A New Approach for Imbalanced Data Classification Based on Minimize Loss Learning

Chunkai Zhang, Guoquan Wang, Ying Zhou, Jiayao Jiang
School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, People's Republic of China
ckzhang812@gmail.com, gqwhitsz@gmail.com, {442049887, 616905919}@qq.com

***Abstract*—The class imbalance problem occurs when instances in one class are more than that in another. It has been reported to severely hinder classification performance of many traditional classification algorithms and many researchers have paid a great deal of attention to this field. Different kinds of methods have been pro-posed to solve the problem these years, such as resampling methods, integrated learning method. However, these conventional class imbalance handling methods might suffer from the loss of potentially useful information, unexpected mistakes or increasing the likelihood of overfitting because they may alter the original data distribution. In this study, we propose a new method for imbalanced data sets which is different from previously proposed solutions to the class imbalance problem. We put forward the idea that treat the performance measures as training target, then designed the loss function and build a model based on artificial neural network to solve the problem. The experimental results on 8 imbalanced data sets show that our proposed method is usually superior to the conventional imbalanced data handling methods.**

Keywords- Imbalanced data classification, F1-measure, Artificial neural network, Minimize loss learning

I. INTRODUCTION

In recent years, imbalanced classification problems have attracted a lot of re-searcher's interests due to its challenges in various real-world applications. Different from standard classification problems, an imbalanced task involves a data set that has an imbalanced class distribution, i.e., the number of instances in one class(majority class) is outnumbered by the number of instances in another class(minority class)[1]. There exist imbalanced class distributions in many real world domains, such as detecting oil spills from satellite images [2], identifying fraudulent credit card transactions [3], medical diagnosis [4], anomaly detection [5], finance risk management [6], software defect prediction [7], and bioinformatics [8]. Thus, to maximize the recognition rate of the minority class on the premise of considering a good tradeoff for both the minority and majority classes is the main goal of an imbalanced classification task.

In abovementioned imbalanced classification problems, we always care more about the minority class because they contain more vital information. However, a problem usually occurs because traditional classification algorithms tend to be biased towards to the majority class [9, 10]. These algorithms such as decision trees, k-nearest neighbors and artificial neural networks often tend to generate models which maximize the overall classification accuracy but ignore the minority class. For example, for a medical detection data set, the number of patients who suffers from cancer is 2% of all instances. Even if a disease classifier predicts that all the instances are the majority class, it will perform well and achieves an overall accuracy of 98%. However, the real cancer patients, which we want to accurately classify, are all misclassified by the model. For the reason, a lot of methods have been developed to solve the imbalanced classification problems.

Classical imbalanced classification methods can be divided into three categories: (i) data level, (ii) algorithmic level and (iii) ensemble classifiers. The data level methods are applied as a pre-processing technique of the existing classifiers. Common approaches include random over-sampling of the minority class [11], random under-sampling of the majority class [12, 13], or synthetic over-sampling (SMOTE) to create new minority instances by interpolating between similar known instances[14]. Algorithmic level approaches try to adapt specific classification algorithm to reinforce the learning towards the minority class. For example, a Support Vector Machine (SVM) with a kernel function biased to the minority class is proposed in [15] to improve the minority class prediction. Cost sensitive learning [16, 17] solve the class imbalance problem by using different cost matrices that describe the costs for misclassifying any particular data example, usually high cost for the minority class and low cost for the majority class. Various empirical studies have shown cost sensitive learning is superior to sampling methods in some application domains. Moreover, ensembles of classifiers are another popular approaches in the last decade [18]. It mainly contains two approaches: bagging [19, 20] and boosting [21, 22, 23]. Bagging contains different classifiers which are applied to subsets of the data. Alternatively, in boosting, the whole data set is used to train classifiers in each iteration while more attention is given to the classification of the samples that are misclassified in the previous iteration.

