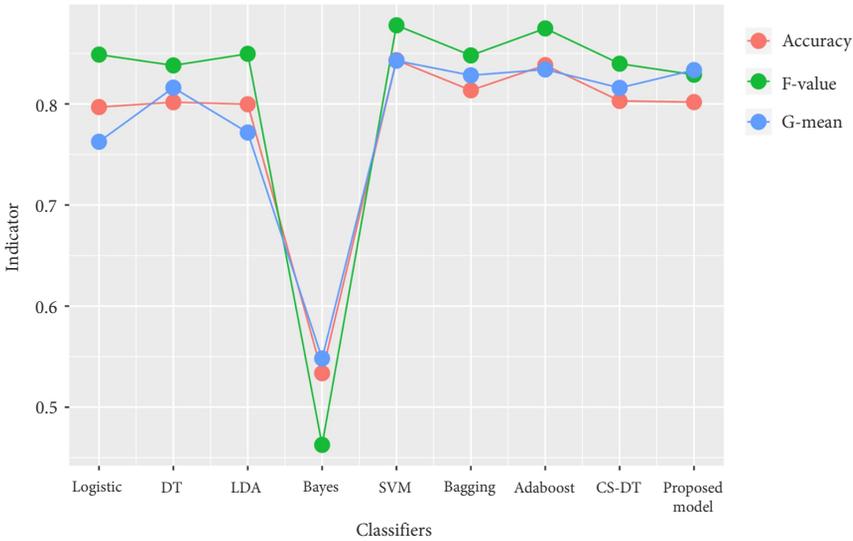


Figure 11. F value, G-mean and accuracy line graph

### 3.5. Discussion and Robust test

This section first compares the results of the proposed multi-objective model with the results of the single goal, which the PSO method is used to estimate the logistic regression parameters based on minimum cost or maximum AUC. The comparative experimental results are shown in Table 7, where PSO(Cost) represents the experimental results of the PSO optimization based on the minimum cost, and PSO(AUC) represents the experimental results of the PSO optimization based on the maximum AUC.

Table 7. Experimental results of performance comparison

Indicators	AUC	paired t test	TC	paired t test
PSO(Cost)	0.8836	-0.84	463.6	2.97**
PSO(AUC)	0.8871	3.99***	850.7	6.40***
Proposed model	0.8844	-	369.3	-

Note: For t test, Proposed model is used as baseline; \*\*\* Significant at 1% level; \*\* Significant at 5% level; \* Significant at 10% level.

It can be seen from the experimental results that the results based on multi-objective optimization had the lowest total cost compared to the single-objective optimization. Although the AUC based on the multi-objective optimization was significantly lower than the result of the AUC-based single-objective optimization by 3%, the total cost based on the multi-objective optimization was less than half. The box plots of the corresponding comparisons are shown in Figures 12 and 13. As can be seen from Figure 12, the total cost based on the multi-objective optimization was the lowest and had the most stable distribution. At the same

