Random Forest Algorithm Based on GAN for Imbalanced Data Classification

Qihui Shu¹, Tianyu Hu¹ and Song Liu¹,*

¹ College of Computer Science and Technology, Qilu University of Technology (Shandong academy of sciences), Jinan, Shandong Province, 250000, China
*Corresponding author’s e-mail: liusong@qlu.edu.cn

Abstract. Data imbalance increases the difficulty of knowledge mining from data. Aiming at the problem of data imbalanced classification, a random forest algorithm based on GAN is proposed, which can effectively classify imbalanced data sets. The GAN model is used to generate a few class samples, which are mixed with the original data samples to form a new data set to reduce the imbalance of data. Then the new data set is divided into several data subsets with a balanced sample distribution. Finally, all the decision trees are collected to form the forest and the classification results are obtained. The random forest algorithm based on GAN improves the classification effect on minority classes, which makes the equilibrium rate of the dataset reach 50%. The algorithm in this paper has performed relevant experiments on multiple imbalanced datasets, the results show that the algorithm has good performance in data imbalance classification. The optimized algorithm after parallelization improves the running speed.

1. Introduction

With the increasing amount of data, big data is becoming more and more important. It has become the requirement and trend of the times to extract meaningful knowledge from big data. The phenomenon of imbalanced data makes it more difficult to mine knowledge from data. Imbalanced data classification exists in many areas of real life. For example, in the field of intrusion detection [1], abnormal behaviors are far less than normal behaviors, but correct identification of abnormal behavior is far more important than normal behavior. In the field of fault detection [2], the number of damage events is far less than that of normal events, but the detection of damage events is very important. Many unnecessary losses will be brought while the damage events cannot be diagnosed accurately. Due to the complex nature of unbalanced data, it is difficult for random forest algorithm to perform well in the recognition of minority classes. Such problems are pervasive in such fields as anomaly detection [3] or bank credit rating [4]. So this article will start from the perspective of imbalanced data classification to solve the problems.

There are two main solutions to traditional unbalanced classification, one is based on data level, and the other is based on algorithm level. The data-based approach starts with the data set. The imbalance degree of the data set can be reduced by changing the distribution of samples. The classic solutions to this are random oversampling, random undersampling and the methods of generating minority sample: SMPTE proposed in literature [5], adaptive oversampling method proposed in literature [6], for example. Algorithm-based methods are carried out from specific data mining algorithms. According to the problems encountered by the specific algorithm in solving the imbalanced classification problem, it is appropriate to modify the algorithm to some extent so that it
can solve the imbalanced classification problem, the main methods of which are cost-sensitive learning and integrated learning and so on. For instance, a random subspace AdaBoost algorithm (RSBoost) proposed in literature [7] trains the base classifier on each sub-training sample set, takes the base classifier with the largest mean of the minimum class interval as the selected classifier in this round, and iteratively form the final ensemble classifier.

In order to solve the problems, some domestic and foreign scholars have done a lot of research. But there are still many problems. Although the random forest algorithm based on SMOTE proposed by Fang Xiaonan in literature [8] can generate minority samples, it is easy for the random forest algorithm to fall into overfitting because the samples generated by SMOTE do not have diversity. The method proposed by Liu Yaojie in literature [9] comprehensively considers the sample proportion of the nodes in the random forest algorithm, and artificially increases the classification information of the minority class to increase the recognition rate of the minority class. However, artificially adding increases the classification information of the minority class is not an easy task. And if there are many random forest nodes, it is a laborious task to consider each node.

Deep learning has become popular in recent years due to the rapid development of big data and related hardware and software. Starting from the DNN (Deep Neural Network) in 2012, a series of CNN models such as VGG[10] and GoogleNet[11] have achieved great success in ImageNet classification tasks. Deep learning has achieved great success not only in image classification, but also in other aspects, such as target tracking, automatic driving, object recognition and so on. GAN (Generative Adversarial Network)[12] proposed by GoodFellow is a kind of generative model belonging to deep learning algorithms. GAN model is a generator that simulates a probability distribution of data through certain methods, and makes this probability distribution of data as consistent as possible with the probabilistic statistical distribution of some observed data. So far, GAN model has been widely used in machine vision, NLP, voice and other fields.

In order to solve the problems of random forest algorithm in imbalanced dataset, a random forest algorithm based on GAN((GAN-Random Forest)) is proposed in this paper. First, GAN is used to generate minority class samples. Second, the generated minority class samples are combined with the original data set to form a new dataset with balanced sample distribution. Then the new dataset is divided into several data subsets with balanced sample distribution. Each data subset corresponds to a decision tree. Finally, all the decision trees are aggregated to form the forest and the classification results are obtained.