Even though the above mentioned methods have achieved great success in solving some class imbalance problems. They might encounter some unexpected problems in some cases. For example, sampling methods alter the original class distribution of imbalanced data such that traditional classifiers can be used to solve the problem. However, the main idea of traditional machine learning algorithm is based on the summary of the independent and identically distribution hypothesis and the induction bias. If the distribution of the original data set is changed, the traditional machine learning method may cannot

*This work is supported in part by National High Technology Research and Development Program of China (No. 2015AA016008)

978-1-5386-1600-0/17 $31.00 © 2017 IEEE
DOI 10.1109/DSC.2017.58

reasonably fit the original data set stably. Similarly, Bagging and Boosting methods based on ensemble classifiers may suffer from overfitting as well because they also use sampling methods to obtain balanced data in each of their iteration procedure. Furthermore, as for cost-sensitive approaches, it is hard to set the cost matrices properly and may depend on the characteristics of the data sets. The standard public classification data sets do not contain the costs [23] and over-training is highly possible when searching to find the most appropriate costs, so the classification results are not always stable.

In this study, we present a novel approach to tackle the imbalanced data classification problems. Our proposed method handles the class imbalance problems directly using the widely used imbalance data classification evaluation criteria F1-measure value as the training target based on minimize losing learning. We put forward the idea that treat the performance measures as training target, then designed the loss function and build a model based on artificial neural network to solve the problem. To validate the effectiveness of the proposed method, we applied the proposed method to eight benchmark imbalanced data sets. We also compared the proposed method with some existing classification methods. The experimental results on 8 data sets show that our proposed method is usually superior to the conventional imbalanced data handling methods.

The rest of this paper is organized as follows: Section II provides basic information about relevant background. Section III explains the proposed method. Section IV follows with the results of these experiments. Finally, concluding remarks are contained in Section V.

II. Related Work

A. Minimize Loss Learning

For most of the existing machine learning algorithms, we usually think that the training data set $S = ((x_1, y_1), \dots, (x_n, y_n))$ and the test data set satisfy the independent and identically distribution, this distribution is also considered the true spatial distribution of all samples, that is, we summed up the training data and test data summarized Bias to classify unknown samples in real space H. The training goal of the machine learning algorithm is to find a special hypothesis $h \in H$ in all hypothesis spaces to make the test data set reach the set minimum value of loss, namely the formula (1) reaches the minimum value. For the traditional loss function, as it is usually considered that the samples in the space are independent from each other, the loss function can be transformed into the form of (2), that is, the overall loss is transformed into the sum of the individual losses of all samples.

So according to Equations (1) and (2), and based on the hypothesis of independent and identically distribution between training data and real data, we can finally convert (1) to the form of (3), which is why we can fit the real space through the training data set.

$$R^{\Delta}(h) = \int \Delta((h(x_1'), \dots, h(x_n'), (y_1', \dots, y_n')))dPr(S') \quad (1)$$

$$\Delta\left(\left(h(x_1'), \dots, h(x_n'), (y_1', \dots, y_n')\right)\right) = \sum_{i=1}^{n'} \delta(h(x_i'), y_i') \quad (2)$$

$$R^{\Delta}(h) = R^{\delta}(h) = \int \delta(h(x', y')dPr(x', y') \quad (3)$$

$$J(X, Y; W) = \int_{\Omega} loss(x', y')d(x', y') = \sum_{(x', y') \in \text{training set}} loss(x', y') \quad (4)$$

For most of the traditional machine learning methods, their training idea is usually to construct the error function between the output of a single sample and the target output, and then add the errors of all the samples as the total loss of the training set. The ideal loss function is the 0-1 loss function. If the output and the target output belong to the same category, the loss is 0, and 1 otherwise, as shown in Equation (5).

$$loss_{0-1} = \begin{cases} 0, & h(x) \times y > 0 \\ 1, & h(x) \times y < 0 \end{cases} \quad (5)$$

However the average of 0-1 loss cannot apply to all the problems, so the idea of minimize loss learning is proposed. By using a custom loss function instead of 0-1 loss, in order to adapt to different problems, this idea was first applied to structured support vector machines.

