Scientific Research Publishing

Intelligent Information Management, 2021, 13, 251-267
https://www.scirp.org/journal/iim
ISSN Online: 2160-5920
ISSN Print: 2160-5912

Adaptive Optimization Swarm Algorithm Ensemble Model Applied to the Classification of Unbalanced Data

Qingqing He[1], Chao Qin[2]

[1]School of Business Administration, Shandong University of Finance and Economics, Jinan, China
[2]School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China
Email: hq-mm@163.com

How to cite this paper: He, Q.Q. and Qin, C. (2021) Adaptive Optimization Swarm Algorithm Ensemble Model Applied to the Classification of Unbalanced Data. *Intelligent Information Management*, **13**, 251-267.
https://doi.org/10.4236/iim.2021.135014

Received: June 27, 2021
Accepted: September 13, 2021
Published: September 16, 2021

Abstract

In order to solve the problem that the hyper-parameters of the existing random forest-based classification prediction model depend on empirical settings, which leads to unsatisfactory model performance. We propose a based on adaptive particle swarm optimization algorithm random forest model to optimize data classification and an adaptive particle swarm algorithm for optimizing hyper-parameters in the random forest to ensure that the model can better predict unbalanced data. Aiming at the premature convergence problem in the particle swarm optimization algorithm, the population is adaptively divided according to the fitness of the population, and an adaptive update strategy is introduced to enhance the ability of particles to jump out of the local optimum. The main steps of the model are as follows: Normalize the data set, initialize the model on the training set, and then use the particle swarm optimization algorithm to optimize the modeling process to establish a classification model. Experimental results show that our proposed algorithm is better than traditional algorithms, especially in terms of F1-Measure and ACC evaluation standards. The results of the six-keel imbalanced data set demonstrate the advantages of our proposed algorithm.

Keywords

Random Forest, APSO, Unbalanced Data, Parameter Optimization

1. Introduction

The problem of unbalanced data classification often exists in the field of data classification, such as bioinformatics, intrusion detection system and classification problem [1] [2] [3] [4] [5]. And it has become one of the hot issues in recent years. Unlike balanced data, the number of samples in different categories in unbalanced data varies greatly. In general, the category with more samples in unbalanced data is called negative class, while the category with fewer samples is called positive class. With a small number, the information provided to the classifier is relatively less. On the contrary, there are more negative sample data, which can provide more information to the classifier. In the case of unbalanced classification of data sets, the standard classifier is usually unable to achieve good classification results. Unbalanced data set classification often appears in many practical applications. For example, compared with people with good credit, default samples are usually small and the identification target should be the default samples in credit scoring. A good classification model should be able to produce high recognition accuracy for the default application. Misclassification of positive samples in unbalanced data classification will lead to serious consequences. So, it is very important to choose a classification model that can deal with unbalanced data.

The most commonly used methods to solve the problem of class imbalance are 1) Resampling method [6], which through under-sampling and over-sampling methods to eliminate most class instances or increase a few class instances to change the original class distribution of unbalanced data; it would increase the misclassification of minority classes and loss information in general rules. 2) Cost-sensitive learning method [7] assigns different values to the misclassification costs of different categories, generally, the minority in the categories are expensive, and the cost of majority is low; the approach of a cost-sensitive classifier is to handle the problems with different error costs. It also might end up with over-specific rules. 3) Ensemble strategy, which improves the generalization performance of existing learning algorithms effective strategies, such as ensemble methods based on Bagging and Boosting. According to the famous "No Free Lunch Theorem" [8], a single classifier is not an effective solution for classification as the characteristics of different data are disparate due to the size of the data set, data structure, and features. The concept of ensemble learning is to combine multiple classifications, process different hypotheses to form a better hypothesis, and make predictions. Dietterich [9] explained the three basic reasons for the success of the ensemble method from a mathematical point of view: statistics, calculation and representativeness. Kearns and Valiant [10] proved that as long as there is enough data, single learning algorithms can generate arbitrarily high-precision estimates through the ensemble. These studies show that an ensemble classifier has better learning ability than a single classifier.

Thereinto, the Random Forest (RF) algorithm is a bagging ensemble learning algorithm based on the random subspace method by Breiman L. *et al.* [11]. This algorithm is a combined classification method. It is based on the Bootstrap sampling principle and randomly selects several different ones from the original data set. The advantage of RF is that it can handle a large number of data features; and generate unbiased estimates for generalized errors within the model; it can deal with the problem of data missing, especially for unbalanced classification data sets, RF can balance errors, and the algorithm is modeled in parallel, which runs fast. For imbalanced datasets, RF can balance errors. When there is a classification imbalance, RF can provide an effective method to balance the data set error. Alhudhaif and Adi [12] used RF to classify the EEG signals of landlords with unbalanced data distribution. An adaptive sampling method is used to stabilize each sample and then the RF is used to classify each balance block. The experimental results show that the RF effectively classifies unbalanced data signals. However, the above method cannot build a tree structure that can accommodate unbalanced data due to the normal poor setting of hyper-parameters. The performance of classification accuracy may be reduced if the model setting can't well organize the model to learn from a few classes. Therefore, it is an important problem to choose the best setting of hyper-parameter for unbalanced data. Artificial adjustment on parameters is time-consuming and laborious. The better performance of RF depends on the appropriate hyper-parameter settings. When in the data classification, the selection of hyper-parameters such as the maximum number of features used by a single decision tree and the number of sub-tree will directly control the tree structure of the model, which has a great impact on the performance of the classifier, unsuitable parameter values may lead to over-learning or under-learning. Especially in the face of unbalanced data sets, reasonable hyper-parameter settings can help the model to pay more attention to a small number of samples so that the model can more effectively balance the error.

