A Hybrid Feature Selection Method for Software Defect Prediction

Lina Jia
School of Information Science and Engineering Zaozhuang University Zaozhuang, China
jlnkejian@163.com

Abstract. Software Defect Prediction (SDP) is one of the important ways of software quality assurance, which uses the metric data to predict whether software module is defect. The quality of data influences the perfection of the prediction model. The high latitude containing some unnecessary features is one of the quality problem that dataset. To solve this problem, we proposed a hybrid feature selection (HFS) method combined different feature sorting technology. Firstly, we calculate the values of each feature include chi-squared (cs), Information gain (IG) and Pearson Correlation coefficient, respectively. Secondly, we sort the features based on the ranking of the three values to select features. Finally, we use the random forest to build the model. In order to validity the approach, we did experiments on 5 datasets in NASA. The result shows that our approach can select a smaller subset of features to improve the preformation in F-measure.

1. Introduction
SDP is one of the important ways of software quality assurance, which has become a very important research topic in software engineering in recent years. The learning algorithm of software defect prediction model is established on the basis of defect dataset which may contain some unnecessary information. So before training, we should deal with or delete the data with quality problem.

The High latitude is one of the data quality problems that contain some unnecessary features including unrelated features and redundant features. For unrelated features, the performance of the prediction model is not affected, but the processing time of the model is increased. The redundant feature set adds more noise to the prediction model even if it adds useful information [1]. Moreover, early studies have confirmed that irrelevant features lead to reduce accuracy of prediction model [2-3]. Therefore, feature selection can promote the understanding and visualization of data, reduce training time, reduce test cost and storage demand, and improve the performance of prediction model [4].

At present, feature selection method can be divided into two categories: filtering and wrapper [5]. The filtering method is to evaluate the feature set according to the general characteristics of the training dataset, regardless of the classification algorithm used. The filtering method is more general and less expensive, but the accuracy is not guaranteed. Khoshgoftaar [6] conducted experiments using 16 software datasets to compare seven filtering feature sorting technologies, including chi-squared (CS), information gain (IG), gain ratio (GR), symmetric uncertainty (SU) and RelieF (two variants)(RF and RFW). He [7] proposed a feature selection method of Topk. Issam [8] adopt the
method of feature selection based on the greedy method based on Pearson coefficient. Ömer [9] extracted the features based on the related feature selection filter (CFS), and build the prediction model by the improved naive bayes (NB) considering the probability of the correlation between metrics.

The wrapper method selects the best feature subset according to the effect of classification, which can achieve higher prediction accuracy, but increase the computational cost. Song [10] use the wrapper method to make feature selection on the training set based on static code metrics. Marco [11] used wrapper method to select the feature subset based on code churns metric and entropy metric. Chen [12] transformed the defect prediction into a multi-objective optimization problem, and considered the high latitude problem, using the Wrapper MOFES method for feature selection. Juan [13] proposed a learning plan, using the genetic algorithm to select the best learning arrangement, and using different Wrapper methods to select the attributes. Wang [14] used five different Wrapper methods for feature extraction according to different classification indexes.

In contrast to the above research work, we present a hybrid feature selection method based on filtering method. Firstly, we use chi-squared (cs), Information gain (IG) and Pearson Correlation coefficient as feature ranking evaluator to give an evaluation values of each feature, respectively. Secondly, based on the evaluation values, we sort features using the average ranking of each feature. Finally, we trained the dataset with selected feature using RF to predict the software module.

The structure of this paper is as follows: In Section 2, we describe details hybrid feature selection method; experimental setup is presented in section 3; Case study results are presented in section 4; we conclude the paper and provide insights for future work in section 5.

2. Our approach

2.1. The framework of hybrid feature selection

The framework of hybrid feature selection is shown in Figure 1. Assume that the defect data has n features. Firstly we sort the n features using chi-squared (cs), Information gain (IG) and Pearson Correlation coefficient, respectively, and the i feature ranks n1, n2 and n3. Then we calculate the average ranking ni of the feature: \( n_i = (n_1 + n_2 + n_3) / 3 \). Finally, we sort the value of ni to rank the features, and use the strategy of Topk to select the features.

