A software defect prediction method based on sampling and integration

Peng Shen, Xiaoming Ding, Xuyun Mu and Jianing Xu*

College of Computer and Information Science, Southwest University, ChongQing,China

*Corresponding author e-mail: xmding@swu.edu.cn

Abstract. This paper mainly analyzes the characteristics of software defect prediction from the perspective of machine learning, and proposes a semi-supervised software defect prediction method based on sampling and integration for the problem of class imbalance in software defect data and the incomplete classification of data sets. SISDP. SISDP firstly constructs a robust KNN marking model by taking a balanced sample of samples to mark a batch of unmarked data, and then iteratively adds the newly marked data to the original data set for the next marking model. , iterate until the data is marked. For the marked data set, the hybrid sampling algorithm is used to obtain the training set, and the integrated classification model composed of the multi-classification algorithm is classified and trained. SISDP not only reduces the interference of a few classes on the marking process, but also improves the generalization ability of the defect prediction model.

1. Introduction

In the traditional learning algorithm, the category labels of the training set samples are generally balanced, so the training classification model tends to have better classification prediction performance and has good generalization ability. However, in the actual situation of software defect prediction, the defect-free data instance is much larger than the defective data instance. When the traditional defect prediction model performs classification training and verification, even if all sample categories are predicted to be defect-free, the accuracy will still be Very high, this will be very unfavorable for the prediction of small class defective samples. The purpose of the defect prediction model is to identify defective samples as much as possible. Once the output from the defect prediction model is all non-defective, the generalization ability of the model is greatly reduced, and the prediction accuracy of the small sample is greatly reduced. It will be very low, naturally losing the meaning of the defect prediction model construction. Therefore, it is necessary to consider how to train a more suitable classification model in the unbalanced data set to greatly improve the prediction accuracy of small sample.

2. Related work

A series of related studies have emerged for the classification of unbalanced data sets. For example, the robust bounded logistic regression method proposed by Xu et al. [1] for the class imbalance problem is established. The method establishes robust bounded logistic regression with different error costs to reduce the combined effects of outliers and class imbalances. Tsai [2] et al. proposed cluster-based instance selection (CBIS) algorithm for undersampling, which combines cluster analysis
and instance selection. Cluster analysis groups most types of data samples, and instance selection filters from each group. The unrepresentative data samples show that the CBIS method has certain advantages over other undersampling methods.

For the unbalanced data in credit card fraud detection, Anis [3] proposed a similarity measurement method centered on Mahalanobis distance, which uses data as the center to find key samples and divides the majority of samples in the boundary by boundary. The example implements undersampling for most classes. By oversampling a few classes and determining the weights of a few classes based on the proximity and classification misjudgment rate, and thus increasing the attention to these few classes, this method effectively improves the prediction accuracy of a few classes.

The above studies are all about the problem of two-class imbalance, and the complex multi-class imbalance problem has also received extensive attention. In the medical field, Sellami [4] et al. proposed a robust deep convolutional neural network with batch weight loss for five-level heartbeat classification, which uses a batch weighted loss function to better quantify losses, thereby overcoming an imbalance between classes.

Although the classification performance on unbalanced data is not as good as the classification performance on the balanced data set, the former is more in line with the actual situation in the actual research work. Accurate predictions for a few categories have become more important in many areas, such as disease diagnosis, genetic information analysis, detection of fraudulent numbers, image recognition, and more. The importance of unbalanced data research has led to rapid development of research in this field, showing an eye-catching prospect.

3. Software defect prediction method based on sampling and integration

3.1. Sampling algorithm for unbalanced data

At present, the processing methods for unbalanced data can be mainly divided into three categories [5]: feature selection, data distribution adjustment, and classification model adjustment. The most intuitive and effective way to deal with unbalanced data is data distribution adjustment. Data distribution adjustment uses sampling method to achieve rebalancing sampling of unbalanced data. The sampling method is mainly divided into three categories: oversampling, undersampling and mixed sampling [6].

(1) Oversampling achieves the goal of balance by adding a small number of samples. Chawla et al. [7] proposed the Synthetic Minority Over-Sampling Technique (SMOTE), which uses the K-nearest neighbors of small class samples to compute neighbors and small class samples by a stochastic linear formula, and thus obtains new subclass samples. The SMOTE algorithm is based on the KNN algorithm to synthesize new samples in the neighborhood of small classes, which may cause certain sample overlap problems, and the selection of K also has certain blindness. Han [8] et al. proposed the Borderline-SMOTE algorithm for this kind of problem, which is to interpolate in the appropriate region of a few types of sample space, so that the synthesized data is more effective.

(2) Undersampling The purpose of class balancing is achieved by reducing the number of samples of most types. Liu [9] et al. proposed the EasyEnsemble and BalanceCascade algorithms to improve random undersampling. EasyEnsemble extracts multiple subsets from most classes and a plurality of balanced data sets with a few classes, and trains multiple balanced data sets. The classifier obtains an average value of the determination results of the common output of the plurality of classifiers. BalanceCascade removes most of the class samples that have been correctly classified according to the training order, and disguise increases the weight of the most difficult classes in the majority class. Both of these algorithms greatly improve the loss of most types of sample information in random undersampling.