The specific idea is to convert the form of a single input to a single output, such as Equation (6), into the form of Equation (7). In the real training process, we use the training set's feature set $\bar{x} = \{x_1, \dots, x_n\}$and target output set $\bar{y} = \{y_1, \dots, y_n\}$ to solve the problem. Therefore, the training space in Equation (7) is $\bar{X} = X \times \dots \times X$, the target space is $\bar{Y} = \{0,1\}^n$. Assume that h is the global assumption, corresponding to all the sample inputs and all the classifier outputs. As the overall loss is transformed from the form of Equation (4) as the loss of each sample to the form of Equation (8). Solving assumption (7) is equivalent to solving the results of Equation (9). By such a way, the traditional machine learning classification algorithm is transformed into the method of minimizing the global loss, so as to establish the loss function through the imbalanced dataset classification evaluation criteria, and to adapt to the imbalanced data classification problem.

$$h: X \to Y \quad (6)$$

$$h: \bar{X} \to \bar{Y} \quad (7)$$

$$E(h) = Loss(h(\bar{x}), \bar{y}) \quad (8)$$

$$h = \underset{h \in H}{\operatorname{argmin}}\big(E(h)\big) = argmin(Loss(h(\bar{x}), \bar{y})) \quad (9)$$

B. Classification Evaluation Criteria

In imbalanced classification problems, a specific metric is needed to evaluate the performance of the classifier, just like Table I.

The performance of classifier is usually measured by accuracy (accuracy $= \frac{TP+TN}{TP+FP+TN+FN}$) in conventional methods. But in the imbalanced classification problem, the misclassification of the minority class costs more than the majority one, so the researchers used a variety of criterions to measure the classification results.

Table I. Confusion matrix for binary classification

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

The receiver operating characteristic (ROC) graphic is commonly used as an evaluation criterion. The ROC graphic depicts the trade-off between TPrate ($\text{TPrate} = \frac{TP}{TP+FN}$, x-axis) and FPrate ($\text{FPrate} = \frac{FP}{FP+TN}$, y-axis), the area under the ROC curve (AUC) is a useful metric for classifying performance because it gives the probability that a randomly selected pair of samples (one positive and one negative) would have their predicted probabilities correctly ordered[24] .

The F1-value is a trade-off between precision (P) and recall (R) which is described as follows:

$$\mathrm{P} = \frac{TP}{TP+FP} \quad , \quad \mathrm{R} = \frac{TP}{TP+FN}$$

$$\mathrm{F} = \frac{PR}{P+R}$$

The number of true positive samples has the greatest influence on the F1-value, so it is used in imbalanced classification problems in which the positive samples are more important. So in this paper, we use F1-value to evaluate the performance of the classifier.

III. PROPOSED METHOD

A. Imbalance Learning and Minimize Loss Learning

Assume that there is a one-dimensional imbalanced data set, which contains both majority samples and minority samples. The probability density curve is shown in Figure 1, and it is assumed that the ratio of the majority class to the minority class is n:1, where n>1. It is obvious that the final training target of traditional classifier is to maximize the global accuracy rate. For the demarcation of the two parts, even if their probability density are similar, the number of majority class samples will be far larger than the number of minority samples because the base number of majority and minority classes are quite different. So the final decision boundary line is likely near the location of line b in figure 1, in favor of the side of minority class.

The idea of the classical imbalanced data classification algorithm is to reduce the sample ratio between majority and minority classes directly by using sampling or other methods. Usually, the number of samples of the two classes will be changed to be very close, and then applied to the traditional classification algorithms. If the probability density curve of the original dataset is still same as that shown in Figure 1, since there is no such problem that the two sample base numbers are different, the classification boundary line which seeks the highest global accuracy should be the line a, which is based on the abscissa of the intersection of two probability density curves as the threshold of demarcation. The minority samples on the left side and majority class samples on the right side of the demarcation line are misclassified samples, which is easy to prove by the area method.