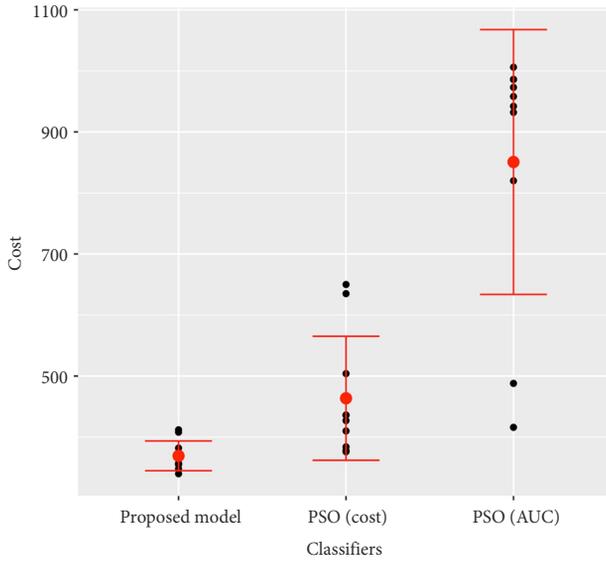


Figure 12. Total cost boxplot

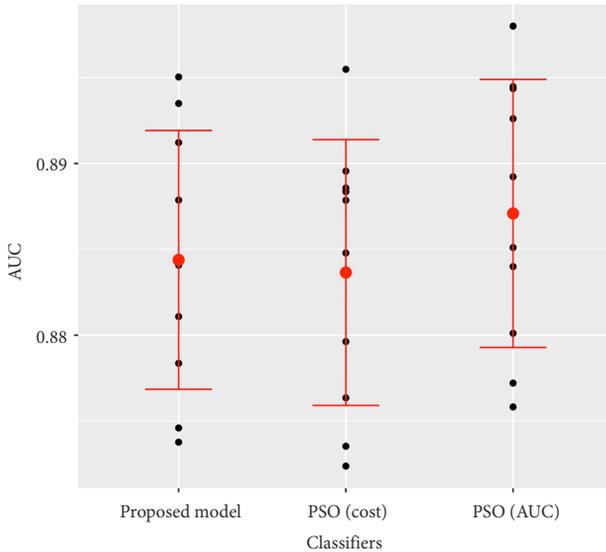


Figure 13. AUC boxplot

time, it can be seen from the results of the significance test that the cost of the multi-objective optimization model proposed in this paper is significantly lower than that of single-objective optimization. Therefore, it can be seen from the above analysis that the model had a better practical effect based on the multi-objective optimization than on the single-objective optimization. When using the credit scoring model to decide whether to issue a loan in practice, it is necessary to pay greater attention to the total cost indicator.

The multi-objective optimization results of the MOPSO were also compared with the NSGA-II (Deb, Agrawal, Pratap, & Meyarivan, 2000), PAES (Knowles & Corne, 2000), and microGA (Coello & Pulido, 2001), they are common multi-objective optimization algorithms. The experimental results of performance comparison are shown in Table 8.

It can be seen from the experimental results that the results based on MOPSO had the largest AUC, while the total cost is significantly lower than the other multi-objective optimization methods. Therefore, it can be seen from the above indicators that determining the optimal solution based on MOPSO produced the best results for the constructed multi-objective credit scoring model.

As this paper's focus was on the total cost and AUC indicators, this section uses the box plot to show the results of the 10-fold cross-validation. The total cost and AUC box plots for the MOPSO and the other multi-objective optimization methods are shown in Figures 14 and 15. As can be seen from Figure 14, the total cost based on the MOPSO was the lowest and had the most stable distribution. Figure 15 shows that the MOPSO optimization also performed better than the other three multi-objective optimization algorithms on AUC.

Further, to verify the robustness of the proposed model, this paper used a dataset from a "lending club" website for the empirical analysis. Lending club data with a credit period of three years were downloaded from the associated website in 2015. After missing value processing and variable selection, 420,000 data points and 17 characteristic variables were obtained, after which 20000 samples; 10000 non-defaulters and 10000 defaulters; were randomly extracted for the empirical research. The empirical results are shown in Table 9.

From the experimental results in Table 9, it can be seen that the AUC for the proposed model in this paper was significantly higher than the decision tree, Bayes, SVM, Bagging, and CS-DT, and there was no significant difference between the LDA and Adaboost. The proposed model was therefore found to have a better classification performance. From a total cost perspective, the total cost for the model constructed in this paper was significantly lower than all other comparison models; therefore, the model proposed in this paper had better classification results for the lending club dataset.

Table 8. Experimental results of performance comparison

Indicators	AUC	Paired t test	TC	Paired t test
NSGA-II	0.8838	-0.186	416.9	4.95***
PAES	0.8840	-0.400	898.6	16.60***
MicroGA	0.8844	0.803	448	30.76***
MOPSO	0.8844	-	369.3	-

Note: For t test, Proposed model is used as baseline; \*\*\* Significant at 1% level; \*\* Significant at 5% level; \* Significant at 10% level.