(3) Mixed sampling combines oversampling and undersampling. Both oversampling and undersampling can rebalance the dataset. Oversampling increases the size of the original dataset by adding a small number of samples. Although data balance is achieved, it is easy to overfit and increase training time. Undersampling reduces the size of the original data set by reducing the majority of the sample size. Although data balance is achieved, most of the class information is easily lost. Therefore, in order to make up for the shortcomings of both, Zhu Ming et al. [10] proposed an SVM classification
algorithm based on random undersampling and SMOTE oversampling. The algorithm performs de-redundant undersampling on most classes and performs on a few classes. This not only reduces overfitting, but also preserves as much useful information as possible in most classes.

3.2. ensemble learning

In machine learning, the ensemble algorithm is a typical collective intelligence algorithm. The core idea is to combine multiple classification algorithms and participate in the final output decision to improve classification performance. At present, there are three popular integrated algorithms: Bagging, which divides the data set into multiple training subsets and correspondingly trains multiple classification models. The typical algorithm in this method is random forest algorithm; (Boosting), the method is to form a sequence by training multiple models, and let each model in the sequence correct the error of the previous model. The typical algorithm in this method is AdaBoost algorithm [11]; voting method (Voting), the method uses the same training set to train multiple classification models at the same time, and then counts the number of output categories of each classifier, and determines the final output category according to the majority.

3.3. Algorithm Description

Based on the above analysis, this paper proposes the SISDP algorithm, which is divided into three stages according to the theoretical needs. The first stage: the marking model based on KNN algorithm marks the unlabeled defect samples; the second stage: through oversampling and undersampling, the combined mixed sampling method is sampled to form a new class-balanced data set; the third stage is to form an integrated classifier by integrating CART, LOGISTIC, and SVM classification algorithms. The algorithm is described as follows:

SISDP algorithm

Input: \( SDD_n = \{X_{mxn}, Y_{1x1}\} \): incompletely labeled defect data set, \( Y \) : label, \( l < m \)
Output: \( SDD_n^* = \{X_{mxn}, Y_{mx1}\} \): completely labeled defect data set, Classifier: \( H(x) \)

1 begin
2 The data set is preprocessed, and the label \{Y, N\} is digitized as \{0, 1\}; then the feature data in \( SDD_n \) is normalized: \( X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \)
3 \( SDD_n = SDD_n1 + SDD_n2; \) \( SDD_n1 = \{X_{1xn}, Y_{1x1}\} \) is tagged data, \( SDD_n2 = \{X_{uxn}\} \) is unlabeled data, \( l+u=m \).
4 for \( X_i \) in \( SDD_n2 \) do:
5 for \( X_j \) in \( SDD_n1 \) do:
6 \( d = \sqrt{(X_i - X_j)^2} \), \( d \) is the distance between \( X_i \) and \( X_j \);
7 end for
8 Select K training objects closest to the unlabeled sample \( X_i \) from \( SDD_n1 \)
\( c_{X_i} = \arg \max_{v \in Y} \sum_{X_j \in K} I(v = \text{class}(c_{X_j})) \), \( SDD_n1 \). append(\( X_i, c_{X_i} \))
\( I(v = \text{class}(c_{X_j})) \) is the indicator function and returns 1 if its value is true, otherwise it returns zero.
9 return \( SDD_n = \{X_{mxn}, Y_{mx1}\} \)
10 for $S_i$ in $SDD^*_n$: if ($Y_i$ == '0') $SDD^*_\text{large}.append(S_i)$, else $SDD^*_\text{small}.append(S_i)$

11 $SDD^*_\text{balance}$ = SMOTE($SDD^*_\text{small}$) + UnderSampling($SDD^*_\text{large}$)

12 Divide $SDD^*_\text{balance}$ into training set $TD_n = \{X_k \times n, Y_k \times 1\}$ and verification set $VD_n = \{X_l \times n, Y_l \times 1\}$, with $TD_n$. At the same time, train three learners $h_{\text{CART}}$, $h_{\text{LOGISTIC}}$ and $h_{\text{SVM}}$, and combine the three learners into the final learning model $H(x)$. $H(x)$ is as follows:

$$H(x) = \arg\max_{class_j \in Y} \sum (g(h_i(x), class_j))$$

Where $h_i(x)$ represents the classification result of the i-th learner, $class_j$ represents the mark of the j-th sample, and $g(h_i(x), class_j)$ represents the indication function.

$$g(H(x), c) = \begin{cases} 0, & h_i(x) \neq class \\ 1, & h_i(x) = class \end{cases}$$

13 return $H(x)$

3.4. Experiment and Analysis

In order to meet the subject of this paper, the experiment selected three unbalanced data sets in the MDP defect database: MC1, PC1, PC2 for experiments. The unbalanced proportion of the MC1 data set is 1:142, the unbalanced proportion of the PC1 data set is 1:12, and the unbalanced proportion of the PC2 data set is 1:100.