2. Related algorithms and Algorithm parallel design

2.1. GAN Algorithm
GAN mainly generates two sub-models, a G generator (generative model) and a D discriminator (discriminative model). In the initial stage of the algorithm, G generation model will generate some data based on the given noise. The D discriminator will determine whether the generated data is real data or generated by the G generative model. The whole training process is like a game between two people. The purpose of the G generative model is to make the generated data as close to the real data as possible and cannot be easily identified by the D discriminative model. The purpose of the D discriminative model is to distinguish between real data and data generated by G generative model[13].

The pseudo-code of the algorithm is as follows:
For number of training iterations do
For k steps do
1. Collect m noise samples \( \{z^{(1)}, \ldots, z^{(m)}\} \) from noise space \( P_g(z) \).
2. Collect m data samples \( \{x^{(1)}, \ldots, x^{(m)}\} \) from the data set \( P_{data}(x) \).
3. Update the discriminator by promoting the random gradient, the specific formula is as follows:
\[
\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log (1 - D(G(z^{(i)}))) \right] \tag{1}
\]
End for
1. Collect m noise samples \( \{z^{(1)}, \ldots, z^{(m)}\} \) from the noise space \( P_g(z) \).
2. Update the generator by reducing the random gradient, the specific formula is as follows:
\[
\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log (1 - D(G(z^{(i)}))) \tag{2}
\]
End for

The algorithm flow of GAN is shown in figure 1 below:

![Image of GAN algorithm flow chart]

Figure 1. The flow chart of GAN algorithm

Although the data generated by GAN can pass for real data, it is not a mere reproduction of real data. It has a certain effect of data interpolation, which can be used as a method of data augmentation and can be used to better train various learning models in combination with other data [14].

2.2. Random Forest Algorithm Based on GAN(GAN-Random Forest)

Although the traditional random forest algorithm has good classification effect, there are still some problems to overcome. When dealing with imbalanced datasets, the traditional random forest algorithm has good classification effect on majority classes, but it does not have satisfactory effect on minority classes [15]. Taking a binary imbalance classification as an example to illustrate, as shown in figure 2, when the random forest algorithm runs to the bagging sampling stage (Here the first column is set to be the category labels, 1 is positive, 0 is negative, and the number of 1 is much larger than the number of 0). Because the dataset is imbalanced, the number of minority class samples is very small. Therefore, it is very likely that there are few or no minority class samples in each data subset extracted, which will cause the random forest algorithm to lose certain classification capabilities for minority classes.
To solve this problem, a random forest algorithm based on GAN is proposed in this paper. The specific improvement idea of the algorithm is:

First use the GAN model to generate minority class samples. More iterations of the GAN model are needed here because there is a certain gap between the samples generated by GAN model and the real samples at the beginning. And the minority class samples are collected until both the G generative model and the D discriminative model in the GAN model tend to be stable. Next, the collected minority class samples are combined with the original dataset to form a new and balanced dataset. Then the new dataset is divided into several balanced data subsets so that the balanced data subsets can participate in the construction of the random forest algorithm.

The pseudo-code of the GAN-Random Forest algorithm is as follows:

Input: the minority class samples, training dataset D, relevant parameters of GAN algorithm, relevant parameters of random forest algorithm

Output: random forest model

1. GAN is used to generate the minority class samples.
2. The minority class samples generated by GAN are merged with the original datasets to form a new data set with balanced data distribution.
3. The new dataset is divided into several balanced data subsets.
4. Each data subset corresponds to a decision tree.
5. Gathering all decision trees to generate random forests.

The process of the GAN-Random Forest algorithm is shown in figure 3:
2.3. Parallel Design of GAN-Random Forest Algorithm

In order to make GAN-Random Forest algorithm with the ability to process big data, the parallel design of the algorithm is implemented based on Tensorflow+Spark in this paper.

The parallel design of the GAN-Random Forest algorithm in this paper is divided into two stages. The first stage is the generation of the minority class samples. And the second stage is the construction stage of the random forest model.

The main purpose for the first stage is to generate some minority class samples, which mainly depending on GAN model. Therefore, the design idea of the first stage is mainly about HDFS+GAN model based on Tensorflow. The detailed design idea is shown in figure 4:

![Figure 3: The flow chart of GAN-Random Forest algorithm](image)

Figure 3 The flow chart of GAN-Random Forest algorithm

The main purpose for the second stage is to construct the random forest model. Therefore, the design idea of the second stage is mainly about HDFS+Spark based random forest model. The specific design idea is shown in figure 5:

![Figure 4: The parallelization Design flow chart of the first stage](image)

Figure 4 The parallelization Design flow chart of the first stage
The overall design idea of the optimized parallel GAN-Random Forest algorithm is as follows:

1. Firstly, the minority class samples of the dataset are read from the database and stored in HDFS. The GAN model reads the minority class samples from HDFS, then the GAN model generates more minority class samples.