In response to the problem of poor performance random forest model on unbalanced data due to unreasonable hyper-parameter setting, we used the adaptive particle swarm optimization (APSO)-RF model for data classification to obtain a high precision prediction. We use the idea of clustering [13] to adaptively divide the particle swarm into different populations and guide the populations by applying different update strategies. This enhances the diversity of particles and helps particles jump out of a local optimum. Through adaptive mechanisms, APSO is suitable for the parameter optimization of RF, and it improves the model prediction accuracy.

2. Related Work

In this section, we introduced the related works about techniques of RF and PSO.

2.1. Decision Tree

Classification and Regression Tree (CART) is an inductive learning algorithm for a single classification regressor, which is composed of root nodes, leaf nodes and non-leaf nodes. The decision tree generates a path from the root node to the leaf node through regression analysis on the training set and analyzes the path rules. Classify or predict new instance according to path rules. CART is based on information entropy and uses the Gini coefficient minimum principle index to split the node. The input space of the training set $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$ is divided into regions, and each sample is recursively divided into the corresponding region and a determined output value is obtained. The steps of the algorithm are as follows:

1) Assuming that the characteristic of the independent variable is *j*, the value of this characteristic is *s*. Assuming that the value *s* divides the space of feature *j* into two regions, the formula is as follows:

$$R_1(j,s)=\{x\,|\,x^{(j)}\le s\},\quad R_2(j,s)=\{x\,|\,x^{(j)}>s\} \tag{1}$$

2) Traverse and calculate the loss function(*LF*) of each segmentation point (*j*, *s*) in turn, and select the segmentation point with the smallest loss function.

$$LF=\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{2}$$

Among them, c_1 and c_2 are the output average value in the interval R_1, R_2 respectively.

3) Calculate the point of division, proceed in sequence until the division can no longer be continued.

4) Divide the input space into *M* parts $R_1,R_2,\cdots,R_M$ to generate the final decision tree as

$$f(x)=\sum_{m=1}^{M}c_m I(x\in R_m) \tag{3}$$

2.2. Random Forest

RF is composed of multiple decision trees combined into a strong classifier on the basis of bagging. (It shown in Figure 1) It uses Bootstrap to randomly sample *m* instances with a replacement on the training set, and selects random features for each decision tree. Build *m* decision tree models from these *m* samples. Finally, the results are obtained by voting through these *m* models. The specific algorithm steps are as follows:

1) The training set *D* input.

2) Using Bootstrap sampling to form *k* training subsets.

3) Randomly extract *m* features from the original features.

4) Perform training on the training subset, make the optimal segmentation of the randomly selected *m* features, and obtain *k* decision tree prediction results.

5) Voting based on *k* prediction results to get the prediction result with the highest number of votes.

2.3. PSO

The PSO algorithm simulates a bird in a flock of birds by designing a massless particle. This particle has only two attributes: speed and position. Speed represents the speed at which it moves, and position represents its spatial position. Each

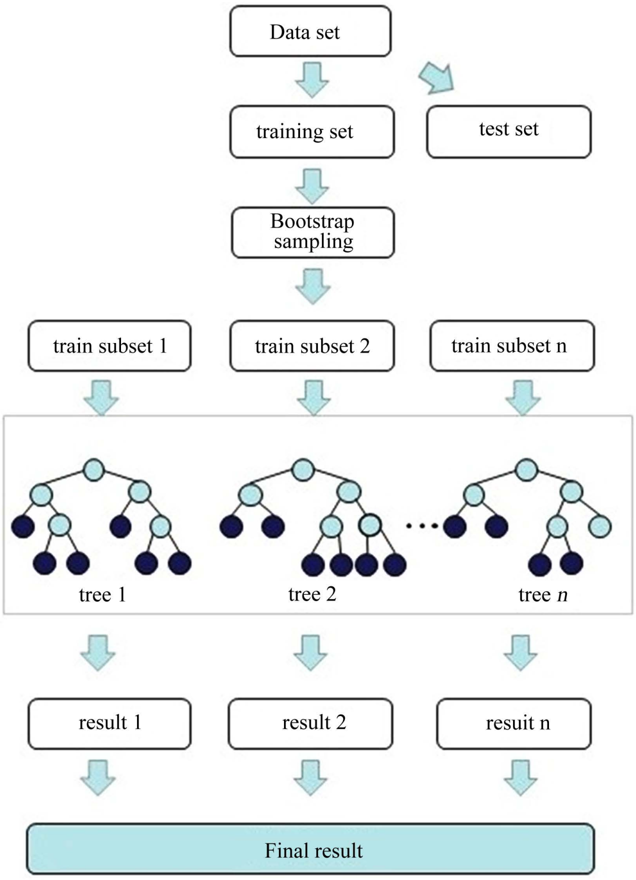

Figure 1. Random forest.