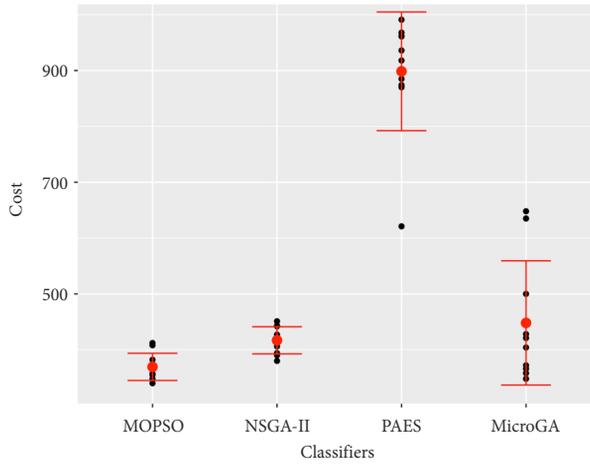


Figure 14. Total cost boxplot

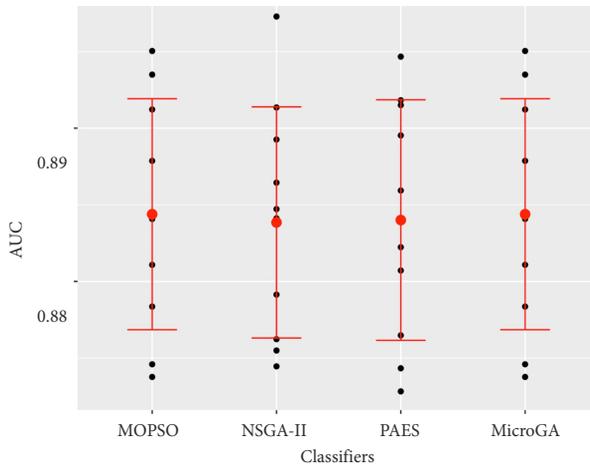


Figure 15. AUC boxplot

Table 9. Experimental results from the lending club dataset

Indicators	AUC	paired t test	TC	paired t test
Traditional logistic	0.702	0.70	497.70	15.02***
Decision Tree	0.637	-18.71***	419.29	7.60***
Linear discrimination	0.702	1.23	499.01	15.48***
Bayes	0.673	-7.99***	414.82	7.87***
SVM	0.696	-2.04**	490.20	18.30***
Bagging	0.675	-12.96***	604.01	29.65***
Adaboost	0.696	-1.64	489.96	16.37***
CS-DT	0.650	-15.90***	501.81	17.72***
Proposed model	0.701	-	367.67	-

Note: For the t test, the proposed model was used as the baseline; \*\*\* Significant at 1% level; \*\* Significant at 5% level; \* Significant at 10% level.