2. The generated minority class samples are combined with the original dataset to form a new and balanced dataset. The new data set is divided into multiple balanced data subsets, then the multiple balanced data subsets are stored in HDFS to ensure that each partition stores a data subset.

3. Each data subset will construct a corresponding decision tree model according to the generation mode of the decision tree.

4. All the decision trees are brought together to form a forest.

The specific process of the overall GAN-Random Forest algorithm is shown in figure 6:

3. Experimental Demonstration

3.1. Experimental dataset

The datasets used in this article are public datasets mainly from the UCI public datasets and Kaggle data competition. They are mainly UCI_Bank Marketing, UCI_credit, Kaggle_Lending club and other imbalanced datasets. The specific datasets are shown in table 1:

| DataSet         | Sample size | Number of attributes | Number of categories |
|-----------------|-------------|----------------------|----------------------|
| Bank Marketing  | 41188       | 16                   | 2                    |
| Credit          | 284808      | 28                   | 2                    |
| Lending club    | 107866      | 75                   | 2                    |

Figure. 5 The parallelization design flow chart of the second phase

Figure. 6 The parallelization design flow chart of the overall algorithm
3.2. Experimental Process and Results

3.2.1. Generation of Minority Class Samples

First, the parameters of GAN model are set. It was found during the experiment that GAN needs to iterate for many times before the generated data can approach the real data. For example, a generated sample slowly approximates the real data distribution as the number of GAN iterations increases. As shown in the figure, the solid red line in the middle is the real data distribution (During each iteration, we choose different real data to judge the learning ability of GAN). The green dotted line is generated by GAN (Representing the data sample generated by GAN).

When GAN iterated for 100 times, the effect was not very good, as shown in figure 7. When GAN iterated to 700 times, the data generated by GAN gradually approached the real data, as shown in figure 8. When GAN iteration reached 1500 times, the data generated by GAN model was already similar to the real data, as shown in figure 9. When the GAN iteration reached 3000 times, the data generated by GAN made a just-as-good fake which was hard to detect, as shown in figure 10.

Therefore, for the optimized algorithm proposed in this paper, we took the minority class samples generated by GAN model iterating after 3000. Combining the generated minority class samples with the original data set for the next step of random forest algorithm, the equilibrium rate of the merged dataset reached 50%.

3.2.2. Comparative Experiment

The imbalance degree of the data set used in the experiment is shown in table 2:

| Name          | Data Size | Data Distribution | Imbalance Rate |
|---------------|-----------|-------------------|----------------|
| credit        | 284808    | 492, 284316       | 99.8%          |
| Bank Marketing| 41188     | 5361, 35827       | 87%            |
| Lending club  | 107866    | 7553, 100313      | 93%            |

In order to verify the superiority of our optimized random forest algorithm on imbalanced datasets, a variety of classification algorithms were chosen in this article for comparison. Among them were Naive Bayes, SVM, Logistic Regression, Decision Tree, Random Forest, Random Forest Algorithm...
based on SMOTE, Random Forest Algorithm based on Random Undersampling, Random Forest Algorithm based on Random Oversampling, and GAN-Random Forest.

The following tables are the performance of related algorithms on the UCI_Bank Marketing, UCI_credit and Kaggle_Lending club datasets, as shown in Table 3, Table 4, and Table 5:

Table 3 Performance of the algorithms on the UCI_credit dataset.

| Algorithm                      | AUC  | F1  |
|-------------------------------|------|-----|
| Naive Bayesian                | 0.817| 0.248|
| SVM                           | 0.524| 0.09 |
| Logistic Regression           | 0.779| 0.641|
| Decision Tree                 | 0.826| 0.725|
| Random Forest                 | 0.854| 0.804|
| SMOTE-Random Forest           | 0.866| 0.796|
| Undersampling-Random Forest   | 0.814| 0.725|
| Oversampling-Random Forest    | 0.769| 0.648|
| GAN-Random Forest             | **0.915** | **0.859** |

Table 4 Performance of the algorithms on the UCI_Bank Marketing dataset.

| Algorithm                      | AUC  | F1  |
|-------------------------------|------|-----|
| Naive Bayesian                | 0.774| 0.386|
| SVM                           | 0.478| 0.215|
| Logistic Regression           | 0.689| 0.655|
| Decision Tree                 | 0.798| 0.713|
| Random Forest                 | 0.855| 0.746|
| SMOTE-Random Forest           | 0.834| 0.796|
| Undersampling-Random Forest   | 0.811| 0.693|
| Oversampling-Random Forest    | 0.763| 0.688|
| GAN-Random Forest             | **0.894** | **0.859** |

Table 5 Performance of the algorithms on the Kaggle_Lending club dataset.