particle finds the optimal solution in the individual search space, stores it as the current individual extreme value, finds the current global optimal solution according to the individual extreme values of all current particles, and adjusts its speed and position for the entire particle swarm. The traditional PSO algorithm is described as follows:

Suppose there is a population of *m* particles in the *d*-dimensional search space. Suppose that at time *T*, population particle information: Position $X_i = \left[x_i^1, x_i^2, \cdots, x_i^d\right]$, speed $V_i = \left[v_i^1, v_i^2, \cdots, v_i^d\right]$, personal best position $p_i = \left[p_i^1, p_i^2, \cdots, p_i^d\right]$, global optimal position $p_g = \left[p_g^1, p_g^2, \cdots, p_g^d\right]$.

Then, the speed and position information of the particles are updated at time *T* + 1 by the following formula:

$$\begin{aligned} v_i^{t+1} &= \omega v_i^t + c_1 r_1^t \left(p_i^t - x_i^t\right) + c_2 r_2^t \left(p_g^t - x_i^t\right), \\ x_i^{t+1} &= x_i^t + v_i^{t+1} \end{aligned} \tag{4}$$

Among them, the inertia weight maintains an effective balance between global exploration and local exploration, and is the learning factor, respectively responsible for adjusting the step length in the exploration direction to the optimal position of the population and the exploration direction to the global optimal position, and is Random numbers on the uniform distribution function. In order to avoid blind search of particles, their speed and position are usually limited to $[-V_{\max}, V_{\max}]$, $[-X_{\max}, X_{\max}]$.

3. APSO-RF Unbalanced Data Classification Model

In this section, we introduce the structure of the model APSO-RF in detail. First, PSO improved by adaptive learning strategies is shown. In the process of searching, group is adaptively divided into subgroups according to the particle distribution. In each subgroup, we use two different learning strategies to guide the search directions of two different types of particles. Then, the optimization model building process is introduced. By applying APSO to optimize the selected hyper-parameters, the classification model was established.

Relevant studies have shown that the diversity of the population is the key to avoiding the premature convergence of PSO; the core guiding principle of the algorithm is clustering. According to the distribution of each particle, the fast search clustering method [14] is adopted to perform the adaptive division of the population into several subgroups. This method can automatically discover the data set samples' class cluster centre. The basic principle is that the centre of the class cluster has two basic features: The first is that it is surrounded by points with lower local density, and the second is that it has a greater distance from points with a higher local density. Therefore, for a population of N particles $S=\{x_i\}_{i=1}^{N}$, the two properties ρ_i and δ_i are defined for each particle. ρ_i, the distance between the local density of the particle and a higher local density of particles, is defined as follows:

$$\rho_i = \sum\nolimits_{j\neq i} \exp\left(-\left(\frac{d_{ij}^2}{d_c}\right)\right) \tag{5}$$

where d_{ij} is the Euclidean distance of particles between x_i, and x_j and d_c is the truncation distance. The truncation distance is $d_c = d_{R*M}$, where R represents the proportion and M indicates that the matrix d_{ij} contains $M=\frac{1}{2}N(N-1)$ values, where N represents the number of particles. It can be seen that d_c is the distance corresponding to the $R*Mth$ value of d_{ij}. (6) gives the expression of the distance δ_i, representing the minimum distance from particle i to other particles that have a higher ρ_i:

$$\delta_i = \min_{j:\rho_j>\rho_i}\left(d_{ij}\right) \tag{6}$$

For the maximum local density ρ of the sample, $\delta_i = \max\limits_j d_{ij}$.

According to Equation (5), if the density of particle x_i is the maximum, δ_i is much larger than the distance δ of its nearest particles. Therefore, the centre of the subgroup consists of particles that have an unusually large distance δ and a relatively high density as well. In other words, the particles with larger ρ and δ values are selected as the centre of the cluster. According to the above idea from [14], the formula $\gamma_i = \rho_i * \delta_i$ is used to filter out particles that may become cluster centers. We arrange the γ_i values in descending order, then use the truncation distance to filter out the cluster centers from the order. Because the γ value of the top particle is more likely to increase exponentially than those of the other particles, it is distinguished from the γ value of the next particle. Referring to [14], R is set to be between 0.1 and 0.2. Through a parameter sensitivity analysis, we found that the value of the distribution parameter has no effect on the performance of the particle swarm algorithm. The default value in this article is 2. The cluster centre is obtained by dividing by the truncation distance after placing the other particles x_j in subgroups where the denser ρ is larger than the ρ of x_j and the δ is the closest to the δ of x_j.