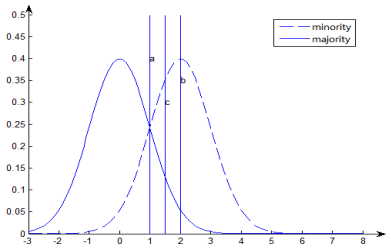

Figure 1. Probability density curve of the data set

However, due to the change of the sample space distribution, we can only think that the solution (line a) is the best classification line in the current changed sample space but cannot determine whether the demarcation line is the best line on the original data set. In general, there are special evaluation criteria to evaluate the effectiveness of the classification algorithms. The classical imbalanced data classification method usually can increases the imbalanced classification result on the original sample space a little only, but they cannot prove that the solution must be the optimal solution in the original sample space.

Therefore, this proposed algorithm is based on minimize loss learning, which uses the widely used imbalanced data classification evaluation criteria F1-measure value to construct a special loss function and then uses the neural network model to solve the problem of imbalanced data classification.

B. The Construction of Loss Function

In this algorithm, we choose F1 as the optimization target, so we can set the loss function as (1-F1). For the training of neural network, the minimum value of loss is same as the maximum of F1, as long as the sign before the gradient can be changed. So we extend the concept of minimizing the loss to maximizing the objective function here, namely the form of (11).

$$\mathrm{h} = \operatorname{argmax}\big(\mathrm{E}(\mathrm{h})\big) = \operatorname*{argmax}_{h\in H}(F(h(\overline{x}),\overline{y})) \qquad (11)$$

For the neural network model in this paper, we use the sgn function of the traditional neural network shown in Equation (12) as the final classification criterion. For the output on the whole training set $\bar{x}$, we use $\bar{z} \in \{0,1\}^n$ to represent, and the target output $\bar{y} \in \{0,1\}^n$ is still used. In order to express the final value of F1 with $\bar{z}$ and $\bar{y}$, we need to find the recall rate (Recall) and accuracy (precision) first. According to the observation of the confusion matrix and the relationship between the confusion matrix, we can find that the parameter TP meet the Equation (13), and the parameters Recall and Precision can be converted into the form of the formula (14) and (15). So the final F1 value can be expressed as the form (16).

$$\mathrm{y} = \mathrm{sgn}(\mathrm{x}) = \begin{cases} 0\,, h(x) < 0.5 \\ 1\,, h(x) \geq 0.5 \end{cases} \qquad (12)$$

$$\mathrm{TP} = \overline{\mathrm{z}} \cdot \overline{\mathrm{y}} \qquad (13)$$

$$\mathrm{Recall}(\overline{\mathrm{z}},\overline{\mathrm{y}}) = \frac{TP}{TP+FN} = \frac{\overline{z}\cdot\overline{y}}{\overline{y}^2} \qquad (14)$$

$$\mathrm{Precision}(\overline{\mathrm{z}},\overline{\mathrm{y}}) = \frac{TP}{TP+FP} = \frac{\overline{z}\cdot\overline{y}}{\overline{z}^2} \qquad (15)$$

$$F1(\bar{z},\bar{y}) = \frac{1}{\frac{1}{recall(\bar{z},\bar{y})}+\frac{1}{precision(\bar{z},\bar{y})}} = \frac{2\cdot\bar{z}\cdot\bar{y}}{\bar{y}^2+\bar{z}^2} \quad (16)$$

However, by observing Equation (16), we can find that $\bar{z}$ and $\bar{y}$ are all sequences comprised by 0 and 1. Although the training target of the entire training set can be expressed by $\bar{z}$ and $\bar{y}$, the F1 value is still discrete because the process of solving $\bar{z}$ involves a step $\mathrm{sgn}(h(x))$ operation and cannot establish a direct numerical connection with our neural network output and the hidden layers between the various nodes. In structured SVM, we use spatial traversal and double optimization to solve arbitrary objective function, and therefore a huge amount of time is needed. In this paper, we give up this idea, but use neural Network output layer sigmoid function to establish the relationship between $F1(\bar{z},\bar{y})$ and $h(x)$.