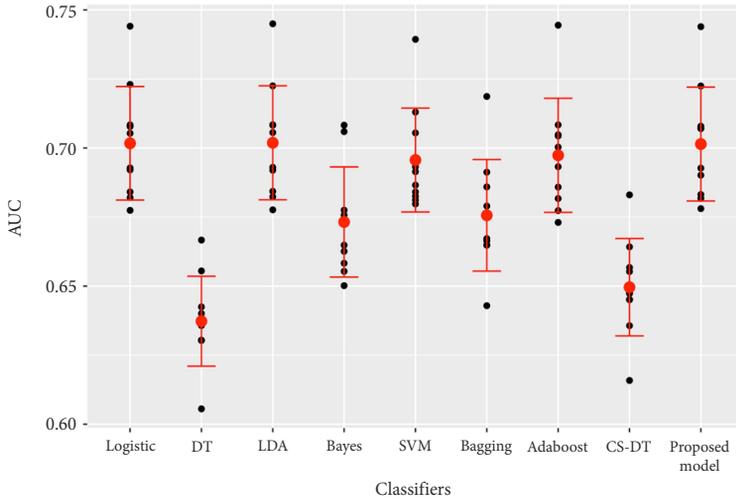


Figure 16. AUC boxplot for the lending club dataset

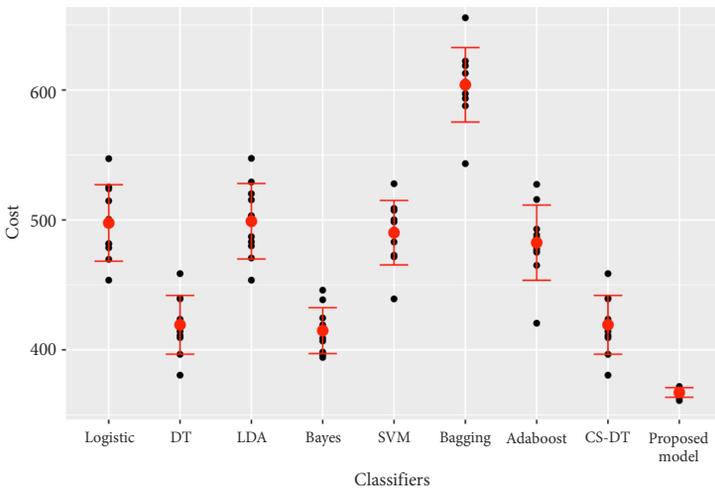


Figure 17. Total cost boxplot for the lending club dataset

Figures 16 and 17 show the results for the ten-fold cross-validation, from which it can be seen that the AUC indicator was on a higher level, the total cost indicator was on the lowest level, and the distribution was the most stable. Therefore, the proposed model in this paper was shown to have a certain robustness.

### Conclusion and future work

Credit scoring is an important process for P2P lending companies as it determines the probability of loan applicant defaults. Traditional credit scoring models only focus on classification error rates and the different classification errors are assumed to attract the same costs.

However, this is unrealistic, there is an inherent cost-sensitive problem in the credit scoring process. Therefore, in this paper, a new cost-sensitive logistic regression credit scoring model based on multi-objective optimization approach was proposed, in which there were two objectives in the logistic regression: maximizing the AUC and minimizing the total classification error costs. An empirical analysis applied the proposed model to the credit scoring of a Chinese famous P2P company, from which it was found that compared to traditional credit scoring models, the proposed model was able to significantly reduce the type II error rate and the total classification error costs. This paper then compared the proposed model with single-objective optimization methods and other multi-objective optimization approaches, which clearly showed that the MOPSO was the best multi-objective optimization approach for cost-sensitive logistic regression credit scoring model. In future research, we plan to integrate cost-sensitive problems into other base classifiers to develop other direct cost-sensitive credit scoring models. Ensemble classifiers have become more popular in recent years because of the unstable and inaccurate results from individual classifiers. We will also incorporate cost-sensitivity into an ensemble model to develop a cost-sensitive ensemble learning approach.

## Acknowledgements

This research was supported by the Humanities and Social Sciences Foundation of the Ministry of Education of China (no. 17YJC630119). It was also supported by the Fundamental Research Funds for the Central Universities (no. JBK1801009) and the project of Research Center for System Sciences and Enterprise Development (no. Xq18B07).