The particles of each subgroup are divided into ordinary particles, and local optimal particles based on the result of the division of subgroups. Under the primary guidance of the optimal particles, the ordinary particles exert their local search ability, and the updated formula is given as (7).

$$x_i^d = \omega x_i^d + c_1 rand_1^d \left(pbest_i^d - x_i^d \right) + c_2 rand_2^d \left(cgbest_c^d - x_i^d \right) \tag{7}$$

where ω is the inertia weight, c_1 and c_2 are the learning factors, $rand_1^d$ and $rand_2^d$ are uniformly distributed random numbers in the interval [0, 1], $pbest_i^d$ is the best position of particles, and $cgbest_c^d$ is the current best position of particle in the subgroup c. To enhance the exchange of information between subgroups, the local optimal particles are mainly updated by integrating the information of each subgroup. The update formula is as follows (see (8)), where C is number of subgroups.

$$x_i^d = \omega x_i^d + c_1 rand_1^d \left(pbest_i^d - x_i^d \right) + c_2 rand_2^d \left(\frac{1}{C} \sum_{c=1} cgbest_c^d - x_i^d \right) \tag{8}$$

Ordinary particles search for local optimality, but more importantly, they are used as the medium for information exchange between subgroups to modify the direction of population search and further improve population diversity. In the same subgroup, unlike a learning strategy that causes too many particles to be gathered locally, the learning strategy integrates the information of the locally optimal particles from different subgroups to obtain more information and help avoid local optima. In addition, learning too much information may lead to the direction of the update being too fuzzy, which may counteract the convergence of particles. Considering that the local optimal particles have the maximum probability of finding the optimal solution in the subgroup, valuable guidance for the optimal solution is provided by their information. Therefore, the $gbest_c^d$ of each subgroup uses the average information to guide the local optimal particle update (see (8)). The transmission of the optimized information in the subgroups can be improved by this approach, the population diversity can be further increased, and particles can be prevented from falling into local optima.

3.1. APSO-RF

In order to make the model structure of RF match the data features more accurately and get the classification prediction results accurately, we use adaptive particle swarm optimization to control the hyper-parameters of the model structure, and build the APSO-RF model (shown in Figure 2). By adaptively dividing the population, the update strategy guides the particle information update to avoid the particles from falling into the local optimum, thereby overcoming the shortcomings of traditional particle swarms.

First, the hyper-parameters in the RF model are taken as the optimization target, and the position information of each particle is randomly initialized in the set hyper-parameter value space.

Second, the particles are divided into adaptive populations. This step is realized by calculating the local density of the particles and the distance to the higher local density particles. According to the value determined by the particle position, the hyper-parameters of the RF model are assigned, and the verification data is brought into the model for prediction, and the loss function value of the model on the verification data set is used as the particle fitness value.

Among them, and respectively represent the true value and the probability prediction value. According to the fitness value of each particle, the subgroup is divided into various types of particles. Use the update strategy to update the information of different types of particles. When the termination condition is reached, the optimal value in the current parameter space is obtained. Finally, the RF model is constructed with the optimal value of the hyper-parameter.

3.2. Data Partition

The theory of cross-validation was started by Seymour Geisser [15]. It is important to guard against testing hypotheses. Especially if the subsequent samples are dangerous, they are too expensive or impossible to collect. Cross-validation: Sometimes called cyclic estimation, this is a statistically useful method of slicing data samples into smaller subsets. Mainly used in modeling applications, such as PCR,

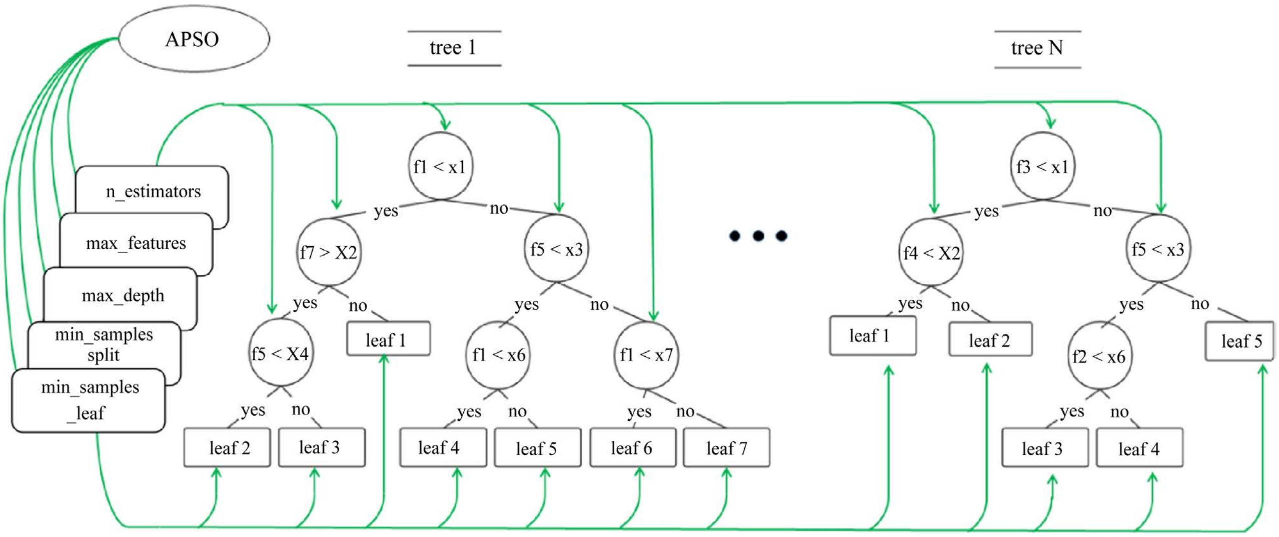

Figure 2. APSO-RF.