The neural network training process in this algorithm is to take use of the current state of the network to classify first, then solve the loss and optimize the loss to the next better state. We transform the evaluation process in training, no longer using the current final classification results of neural network, but using the current output probability as formula (17) to solve the expectation value and to optimize the expectation. So that not only the direct contact between output and parameters was established, but also it is possible to increase the probability that the target will acquire a higher value by optimizing the expectation value.

$$P(y = 1|x;\theta_m, w_{nm}) \approx h(x) \quad (17)$$

However, we can't use the exact solution process to establish the contact of expectation. Because the exact solution process takes $O(n^3)$ time and a large amount of storage space, which is not acceptable. To solve this problem, we use the approximate relation in Equation (18) below.

$$E(F(h(\bar{x}),\bar{y})) \approx \frac{2\times h(\bar{x})\times\bar{y}}{h(\bar{x})^2+\bar{y}^2} \quad (18)$$

Because of the relationship between expectation and covariance as formula (19), we can infer the relation (20). When the algorithm optimizes the approximation of the right side, the expectation can be expanded and converged to a global optimum or local optimal solution because the expected value is the upper bound of the approximation and is relatively close, it has reached training purposes.

$$E(XY) = E(X)E(Y) + E((X - E(X))(Y - E(Y))) \quad (19)$$

$$E(F(h(\bar{x}),\bar{y})) \geq \frac{2\times h(\bar{x})\times\bar{y}}{h(\bar{x})^2+\bar{y}^2} \quad (20)$$

To sum up, we turn the original neural network training target (6) into the form of equation (7), and design the objective function according to the commonly used F1 value in the evaluation criteria of imbalanced data classification. Then, the discrete function is transformed to a continuous function approximately so that the function can be perfectly associated with each training set sample output and with the neural network parameters. We also proved that the training process of neural network can optimize the approximation function while optimizing the final F1 value on the training set..

C. The Training Process

For the neural network model, the most effective weight updating strategy is the back propagation algorithm. Since the algorithm will train the objective function to the maximum value, the updating process is as formula (21) and (22), where η represents the learning rate. Its magnitude will impact the neural network's convergence rate and convergence accuracy.

$$\vec{\omega}' = \vec{\omega} + \Delta\vec{\omega} \quad (21)$$

$$\Delta\vec{\omega} = \eta\nabla F1_{\omega} \quad (22)$$

Therefore, in order to find out the update amount of each weight, we need to solve the partial differential of the whole objective function F1 for each parameter in each node. Where ω_{ij} is the i_{th} weight of node j, net_j is the result of the inner product of node j, which is the output of sigmoid function, and x_{ji} is the input of node i corresponding to the i_{th}weight. What follows is to solve the final output's partial derivative for each node output in Equation (22).

$$\frac{\partial F1}{\partial\omega_{ji}} = \frac{\partial F1}{\partial net_j} * \frac{\partial net_j}{\partial\omega_{ji}} = \frac{\partial F1}{\partial net_j} * x_{ji} \quad (23)$$

In order to solve the partial differential of each node, we need to classify all the nodes into two classes, one is the output node and the other is the hidden layer node. Since the final objective function is directly related to the output, the output node can solve the partial differential directly through the objective function value in the current state. For the hidden layer node, it needs to solve the partial differential through the downstream node of the node, which is the main idea of the chain propagation algorithm.