## References

- Abdou, H. A., Tsafack, M. D. D., Ntim, C. G., & Baker, R. D. (2016). Predicting creditworthiness in retail banking with limited scoring data. *Knowledge-Based Systems*, 103, 89-103. <https://doi.org/10.1016/j.knosys.2016.03.023>
- Abraham, T. W. (2018). Estimating the effects of financial access on poor farmers in rural northern Nigeria. *Financial Innovation*, 4(25), 1-20. <https://doi.org/10.1186/s40854-018-0112-2>
- Ala'raj, M., & Abbod, M. F. (2016). Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems*, 104, 89-105. <https://doi.org/10.1016/j.knosys.2016.04.013>
- Altman, E. I., & Sabato, G. (2005). Effects of the new Basel capital accord on bank capital requirements for SMEs. *Journal of Financial Services Research*, 28(1-3), 15-42. <https://doi.org/10.1007/s10693-005-4355-5>
- Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Example-dependent cost-sensitive decision trees. *Expert Systems with Applications*, 42(19), 6609-6619. <https://doi.org/10.1016/j.eswa.2015.04.042>
- Bahnsen, A. C., Aouada, D., & Ottersten, B. (2014, December). Example-dependent cost-sensitive logistic regression for credit scoring. In *2014 13th International conference on machine learning and applications* (pp. 263-269). Detroit, MI, USA: IEEE. <https://doi.org/10.1109/ICMLA.2014.48>
- Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5), 412-424. <https://doi.org/10.1093/bioinformatics/16.5.412>
- Bequé, A., Coussement, K., Gayler, R., & Lessmann, S. (2017). Approaches for credit scorecard calibration: An empirical analysis. *Knowledge-Based Systems*, 134, 213-227. <https://doi.org/10.1016/j.knosys.2017.07.034>

- Chen, Z., Li, Y., Wu, Y., & Luo, J. (2017). The transition from traditional banking to mobile internet finance: an organizational innovation perspective—a comparative study of Citibank and ICBC. *Financial Innovation*, 3(12), 1-16. <https://doi.org/10.1186/s40854-017-0062-0>
- Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 256-279. <https://doi.org/10.1109/TEVC.2004.826067>
- Coello, C. A. C. C., & Pulido, G. T. (2001, March). A micro-genetic algorithm for multiobjective optimization. In *International conference on evolutionary multi-criterion optimization* (pp. 126-140). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/3-540-44719-9\\_9](https://doi.org/10.1007/3-540-44719-9_9)
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000, September). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature* (pp. 849-858). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/3-540-45356-3\\_83](https://doi.org/10.1007/3-540-45356-3_83)
- Desai, V. S., Crook, J. N., & Overstreet Jr, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, 95(1), 24-37. [https://doi.org/10.1016/0377-2217\(95\)00246-4](https://doi.org/10.1016/0377-2217(95)00246-4)
- Ding, S., Chen, C., Xin, B., & Pardalos, P. M. (2018). A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches. *Applied Soft Computing*, 63, 249-267. <https://doi.org/10.1016/j.asoc.2017.09.012>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- García, V., Marqués, A. I., & Sánchez, J. S. (2012, November). Improving risk predictions by preprocessing imbalanced credit data. In *International conference on neural information processing* (pp. 68-75). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-34481-7\\_9](https://doi.org/10.1007/978-3-642-34481-7_9)
- Greene, W. (1998). Sample selection in credit-scoring models. *Japan and the World Economy*, 10(3), 299-316. [https://doi.org/10.1016/S0922-1425\(98\)00030-9](https://doi.org/10.1016/S0922-1425(98)00030-9)
- Günemann, N., & Pfeffer, J. (2017, May). Cost matters: a new example-dependent cost-sensitive logistic regression model. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 210-222). Cham: Springer. [https://doi.org/10.1007/978-3-319-57454-7\\_17](https://doi.org/10.1007/978-3-319-57454-7_17)
- Guo, Y., Zhou, W., Luo, C., Liu, C., & Xiong, H. (2016). Instance-based credit risk assessment for investment decisions in P2P lending. *European Journal of Operational Research*, 249(2), 417-426. <https://doi.org/10.1016/j.ejor.2015.05.050>
- Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., & Baesens, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1), 141-154. <https://doi.org/10.1016/j.dss.2010.12.003>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: Springer. <https://doi.org/10.1007/978-1-4614-7138-7>
- Kao, L. J., Chiu, C. C., & Chiu, F. Y. (2012). A Bayesian latent variable model with classification and regression tree approach for behavior and credit scoring. *Knowledge-Based Systems*, 36, 245-252. <https://doi.org/10.1016/j.knsys.2012.07.004>
- Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Systems with Applications*, 37(9), 6233-6239. <https://doi.org/10.1016/j.eswa.2010.02.101>
- Kim, J., Choi, K., Kim, G., & Suh, Y. (2012). Classification cost: An empirical comparison among traditional classifier, Cost-Sensitive Classifier, and MetaCost. *Expert Systems with Applications*, 39(4), 4013-4019. <https://doi.org/10.1016/j.eswa.2011.09.071>
- Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2), 149-172. <https://doi.org/10.1162/106365600568167>