PLS regression modeling. In the given modeling sample, take out most of the sample to build the model, leave a small sample with the model just established for prediction, and find the prediction error of this small sample, record their square sum.

Cross-validation can make full use of limited data to find appropriate model parameters to prevent overfitting. The main steps of K-fold cross-validation are as follows: The initial sampling is divided into K sub-samples, a separate sub-sample is used as the data of the validation model, and other $K - 1$ samples are used for training. Cross-validation repeats K times, each subsample verifies once, the average K times result, finally obtains a single estimate. The advantage of the method is that the randomly generated subsamples are repeatedly used for training and verification. In the experiment, we used the most common 10-fold cross-validation.

3.3. Data Pre-Processing

Although the tree-based algorithm is not affected by scaling, feature normalization can greatly improve the accuracy of classifiers. The training set is described as $D = \{X, Y\}$, where $X = \{x_1, x_2, \cdots, x_m\}$ represent an m dimensional eigenspace, $Y = \{0, 1\}$ represents the target value. If x is a certain feature, it by 0 - 1 scaling as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{9}$$

where x'expresses the standardized value.

4. Experiment Setting

4.1. Data Set

The experimental data of this study is an unbalanced data set obtained in the keel data mining platform (see Table 1). All the imbalanced data sets are available with imbalance ratio between 1.5 and 9. The specific details of the data set are shown in the table. IR represents the class imbalacen ratio.

Table 1. The Keel data set.

Name	Attributes	Examples	IR
ecoli-3	7	336	8.6
glass-1	9	214	1.82
new-thyroid-1	5	215	5.14
page-blocks-0	10	5472	8.79
vehicle-1	18	846	2.9
wisconsin	9	683	1.86
yeast-1	8	1484	2.46

4.2. Data Pre-Processing

Data standardization scales data so that it falls into a small specified interval. This removes the unit limitation of the data and turns it into a dimensionless, pure value that can be compared and weighted across different units or orders of magnitude. Figure 3 and Figure 4 show the numerical range distribution after standardization.

4.3. Data Partition

After a large number of experiments proved that 10-fold cross-validation is the most widely used and the best effect, and before verifying the validity of the model, we unified all the cross-validation on different models, all of which were 10-fold cross-validation (See Figure 5).

4.4. The Setting Rangle of Hyper-Parameters

According to the previous RF parameter optimization research, we put a group of hyper-parameters as optimization targets and set their search space. The Settings range is shown in Table 2.

4.5. Measure

To compare the results of the evaluation model, we use the evaluation criteria based on confusion matrix (see Table 3). True Positive (TP) is the number of

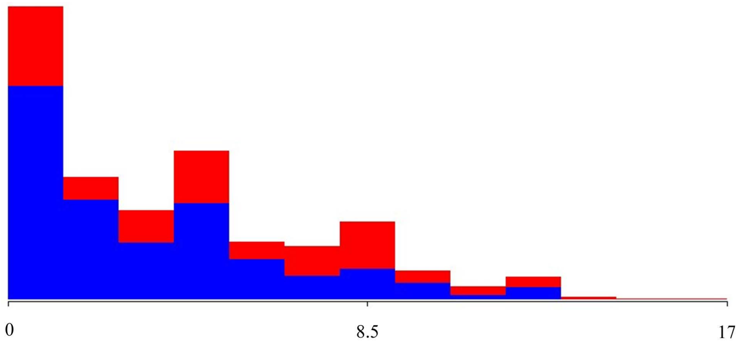

Figure 3. Original data.

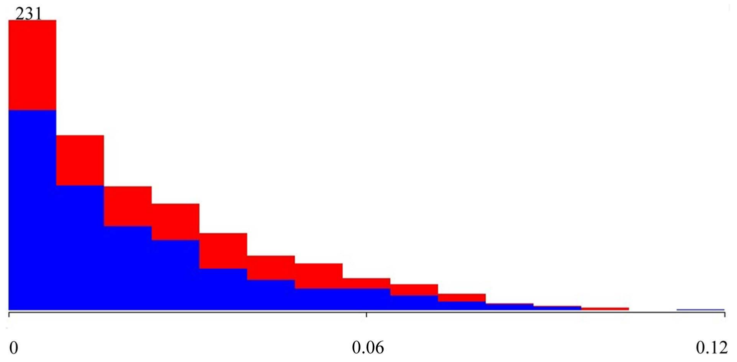

Figure 4. Standardized data.