For the output node, we use the chain rule to decompose the request problem into the form as Equation (24).

$$\frac{\partial F1}{\partial net_j} = \frac{\partial F1}{\partial o_j} * \frac{\partial o_j}{\partial net_j} \quad (24)$$

where $\frac{\partial o_j}{\partial net_j} = sigmod'(net_j) = o_j(1 - o_j)$,

and $\frac{\partial F1}{\partial o_j} = \frac{\partial}{\partial o_j} \times \frac{h(\bar{x})\times\bar{y}}{h(x)^2+\bar{y}^2} = \frac{y_j}{h(x)^2+\bar{y}^2} - 2(h(\bar{x}) \times \bar{y}) \times (h(x)^2 + \bar{y}^2)^{-2} \times o_j$

For hidden nodes, there is no way to use the objective function directly, so we should update the connection parameters according to the downstream node of each hidden node. The update scheme is shown in formula 25. Because this algorithm is a binary classification problem, there is only one output node. When the method of solving the output node and the derivative of the sigmoid function are plugged into Equation 25, the form of Equation 26 can be obtained, where δ_k is the partial differential of output node $\partial o_j / \partial net_j$ and w_{kj} is the j_{th} weight of hidden layer node k.

$$\frac{\partial F1}{\partial net_j} = \sum_{k\in DownStream} \frac{\partial F1}{\partial net_j} \times \frac{\partial net_k}{\partial net_j} \quad (25)$$

$$\frac{\partial F1}{\partial net_j} = \delta_k \cdot \frac{\partial net_k}{\partial net_j} = \delta_k \cdot \frac{\partial net_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} = \delta_k w_{kj} o_j(1 - o_j) \quad (26)$$

And detailed process of Minimization Loss Learning Algorithm is shown as the follow algorithm I.

Algorithm I. Minimization Loss ANN

Input: Learning rate η, Maximum number of iterations m Sample set $\bar{x}=(x_1,...,x_n), \bar{y}=(y_1,...,y_n)$, Number of hidden nodes l, Target F1 value f

Output: Input - hidden layer connection coefficient matrix ω_{kl}, Hidden - Output layer connection coefficient vector θ_l

(1) Initialize ω_{kl} and θ_l, the range of each component is (-0.1, 0.1)

(2) $\omega_{kl}' \leftarrow 0, \theta_l' \leftarrow 0, f' \leftarrow 0$

(3) For i = 1 To m

(4) Solve $h(\bar{x})$ for each sample set $\bar{x}$, and solve the current F1 value f_{now}

(5) If($f_{now} > f$)

(6) Return current ω_{kl}, θ_l

(7) Else if ($f_{now} > f'$)

(8) $\omega_{kl}' \leftarrow \omega_{kl}, \theta_l' \leftarrow \theta_l$

(9) End if

(10) Update θ_l according to Equations (23) and (24), update ω_{kl} according to Equations (25) and (26)

(11) End For

(12) Return ω_{kl}', θ_l'

IV. Experiment and Analysis

The experimental datasets in this chapter are all from the UCI machine learning datasets. In the data set selection process, we select the data sets that are used in other imbalance classification algorithms. The experimental parameters of the 8 datasets are as follow Table II.

In this section, we compare the proposed algorithm ML-ANN with some classical imbalanced data classification algorithms: SMOTE algorithm, Adaboost.M1 algorithm (AD), structured support vector machine algorithm (SSVM), classical neural network algorithm (ANN), cost-sensitive learning algorithm (SCL). F1 measure is used to evaluate the algorithms, and the results are shown in the following table III.

Table II. Data set parameter table

No.	Dataset	Number of sample	Fractional proportion	Feature
1	YEAST	1484	12.60%	8
2	Abalone	4177	8.02%	8
3	Glass	214	23.83%	10
4	Breast Canser	699	34.50%	9
5	Vehicle Silhouettes	946	23.43%	18
6	Haberman	305	26.47%	3
7	Ecoli	335	2.69%	7
8	Credit	30000	22.12%	24

Table III. F1 measure Results for the comparison between proposed method and the other methods