- Kou, G., Chao, X., Peng, Y., Alsaadi, F. E., & Herrera-Viedma, E. (2019). Machine learning methods for systemic risk analysis in financial sectors. *Technological and Economic Development of Economy*, 25(5), 716-742. <https://doi.org/10.3846/tede.2019.8740>
- Ling, C. X., & Sheng, V. S. (2011). Cost-sensitive learning. In *Encyclopedia of machine learning* (pp. 231-235). Boston, MA: Springer. [https://doi.org/10.1007/978-0-387-30164-8\\_181](https://doi.org/10.1007/978-0-387-30164-8_181)
- Marqués, A. I., García, V., & Sánchez, J. S. (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society*, 64(7), 1060-1070. <https://doi.org/10.1057/jors.2012.120>
- Min, J. H., & Lee, Y. C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications*, 28(4), 603-614. <https://doi.org/10.1016/j.eswa.2004.12.008>
- Nayak, S. C., & Misra, B. B. (2018). Estimating stock closing indices using a GA-weighted condensed polynomial neural network. *Financial Innovation*, 4(21), 1-22. <https://doi.org/10.1186/s40854-018-0104-2>
- Rashid, A., & Jabeen, N. (2018). Financial frictions and the cash flow–external financing sensitivity: evidence from a panel of Pakistani firms. *Financial Innovation*, 4(15), 1-20. <https://doi.org/10.1186/s40854-018-0100-6>
- Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3), 287-308. <https://doi.org/10.5019/j.ijcir.2006.68>
- Serrano-Cinca, C., & Gutiérrez-Nieto, B. (2016). The use of profit scoring as an alternative to credit scoring systems in peer-to-peer (P2P) lending. *Decision Support Systems*, 89, 113-122. <https://doi.org/10.1016/j.dss.2016.06.014>
- Shen, F., Zhao, X., Li, Z., Li, K., & Meng, Z. (2019). A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Physica A: Statistical Mechanics and its Applications*, 526, 121073. <https://doi.org/10.1016/j.physa.2019.121073>
- Thomas, L. C. (2010). Consumer credit models pricing, profit and portfolios. *Journal of the Royal Statistical Society*, 173(2), 468-468. [https://doi.org/10.1111/j.1467-985X.2009.00634\\_12.x](https://doi.org/10.1111/j.1467-985X.2009.00634_12.x)
- Tsai, C. F. (2009). Feature selection in bankruptcy prediction. *Knowledge-Based Systems*, 22(2), 120-127. <https://doi.org/10.1016/j.knosys.2008.08.002>
- Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research*, 238(2), 505-513. <https://doi.org/10.1016/j.ejor.2014.04.001>
- Wang, G., Ma, J., Huang, L., & Xu, K. (2012). Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, 26, 61-68. <https://doi.org/10.1016/j.knosys.2011.06.020>
- Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *Journal of Financial and Quantitative Analysis*, 15(3), 757-770. <https://doi.org/10.2307/2330408>
- Xia, Y., Liu, C., & Liu, N. (2017). Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending. *Electronic Commerce Research and Applications*, 24, 30-49. <https://doi.org/10.1016/j.elerap.2017.06.004>
- Yu, L., Yue, W., Wang, S., & Lai, K. K. (2010). Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Systems with Applications*, 37(2), 1351-1360. <https://doi.org/10.1016/j.eswa.2009.06.083>
- Zhu, X., Li, J., Wu, D., Wang, H., & Liang, C. (2013). Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-TOPSIS classification approach. *Knowledge-Based Systems*, 52(6), 258-267. <https://doi.org/10.1016/j.knosys.2013.08.004>