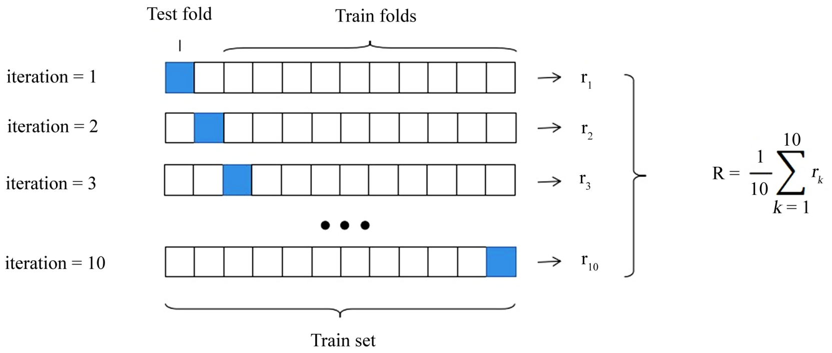

Figure 5. 10-fold cross-validation.

Table 2. The range of hyper-parameters.

Name	Attributes
n estimators	(50 - 200)
max features	(12 - 16)
max depth	350, 400, 450
min samples split	(2, 3)
min samples leaf	(1, 5)

Table 3. Confusion matrix.

Predicted Value	**Actual Value**		**Total**
	0	1	
0	TP	FN	TP + FN
1	FP	TN	FP + TN
	TP + FP	FN + TN	TP + FP + FN + TN

samples that are predicted to be positive class; true negation (TN) is the number of actual negative samples and predicted negative samples; false positive (FP) is the number of actual negative samples and predicted positive samples; false negative (FN) is the number of actual positive samples and predicted negative samples. Both F1-mearsure and Roc Area are comprehensive measures of the ability to deal with unbalanced data sets. The formulas are as follows.

The average accuracy (ACC):

$$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}} \tag{10}$$

The F1-mearsure takes into account both precision and recall of classification models. It is the harmonic average of these two indicators, and it ranges from 0 to 1. ROC is a graph to judge the accuracy of the prediction. If the graph area is close to 1, it is 100% correct.

$$\mathrm{F1} = 2\frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{11}$$

where precision is the proportion of positive samples in positive cases, it is defined as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

And recall is the proportion of predicted positive cases in the total positive cases; it is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

4.6. Baseline Model

In order to analyze and verify the performance of the proposed model for unbalanced data classification research, we selected several commonly used machine learning classification models for comparison.

DT: The DT is a process for classifying instances based on features, where each internal node represents a judgement on an attribute, each branch represents the output of a judgement result, and each leaf node represents a classification result. The algorithm loops all splits and selects the best-partitioned subtree based on the error rate and the cost of misclassification.

Logistic regression (LR): The statistical technique of logistic regression is usually used to solve binary classification problems. Regression analysis is used to describe the relationship between the independent variable x and the dependent variable Y and to predict the dependent variable Y. LR adds a logistic function on the basis of regression.

Multilayer perceptron mechine (MPN): It refers to neural principles, where each neuron can be regarded as a learning unit. The MPN is constructed on the basis of many neurons, which are composed of an input layer, hidden laver, and output layer. These neurons take certain characteristics as input and obtain output according to their own model. The weight assigned to each attribute varies according to its relative importance, and the weight is adjusted iteratively to make the predicted output closer to the actual target.

Support vector machine (SVM): By mapping the feature vector of an instance to a point in space, the purpose of the SVM is to draw a line to best distinguish the two types of points. The SVM finds the hyperplane that separates the data. To best distinguish the data, the sum of the distances from the closest points on both sides of the hyperplane is required to be as large as possible.

Data	Model	ACC	Precision	Recall	F1-Measure	ROC Area
ecoli-3	LR	0.922	0.917	0.922	0.919	0.910
	MPN	0.932	**0.933**	0.934	0.933	0.932
	DT	0.922	0.915	0.922	0.917	0.824
	SVM	0.896	0.802	0.896	0.846	0.502
	RF	0.934	0.929	0.934	0.932	0.936
	APSO-RF	**0.938**	0.931	**0.942**	**0.941**	**0.939**

Continued

glass-1	LR	0.648	0.621	0.648	0.618	0.681
	MPN	0.662	0.668	0.662	0.665	0.676
	DT	0.775	0.770	0.775	0.771	0.749
	SVM	0.793	0.794	0.793	0.784	0.743
	RF	0.836	0.837	0.836	0.824	0.896
	APSO-RF	**0.851**	**0.841**	**0.838**	**0.830**	**0.902**
new-thyroid-1	LR	0.986	0.986	0.986	**0.986**	0.997
	MPN	0.981	0.981	0.981	0.981	0.997
	DT	0.981	0.981	0.981	0.981	0.972
	SVM	0.879	0.894	0.879	0.847	0.629
	RF	0.972	0.972	0.972	0.971	0.998
	APSO-RF	**0.988**	**0.987**	**0.986**	0.982	**0.998**
page-blocks-0	LR	0.951	0.947	0.95	0.947	0.941
	MPN	0.968	0.967	0.968	0.967	0.978
	DT	0.986	0.986	0.986	0.986	0.991
	SVM	0.994	0.992	0.990	0.994	0.977
	RF	0.996	0.995	0.992	0.996	0.993
	APSO-RF	**0.997**	**0.997**	**0.994**	**0.998**	**0.994**
vehicle-1	LR	0.786	0.781	0.786	0.783	0.937
	MPN	0.842	0.838	**0.840**	0.839	0.918
	DT	0.717	0.714	0.717	0.716	0.830
	SVM	0.492	0.242	0.492	0.325	0.502
	RF	0.831	0.803	0.812	0.821	0.933
	APSO-RF	**0.852**	**0.840**	0.832	**0.842**	**0.944**
wisconsin	LR	0.965	0.965	0.965	0.965	0.992
	MPN	0.963	0.963	0.963	0.963	0.992
	DT	0.959	0.959	0.959	0.959	0.957
	SVM	0.960	0.964	0.960	0.961	0.968
	RF	0.969	0.969	0.969	0.969	0.993
	APSO-RF	**0.971**	**0.975**	**0.974**	**0.975**	**0.994**
veast-1	LR	0.757	0.740	0.757	0.728	0.790
	MPN	0.769	0.756	0.769	0.757	0.796
	DT	0.760	0.745	0.760	0.746	0.726
	SVM	0.721	0.713	0.721	0.626	0.526
	RF	0.778	0.767	0.778	0.769	0.806
	APSO-RF	**0.782**	**0.774**	**0.782**	**0.770**	**0.821**