No.	SMOTE	Adaboost	SSVM	ANN	SCL	ML-ANN
1	0.747	0.717	0.645	0.667	0.717	0.846
2	0.406	0.399	0.486	0.352	0.331	0.581
3	0.980	0.922	0.980	0.837	0.866	1.000
4	0.984	0.943	0.961	0.984	0.956	0.994
5	0.995	0.997	0.962	0.945	0.979	0.995
6	0.459	0.423	0.632	0.413	0.660	0.647
7	0.762	0.696	0.727	0.941	0.941	0.941
8	0.534	0.435	--	0.503	0.516	0.542

From table III we can see that the proposed algorithm in this paper can achieve a similar performance with the classic imbalanced classification method, the overall classification results on 8 data sets is slightly better than the classic imbalanced methods over F1 value.

As mentioned before, the classical imbalanced data classification methods address the problem by changing the distribution of original samples, so it has a high upper limit of this kind of imbalanced classification method because there may happen to make the ideal sampling results or weight distribution, so the proposed algorithm is not better than classical imbalanced classification algorithm completely.

It has been proved that the algorithm has the ability to fit the training space, and it also proves that the algorithm in this paper can be used as a machine learning algorithm to solve the imbalanced problems. However, just fitting the training set may not be perfect for engineering problems, machine learning methods need to have good generalization ability and cannot be overfitting, so we will compare the proposed algorithm with the other algorithms on the generalization ability.

We usually use the method of cross validation to measure the generalization ability of an algorithm, the dataset is divided into K parts, and the K tests are carried out, each of the K parts is taken as a test set, and other collections are used as the training set to verify the algorithm. Because we want to verify the generalization ability of the classification algorithm on imbalanced data sets, and the minority class samples may be very little, so K should not be taken particularly large. In this experiment we choose K to be 3. In this section we selected four kinds of algorithms whose generalization ability is good: the traditional neural network, AdaBoost algorithm, structured support vector machine and the proposed method in this paper, the cross validation results are in the following table IV.

From the table we can see that both the proposed algorithm and the existing algorithm, the cross validation results are similar to the prior results on the 8 datasets in this paper, so the proposed algorithm has reliable generalization ability, the algorithm's ability of training space fitting is true and reliable.

Form the results we can conclude from the table that the proposed algorithm has achieved some success in imbalanced classification, and its result is usually better than the previous algorithms.

Table IV. The F1 measure of cross validation test results

No.	Adaboost	ANN	SSVM	ML-ANN
1	0.749	0.724	0.699	0.782
2	0.276	0.377	--	0.545
3	0.879	0.865	0.865	0.879
4	0.932	0.945	0.935	0.946
5	0.805	0.927	0.916	0.927
6	0.248	0.328	0.476	0.575
7	0.667	0.875	0.727	0.941
8	0.505	0.434	--	0.712

V. CONCLUSION

In this paper, we start from the unbalanced dataset classification and evaluation criteria, and use F1 value as evaluation criteria instead of the global accuracy rate in the traditional classification method. We try to address the fundamental problem in which the classifier has poor performance. The proposed algorithm start from the loss function of the classifier. Instead of using the traditional loss function, we construct the loss function associated with the F1 value directly, and take the approximation expectation of the current classifier output to compute the F1 value. We proved the assumption that the approximated expectation is the lower bound of the expectation value of F1 to confirm the feasibility of optimization for F1, and the expectation of F1 value is associated with the output of classifier and is no longer discrete. So the training process can be completed by back propagation algorithm.

For the algorithm in this paper, the following problems can be studied or optimized:

(1) In the strict sense, the algorithm does not establish a direct mathematic relation between the expectation of F1 value and the output of neural network model or with the model parameters. If we can overcome this difficulty, the classifier will have a better performance.

(2) In this paper, the algorithm cannot be parallel and cannot use the traditional artificial neural network learning methods such as the batch to speed up, so the proposed is much slower than classic neural network. So if we can solve this problem, there will be a qualitative leap of performance of the algorithm.