During the experiment, it is necessary to perform the *, EasyEnsemble algorithm, RusTri algorithm [12] and SISDP algorithm on the three unbalanced data sets of MC1, PC1 and PC2 respectively.

| Algorithms | * | EasyEnsemble | RusTri | SISDP |
|------------|---|--------------|--------|-------|
| Precision  | 0.93 | 0.98 | 0.97 | 0.99 |
| Recall     | 0.92 | 0.99 | 0.95 | 0.99 |
| F1-score   | 0.92 | 0.98 | 0.93 | 0.99 |

Table 1. Comparison of experimental results based on sampling and integration on MC1

| Algorithms | * | EasyEnsemble | RusTri | SISDP |
|------------|---|--------------|--------|-------|
| Precision  | 0.90 | 0.92 | 0.91 | 0.92 |
| Recall     | 0.88 | 0.87 | 0.90 | 0.91 |
| F1-score   | 0.88 | 0.89 | 0.90 | 0.91 |

Table 2. Comparison of experimental results based on sampling and integration on PC1

| Algorithms | * | EasyEnsemble | RusTri | SISDP |
|------------|---|--------------|--------|-------|
| Precision  | 0.94 | 0.97 | 0.96 | 0.98 |
| Recall     | 0.94 | 0.96 | 0.97 | 0.99 |
| F1-score   | 0.94 | 0.98 | 0.98 | 0.99 |

Table 3. Comparison of experimental results based on sampling and integration on PC2
Table 1, Table 2, and Table 3 above show that the SISDP algorithm proposed in this paper has obvious advantages over the other three methods. Based on the average of the evaluation indicators on the three data sets, it is found that the average values of Precision, Recall, and F1-score output by the SISDP algorithm are 0.96, 0.96, and 0.96, respectively, which are higher than the *precision, Recall, and F1-score averages of the output: 0.92, 0.91, 0.91; the values of Precision, Recall, and F1-score above the EasyEnsemble output: 0.95, 0.94, 0.95; the values of Precision, Recall, and F1-score above the RusTri output are 0.95, 0.94, and 0.94. The experimental output data objectively shows that the performance of the SISDP algorithm is better than the other three methods.

In Figure 1, it can be objectively and clearly seen that the evaluation index polyline of the SISDP algorithm is higher than the evaluation index polyline of the three methods of *, EasyEnsemble and RusTri, which fully validates the semi-supervised prediction algorithm SISDP based on sampling and integration proposed in this chapter. Missing category labels, sample imbalances, and single learners. The stability of the SISDP algorithm is fully demonstrated by comparison experiments on three data sets. In general, the SISDP method proposed in this chapter can achieve better experimental results in software defect prediction, which reflects the effectiveness of SISDP algorithm in software defect prediction.

4. Conclusion

This chapter proposes a semi-supervised software defect prediction (SISDP) based on sampling and integration, including: sampling algorithm for unbalanced data, classifier integration algorithm, and robust KNN marking model based on semi-supervised learning. In the experiment, the comparison experiments on MC1, PC1 and PC2 datasets show that SISDP has certain advantages in the prediction of semi-supervised software defects based on sampling and integration.

References
[1] Xu G, Hu B G, Principe J C. Robust bounded logistic regression in the class imbalance problem[C]//2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016: 1434-1441.
[2] Tsai C F, Lin W C, Hu Y H, et al. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection[J]. Information Sciences, 2019, 477: 47-54.
[3] Anis M. research on Unbalanced Data in Credit Card Fraud Detection Based on Proximity Resampling and Classifier Sorting [D]. 2018.
[4] Sellami A, Hwang H. A robust deep convolutional neural network with batch-weighted loss for heartbeat classification[J]. Expert Systems with Applications, 2019, 122: 75-84.
[5] Zhao Nan, Zhang Xiaofang, Zhang Lijun. Review of Unbalanced Data Classification Research [J]. Computer Science, 2018, 45(S1): 22-27.

[6] Xiang Hongxin, Yang Yun. Overview of Unbalanced Data Mining Methods [J]. Computer Engineering and Applications, 2019, 55(04): 1-16.

[7] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[8] Han H, Wang W Y, Mao B H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning [C] // International conference on intelligent computing. Springer, Berlin, Heidelberg, 2005: 878-887.

[9] Liu X Y, Wu J, Zhou Z H. Exploratory Undersampling for Class-Imbalance Learning [J]. IEEE TRANSACTIONS ON CYBERNETICS, 2009, 39(2): 539-550.

[10] ZHU Ming, TAO Xinmin. Unbalanced SVM Classification Algorithm Based on Random Downsampling and SMOTE [J]. Information Technology, 2012(1): 39-43.

[11] Dai Wenyuan. Research on migration learning algorithm based on instance and feature [D]. Shanghai: Shanghai Jiaotong University, 2009.

[12] Ma Ying. Research on software defect prediction technology based on machine learning [D]. University of Electronic Science and Technology, 2012.