5. Main Result

This paper proposes an unbalanced data classification model based on RF optimized by APSO. The main flow of the model is as follows. First, the data preprocessing involves standardized datasets. And divide the data sets into train data and test data, train data for the training model, the test data for prediction. Second, initialize the adaptive PSO algorithm. Take the logistic loss function as the fitness value, and calculate the fitness value of each particle. The model is constantly searching for the optimal parameters according to the fitness value updatad by the loss function. Until the termination condition is reached, the optimal value found is output. According to hyper-parameters tuned by APSO, the model is built. In the end, the trained model tests the training set and obtains indicators.

First, divide the data sets, train the data for the training model, verify the data for prediction. Initialize the adaptive PSO algorithm. Take the logistic loss function as the fitness value, and calculate the fitness value of each particle. Build the XGBoost model with the corresponding hyper-parameters determined by current best particle. Training and prediction of data sets, and the fitness value are updated by the loss function. Third, determine the position of the global optimal particle and the local optimal particle according to the result of the population division and the fitness values of the particles. Finally, update the positions of the ordinary particles and locally optimal particles, respectively. Judge whether to terminate. When the maximum number of iterations is n, return the optimal value of the hyper-parameter; otherwise, model continues training. Obtain the optimal hyper-parameters to build the XGBoost model and calculate the indexes.

On the data set ecoli-3, the RF model performs better than other types of models, and most of the indicators surpass other models. RF have obtained good results, which shows that the ensemble model can pay more attention to learning unbalanced data sets. Moreover, APSO-RF model reached the highest value on the F1-measure, 93.8%, which is 0.8% higher than NN.

On the data set glass-1, the results of APSO-RF are satisfactory for its all evaluation criteria are better than other algorithms. Compared with the SVM, our model has improved ACC and F1-measure by 7.3% and 5.8%, respectively. The model with hyper-parameters setting optimized by APSO has improved in all indicators, especially in ACC and F1-measure, compared to RF in these two indicators was 1.7% and 0.7% respectively. This shows that the model can still deal with the problem of data imbalance well meanwhile ensuring the level of overall accuracy.

On the new-thyroid-1 data set, the LR model performs better than RF. RF does not optimize the hyper-parameter settings, which makes it insufficient to learn samples. RF performs better than LR on F1-measure, indicating that RF uses the bagging method it has good generalization ability. The key to adopting this method is to deal with imbalances to obtain effective classifiers while ensuring the diversity of base classifiers; the model APSO-RF optimized by hyper-parameters has reached acc and other indicators to the top, it shows that the improved particle swarm can help the model build a branch structure suitable for the data set, by selecting reasonable hyper-parameter settings.

Most models on page-block-0 performed well, and on the evaluation indicators, APSO-RF algorithm was better than other algorithms. The model's performance in F1-measure ranks in the forefront, indicating that our model is superior to other algorithms in the classification performance of unbalanced data.

On the wisconin data set, the APSO-RF is better 0.2% than RF at ACC; APSO-RF is 0.6% higher in ACC than the third-highest model LR model, and the model has the best recall rate, indicating that the model can distinguish more positive categories.

On the veast-1, our model has achieved the best performance in all indicators, and it also performs well in prediction accuracy and regression rate. A high accuracy rate means that the positive examples in the sample are more accurately predicted. It shows that our proposed algorithm is superior to other algorithms in the classification performance of positive classes.

On the whole, RF has better average performance than other models, which shows that this model can reduce the model error effectively and achieve more accurate unbiased estimation with the help of integrated classification strategy. Specifically, traditional classification algorithms usually use classification accuracy as the evaluation criteria, and aim to maximize the average accuracy. In order to maximize accuracy, they often sacrifice the performance of the minority class; while the RF uses an appropriate induction algorithm to benefit the minority Class classification learning. APSO-RF is improved obviously compared with RF in all indexes, which shows that hyper-parameters can match the fitness value better, and its tree structure more suitable for non-balanced data, so the precision of the model is higher. The algorithm can improve the ability of positive classification obviously without losing the ability of global classification, because APSO is optimizing the hyper-parameter reasonably, as a result, the tree structure that is more suitable for unbalanced data set is not built, and the performance is limited. Adaptive particle swarm optimization uses adaptive group division and different updating strategies to guide particles learning, which helps to maintain the diversity of the population and avoid the model falling into local optimum early.