![Figure 1. The framework of our approach.](image-url)
2.2. Implementation details

The specific implementation details of the hybrid feature selection are shown in algorithm 1.

Algorithm 1 Hybrid feature selection

Input: A training dataset D

Feature set $F=\{f_1,f_2,\ldots,f_n\}$

Three calculation methods of attribute selection:
- Chi-Squared, Info Gain and Pearson Correlation coefficient

Output: subset of features

1. Using Chi-Squared to rank each feature, and the ranking of each feature is: $C=\{c_1,c_2,\ldots,c_n\}$
2. Using Information gain to rank each feature, and the ranking of each feature is: $I=\{i_1,i_2,\ldots,i_n\}$
3. Using Pearson Correlation coefficient to rank each feature, and the ranking of each feature is: $P=\{p_1,p_2,\ldots,p_n\}$
4. For each feature $i$, we calculate the average ranking:
   $$n_i=\frac{c_i+i_i+p_i}{3}$$
5. Sort the feature based on the average ranking
6. Set the $K$ value to get the Top $K$ features $F_{best}$
7. return $F_{best}$

1) Feature selection method of Chi-Squared

Feature selection method of Chi-Squared is by calculating the Chi-Square value of each feature to get the feature subset. If the smaller the Chi-Square value is, the smaller the relation is. However, they assume that the feature is no relationship with target characteristics.

The Chi-Square value is calculated as in

$$\chi^2 = \sum \left( \frac{(Y_a - Y_p)^2}{Y_p} \right)$$

Where $Y_a$ is the actual value of each instance, $Y_p$ is the prediction value of each instance.

2) Feature selection method of Info Gain

Feature selection method of Info Gain is by evaluating the information gain of each attribute associated with the classification. The bigger the Info Gain is, the more import the feature is. The Info Gain is calculated as the following.

$$\text{Info Gain(class, Attribute)} = H(\text{class}) - \frac{H(\text{class|Attribute})}{H(\text{Attribute})}$$

Where $H(X)$ is the entropy of a feature. Assume that $P(X)$ is the prior probability of $X(x \in \mathcal{X})$, then $H(X)$ is calculated as in:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

$H(\text{class|Attribute})$ is the entropy of class given the value of Attribute. The calculation is as

$$H(\text{class|Attribute}) = -\sum_{y \in \text{Attribute}} p(y) \sum_{x \in \text{class}} p(x|y) \log_2 p(x|y)$$

3) Feature selection method of Pearson Correlation coefficient

The feature selection method of Pearson Correlation coefficient is by evaluating the worth of an attribute by measuring the correlation (Pearson’s) between it and the class.
Assume that the Feature is $f_i$ and the class is $C$. The number of the instances is $l$ in the training dataset. $f_{ij}$ represents the value of $f_i$ in instance of $j$. $C_j$ represents the label of the instance of $j$. $\bar{f}_i$ is the average value of $f_i$, and $\bar{C}$ is the average value of class $C$. The calculation is as

$$r_{f_i,C} = \frac{\sum_{j=1}^{l}(f_{ij} - \bar{f}_i)(C_j - \bar{C})}{\sqrt{\sum_{j=1}^{l}(f_{ij} - \bar{f}_i)^2} \sqrt{\sum_{j=1}^{l}(C_j - \bar{C})^2}}$$

(5)

$r_{f_i,C}$ is the correlation between feature of $f_i$ and class $C$ and the value is between -1 and 1. The higher the value is, the more the correlation between the feature and class is.

3. Experiments

To verify the effectiveness of the approach, we conducted experiments on six projects from NASA. The design of the experiment has mainly considered from two aspects:

- Compared with some benchmark methods, can our approach select a smaller subset of features?
- Compared with some benchmark methods, can our approach effectively improve the performance of defect prediction model?