| Algorithm                      | AUC  | F1  |
|-------------------------------|------|-----|
| Naive Bayesian                | 0.655| 0.278|
| SVM                           | 0.549| 0.137|
| Logistic Regression           | 0.479| 0.341|
| Decision Tree                 | 0.464| 0.425|
| Random Forest                 | 0.778| 0.804|
| SMOTE-Random Forest           | 0.753| 0.717|
| Undersampling-Random Forest   | 0.791| 0.689|
| Oversampling-Random Forest    | 0.710| 0.598|
| GAN-Random Forest             | **0.929** | **0.884** |

It can be seen from the tables that the GAN-Random Forest algorithm has better AUC values and F1 values on the three unbalanced datasets of UCI_credit, UCI_Bank Marketing, and Kaggle_Lending club.

3.2.3. Operational Efficiency Comparison

As for the optimized algorithm in this paper, in the first stage it uses GAN to generate some minority class samples in TensorFlow, which does not involve the calculation task of Spark cluster. So the runtime by the algorithm in the first stage doesn’t change. While the second stage involves the parallelization of the random forest algorithm, so the runtime before and after the parallelization of the second stage algorithm is mainly compared in this article.
The sample size of the UCI_credit dataset was expanded to 600,000 and then divided into six data subsets of different sizes, with the data volume of 100,000, 200,000, 300,000, 400,000, 500,000, and 600,000 respectively for comparison experiments. The experiment results are shown in figure 11:

![Figure 11 The comparison diagram of running time](image)

It can be seen from Figure 11 that the parallelized algorithm has the faster running speed. So the parallel design of GAN-Random Forest algorithm based on Spark can handle big data more effectively.

4. Conclusion
In this article we used the GAN model to generate minority class samples, then through the combination of the generated minority samples with the original dataset to form the new and balanced dataset. With the new dataset divided into several data subsets to participate in the training, we parallelized the random forest algorithm on Spark to obtain faster running speed. The experiment results show that the algorithm proposed in this paper has good performance in minority class classification.

Acknowledgments
This work was supported by Science and Technology Development Plan for higher schools in Shandong Province under Grant J18KA360 and Key Research and Development Program in Shandong Province under Grant 2019GGX105010.

References
[1] Fei Z , De-Sheng Z .(2013) Network Intrusion Detection Based on the Integration of SVM and Genetic Algorithm.J. Journal of Qingdao University of Science and Technology(Natural Science Edition).
[2] Ya-Dan Z.(2016)Research on Key Technologies of Cloud Computing Platform Fault Detection.D. Beijing Jiaotong University.
[3] Yu-Ling Z,Chuan-Huan Y.(2018)Android malware outlier detection based on feature frequency.J. CAAI transactions on intelligent systems.13(2): 168–173.
[4] Wang H , Zhong J , Zhang D , et al.(2017)A new classification algorithm for the bank customer credit rating.In:2017 Ninth International Conference on Advanced Computational Intelligence (ICACI). Doha.IEEE.
[5] Demidova L , Klyueva I .(2017)SVM classification: Optimization with the SMOTE algorithm for the class imbalance problem.In:2017 6th Mediterranean Conference on Embedded Computing (MECO). Bar.IEEE.
[6] Şeyda Ertekin.(2013)Adaptive Oversampling for Imbalanced Data Classification.M. Information Sciences and Systems 2013. Springer International Publishing.
[7] Zong-Tang Z, Sen W, Shi-Lin S. (2019) An ensemble learning algorithm for unbalanced data classification. J. Journal of Shandong University (Engineering Science). 49(4): 8-13.
[8] Xiao-Nan F, Hua-Xiang Z, Shuang G. (2013) Web spam detection based on SMOTE and random forests. J. Journal of Shandong University (Engineering Science).
[9] Yao-Jie L, Du-Yu L. (2019) Research on Improved Random Forest Algorithm Based on Unbalanced Datasets. J. Computer Technology and Development. 029(006): 100-104.
[10] Simonyan, Karen, Zisserman, Andrew. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. J. Computer Science.
[11] Szegedy C, Liu W, Jia Y, et al. (2014) Going Deeper with Convolutions. J.
[12] Goodfellow, Ian. (2016) NIPS 2016 Tutorial: Generative Adversarial Networks. J.
[13] Gautam Ramachandra. (2017) Generative Adversarial Networks. C. ArXiv.
[14] Kun-Feng W, Wang-Meng Z, Ying T, Tao Q, Li L, Fei-Yue W. (2018) Generative adversarial networks: from generating data to creating intelligence. Acta Automatica Sinica. 44(5): 769−774.
[15] Xue L, Su-Wei Z. (2016) An new Algorithm for Imbalanced Data Based on Twice Random Forest. J. Soft. 37(7).