6. Conclusions and Suggestions

Unbalanced data classification is a big challenge in the field of data mining. RF as an ensemble learning method is usually used to solve the problem of unbalanced data classification. This paper proposes a particle swarm optimization strategy based on adaptive partitioning, which uses the good global and local search performance of the optimization strategy to optimize the hyper-parameters of the RF, and optimizes the misclassification of samples in the imbalanced data classification problem. The purposed model is verified on six non-equilibrium data sets and gets good prediction results. The result demonstrates that the model has excellent generalization ability and the ability to deal with non-equilibrium data sets.

Our Future work will focus on the improvement of the integrated decision tree structure to further improve the performance of the model itself.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61972227, Grant 61873117 and Grant U1609218, in part by the Natural Science Foundation of Shandong Province under Grant ZR201808160102 and Grant ZR2019MF051, in part by the Primary Research and Development Plan of Shandong Province under Grant GG201710090122, Grant 2017GGX10109, and Grant 2018GGX101013, and in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] He, H.L. and Fan, Y.L. (2021) A Novel Hybrid Ensemble Model Based on Tree-Based Method and Deep Learning Method for Default Prediction. *Expert Systems with Applications*, **176**, Article No. 114899. https://doi.org/10.1016/j.eswa.2021.114899

[2] Zhang, H., Li, J.L., Liu, X.M. and Dong, C. (2021) Multidimensional Feature Fusion and Stacking Ensemble Mechanism for Network Intrusion Detection. *Future Generation Computer Systems*, **122**, 130-143. https://doi.org/10.1016/j.future.2021.03.024

[3] Abkenar, S.B., Mahdipour, E., Jameii, S.M. and Kashani, M.H. (2021) A Hybrid Classification Method for Twitter Spam Detection Based on Differential Evolution and Random Forest. *Concurrency and Computation*: *Practice & Experience*.

[4] Li, H.X., Feng, A., Lin, B., Su, H.C., Liu, Z.X., Duan, X.L., Pu, H.B. and Wang, Y.F. (2021) A Novel Method for Credit Scoring Based on Feature Transformation and Ensemble Model. *PeerJ Computer Science*, **7**, e579.

[5] Sun, Y.M., Wong, A.K.C. and Kamel, M.S. (2009) Classification of Imbalanced Data: A Review. *International Journal of Pattern Recognition & Artificial Intelligence*, **23**, 687-719. https://doi.org/10.1142/S0218001409007326

[6] Janicka, M., Lango, M. and Stefanowski, J. (2019) Using Information on Class Interrelations to Improve Classification of Multiclass Imbalanced Data: A New Resampling Algorithm. *International Journal of Applied Mathematics and Computer Science*, **29**, 769-781. https://doi.org/10.2478/amcs-2019-0057

[7] Pei, W., Xue, B., Shang, L. and Zhang, M. (2019) Genetic Programming for Development of Cost-Sensitive Classifiers for Binary High-Dimensional Unbalanced Classification. *Applied Soft Computing*, **101**, Article No. 106989. https://doi.org/10.1016/j.asoc.2020.106989

[8] Wolpert, D.H. and Macready, W.G. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, **1**, 67-82. https://doi.org/10.1109/4235.585893

[9] Dietterich, T.G. (2000) Ensemble Methods in Machine Learning. In: *MCS* 2000: *Multiple Classifier Systems*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

[10] Kearns, M. and Valiant, L.G. (1989) Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. 21st *Annual ACM Symposium on the Theory of Computing*, Seattle, Washington, USA, May 14-17 1989, 433-444. https://doi.org/10.1145/73007.73049

[11] Breiman, L. (2001) Random Forest. *Machine Learning*, **45**, 5-32. https://doi.org/10.1023/A:1010933404324

[12] Alhudhaif, A. (2021) A Novel Multi-Class Imbalanced EEG Signals Classification Based on the Adaptive Synthetic Sampling (ADASYN) Approach. *PeerJ Computer Science*, **7**, e523. https://doi.org/10.7717/peerj-cs.523

[13] Qiang, Y., Chen, W.N., Deng, J.D., Yun, L. and Zhang, J., (2018) A Level-Based Learning Swarm Optimizer for Large-Scale Optimization. *IEEE Transactions on Evolutionary Computation*, **22**, 578-594. https://doi.org/10.1109/TEVC.2017.2743016

[14] Wang, H.U. (2007) A Simpler and More Effective Particle Swarm Optimization Algorithm. *Journal of Software*, **18**, 861-868.

[15] Lee, J.C., Johnson, W.O. and Zellner, A. (1994) Modelling and Prediction: Honoring Seymour Geisser. https://link.springer.com/book/10.1007%2F978-1-4612-2414-3