3.1. Experimental object

The datasets of the experiment come from the actual development projects in NASA, and can cover different types of defect prediction datasets well. The specific distribution is shown in Table 1.

| Projects | Number of attribute | Number of instances | Defective(%) |
|----------|---------------------|---------------------|--------------|
| CM1      | 6                   | 36                  | 7.327        | 8.116        |
| KC3      | 10                  | 36                  | 11.194       | 12.186       |
| MC1      | 14                  | 36                  | 15.1988      | 16.23        |
| MC2      | 18                  | 36                  | 19.125       | 20.352       |
| MW1      | 22                  | 36                  | 23.253       | 24.106       |
| PC1      | 26                  | 36                  | 27.705       | 28.86        |

3.2. Performance measure

Performance evaluation is based on evaluation indicators of information retrieval on defect prediction model, including accuracy, recall and precision, F-measure and AUC (area under ROC curve). There are some datasets of class imbalance in the selection datasets in NASA. AUC can effectively avoid the effects of class imbalance, so we choose the AUC to evaluate the performance of defect prediction model. AUC is the area under ROC curve. In ROC curve, X-axis represents the FPR (false positive rate) and Y-axis represents the TPR (true positive rate). The AUC is between 0 and 1. The bigger the value is, the better the performance of model is.

In order to evaluate the performance of defect prediction model, we use 10 fold cross-validation way.

3.3. Experiment design

We mainly considered random forest (RF) [15] as the classifier in our defect prediction model. We did experiments on weka, and the parameters were evaluated by default. In empirical, we mainly used three benchmark method comparisons with our approach:

- A approach based on chi-squared (CS). Use the features selected by chi-squared to train the defect prediction model.
- A approach based on Information gain (IG). Use the features selected by Information gain to train the defect prediction model.
A approach based on Pearson Correlation coefficient (PC). Use the features selected by Pearson Correlation coefficient to train the defect prediction model.

3.4. Results analysis

1) In view of RQ 1

We have do experiment using Topk that the value of k is 10%, 20%, 30%, 40%, 50%, 70%, 90%, and 100%. The result is shown in TABLES II-V. For each project, the optimal result is bolded.

Specifically, for CM1 project when the AUC is optimal, the proportion of features selected by HFS is 70%, and the proportion of features selected by chi-squared, Information gain and Pearson Correlation coefficient are both 50%. For KC3 project when the AUC is optimal, the proportion of features selected by HFS is 90%, and the proportion of features selected by chi-squared, Information gain and Pearson Correlation coefficient are both 100%. For MC1 project when the AUC is optimal, the proportion of features selected by HFS is 50%, and the proportion of features selected by chi-squared, Information gain and Pearson Correlation coefficient are 50%, 50% and 40%. For MC2 project when the AUC is optimal, the proportion of features selected by HFS is 40%, and the proportion of features selected by chi-squared, Information gain and Pearson Correlation coefficient are 40%, 40%, and 20%. For MW1 project when the AUC is optimal, the proportion of features selected by HFS is 70%, and the proportion of features selected by chi-squared, Information gain and Pearson Correlation coefficient are 70%, 90% and 70%. For PC1 project when the AUC is optimal, the proportion of features selected by HFS is 70%, and the proportion of features selected by chi-squared, Information gain and Pearson Correlation coefficient are 50%, 50%, and 40%.

Through the above analysis, the proportion of features selected by HFS is about the same as the other three.

2) In view of RQ 2

For each project, we draw different proportions of AUC using different methods. Figures 2-7 show the different.

![Figure 2. The comparison diagram of AUC for CM1.](image)
Specifically, for CM1 project, the optimal of AUC by HFS is 0.726. However, the optimal of AUC by chi-squared, Information gain and Pearson Correlation coefficient are 0.709, 0.711, and 0.719. For KC3 project, the optimal of AUC by HFS is 0.725. However, the optimal of AUC by chi-squared, Information gain and Pearson Correlation coefficient are 0.713, 0.679, and 0.695. For MC1 project, the optimal of AUC by HFS is 0.907. However, the optimal of AUC by chi-squared, Information gain and Pearson Correlation coefficient are 0.904, 0.904, and 0.908. For MC2 project, the optimal of AUC by HFS is 0.779. However, the optimal of AUC by chi-squared, Information gain and Pearson Correlation coefficient are 0.78, 0.78, and 0.738. For MW1 project, the optimal of AUC by HFS is 0.73. However, the optimal of AUC by chi-squared, Information gain and Pearson Correlation coefficient are 0.742, 0.704, and 0.704. For PC1 project, the optimal of AUC by HFS is 0.88. However, the optimal of AUC by chi-squared, Information gain and Pearson Correlation coefficient are 0.882.