REFERENCES

[1] Lizhi Peng, Hongli Zhang, Bo Yang, Yuehui Chen, "A new approach for imbalanced data classification based on data gravitation," Information Sciences 288 (2014) 347–373

[2] M. Kubat, R.C. Holte, S. Matwin, "Machine learning for the detection of oil spills in satellite radar images", Mach. Learn. 30 (1998) 195–215.

[3] T. Fawcett, F. Provost, "Adaptive fraud detection", Data Min. Knowl. Discov. 1(1997) 291–316

[4] M.A. Mazurowski, P.A. Habas, J.M. Zurada, et al, 'Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance", Neural Networks: Off. J. Int. Neural Network Soc. 21 (2008) 427.

[5] W. Khreich, E. Granger, A. Miri, R. Sabourin, "Adaptive roc-based ensembles of hmms applied to anomaly detection", Pattern Recognit. 45 (2012) 208–230.

[6] I. Brown, C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets", Expert Syst. Appl. 39 (2012) 3446–3453.

[7] L. Pelayo, S. Dick, "Applying novel resampling strategies to software defect prediction", in: Annual Meeting of the North American on Fuzzy Information Processing Society, pp. 69–72.

[8] H. Yu, J. Ni, J. Zhao, ACOSampling: an ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, Neurocomputing 101 (2013) 309–318.

[9] S.García,F.Herrera,Evolutionary under sampling for classification with imbalanced datasets: proposals and taxonomy,Evol. Comput. 17 (2009) 275–306.

[10] Y.Tang, Y.Q. Zhang, N.V. Chawla, S. Krasser, SVMs modeling for highly imbalanced classification, IEEE Trans. Syst. Man Cybern. PartB 39 (1) (2009) 281–288.

[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[12] M.A. Tahir, J. Kittler, F. Yan, Inverse random under sampling for class imbalance problem and its application to multi-label classification, Pattern Recognit. 45 (2012) 3738–3750.

[13] S. García, F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy, Evol. Comput. 17 (2009) 275–306.

[14] V.C. Chawla,K.W.Bowyer,L.O.Hall,W.P.Kegelmeyer,SMOTE:synthetic minority over-samplingtechnique,J.Artif.Intell.Res.16(2002)321–357.

[15] G.Wu,E.Y.Chang,Class-boundary alignment for imbalanced data set learning,in:Proceedings of International Conference on Machine Learning 2003 Workshop on Learning from Imbalanced DataSets II, Washington, DC

[16] C. L. Castro and A. P. Braga, "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data," IEEE Trans. Neural Netw. Learn. Sys., vol. 24, pp. 888–899, 2013.

[17] P.Domingos, Metacost: a general method for making classifiers cost-sensitive, in: Proceedings of the 5th ACM SIGKDD International Conferenceon Knowl-edge Discovery and DataMining, 1999, pp.155–164.

[18] M. Galar, A.Fernandez, E.Barrenechea, H.Bustince, F.Herrera, A review on ensembles for the class imbalance problem: bagging, boosting and hybrid-based approaches, IEEE Trans. Syst. Man. Cybern. PartC: Appl.Rev.42(4) (2012)463–484.

[19] X. Liu, J.Wu, Z. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern. B: Cybern. 39 (2009) 539–550.

[20] L. Breiman, Bagging predictors, Mach.Learn.24(1996)123–140.

[21] Y.Freund, R.E.Schapire, A decision-theoretic generalization of on-ine learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.

[22] N.V.Chawla, A.Lazarevic, L.O.Hall, K.W.Bowyer, Smoteboost: improving prediction of the minority class in boosting, in: Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases, 2003,pp.107–119.

[23] M. Galar, A.Fernandez, E.Barrenechea, F.Herrera, EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, PatternRecognit. 46 (2013) 3460–3471.

[24] Zou Q, Xie S, Lin Z, et al. Finding the Best Classification Threshold in Imbalanced Classification. Big Data Research, 2016, 5:2-8.