Through the above analysis, compared with other three approaches, we see that the optimal of AUC by HFS is the best for most projects. So our approach can get the best value using the same proportion of features.
4. Conclusion
SDP is one of the important ways of software quality assurance, which uses the metric data to predict whether software module is defect. In this paper, we proposed a hybrid feature selection approach. Firstly, we calculate the values of each feature using chi-squared (cs), Information gain (IG) and Pearson Correlation coefficient, respectively. Secondly, we sort the features based on the ranking of the three values to select features. Finally, we use the random forest to build the model. We verify the effectiveness of our approach based on actual development projects in NASA. Our approach can get the optimal results using the same proportion of features. In the next step of study, we will consider how we can use the least features to get the optimal result?
The proportion of features (%)

AUC

CS
IG
PC
HFS

PC1

Figure 7. The comparison diagram of AUC for PC1.

References
[1] Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research, 2004, 5(12): 1205-1224.
[2] Kehan G, Khoshgoftaar TM, Wang H, Seliya N. Choosing software metrics for defect prediction: an investigation on feature selection techniques. Software Practice and Experience, 2011.41(5):579-606.
[3] Rodrigez D, Ruiz R, Cuadrado-Gallego J, Aguilar-Ruiz J, Garre M. Attribute selection in software engineering datasets for detecting fault modules//Proceedings of the 33rd EUROMICRO conference on software engineering and advance applications. Lubeck, Germany, 2007:418-423.
[4] Peng Y, Wu Z, Jiang J. A novel feature selection approach for biomedical data classification. Journal of Biomed Inform, 2010, 43(1):15-23.
[5] G. Chandrashekar, F. Sahin, A survey on feature selection methods. Journal of Computers and Electrical Engineering, 2014, 40(1):16-28.
[6] Taghi M. Khoshgoftaar, K. Gao, A. Napolitano. An empirical study of feature ranking techniques for software quality prediction. International Journal of Software Engineering and Knowledge Engineering, 2012, 22(02):161–183.
[7] Peng He, Bing Li, Xiao Liu, Jun Chen, Yutao Ma. An empirical study on software defect prediction with a simplified metric set. Information and Software Technology, 2014, 59(C):170-190.
[8] Issam H. Laradji, Mohammad Alshayeb, Lahouari Ghouti. Software defect prediction using ensemble learning on selected features. Information and Software Technology, 2015, 58(2):388-402.
[9] Ömer Faruk Arar, Kürs, at Ayan. A feature dependent Naive Bayes approach and its application to the software defect prediction problem. Applied Soft Computing, 2017, 59(10):197-209.
[10] Qinbao Song, Zihan Jia, Martin Shepperd, Shi Ying, and Jin Liu. A General Software Defect-Proneness Prediction Framework. IEEE Transaction on Software Engineering, 2011, 37(3):356-370.
[11] Marco D'Ambros, Michele Lanza, Romain Robbes. Evaluating Defect Prediction Approaches: A Benchmark and an Extensive Comparison. Empirical Software Engineering, 2012, 17(4-5):531-577.
[12] Xiang Chen, Yuxiang Shen, Zhanqi Cui, Xiaolin Ju. Applying Feature Selection to Software Defect Prediction using Multi-objective Optimization//Proceedings of 2017 IEEE 41st
Annual Computer Software and Applications Conference. Turin, Italy, 2017:54-59.

[13] Juan Murillo-Morera, Marcelo Jenkins. A Software Defect-Proneness Prediction Framework: A new approach using genetic algorithms to generate learning schemes//Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering, 2015:445-450.

[14] Huanjing Wang, Taghi M. Khoshgoftaar, Amri Napolitano. Choosing the Best Classification Performance Metric for Wrapper-based Software Metric Selection for Defect//Proceedings of the International Conference on Software Engineering and Knowledge Engineering, 2014:540-545.

[15] Q He, B Shen, Y Chen. Software Defect Prediction Using Semi-Supervised Learning with Change Burst//Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference. Atlanta, GA, USA, 2016:113-122.