Software Fault Prediction Model using Clustering Algorithms Determining the Number of Clusters Automatically

Mikyeong Park and Euyseok Hong*

School of Information Technology, Sungshin Women’s University, Korea
parkmikyeong@gmail.com, hes@sungshin.ac.kr

Abstract

Software fault prediction models using supervised learning cannot be applied when training data are not present. In this case, new models using unsupervised learning such as clustering algorithms are quite necessary. Nevertheless, there exist very few studies about unsupervised models because it is difficult to construct the models. One of the difficulties is to decide the number of clusters. To solve this problem, we build unsupervised models using clustering algorithms, EM and Xmeans, which determine the number of clusters automatically and compare them with results of earlier studies. Experimental results show the Xmeans model outperforms the other models.

Keywords: fault prediction model, unsupervised learning, clustering, number of clusters

1. Introduction

In general, a software fault prediction model means a classification model which determines fault-proneness of input modules. It enables improving the performance of a test process, constructing high reliable systems and allocating resources appropriately.

![Figure 1. Fault Prediction Model](image)

Most previous studies of software fault prediction models have focused on supervised models using training data. These studies used statistical or artificial intelligence techniques such as logistic regression, bayesian model, discriminant analysis, classification tree, neural network, Support Vector Machine, Case Based Reasoning and so on [1, 2]. However, most organizations do not have previous data for input data of the prediction model or otherwise, a current project is changed not enabling the use of the previous data. That is, because of dependence on training data, it is difficult for current systems to be applied to previous models. In these cases when training data are in the absence, a new model using an unsupervised or semi-supervised learning method which does not or partially use previous data for modeling is needed [1], referred to the importance of the fault prediction model using unsupervised learning for the future works in this field.

Usually an unsupervised learning method uses clustering techniques such as k-means, hierarchical, DBSCAN, OPTICS, Cobweb and CLOPE algorithms. Clustering algorithms

* Corresponding author
group similar instances into same clusters. The issue here is that the number of clusters should be selected heuristically and the number affects the performance of the prediction model. Besides determining the number makes experts do laborious work changing it repeatedly. The objective of our study is to build fault prediction models using various clustering algorithms which select the number of clusters without experts, to evaluate these models and to compare them with models of earlier study.

2. Related Work

Despite the need for unsupervised models, there are only a few studies about prediction models using unsupervised learning because a process of building the unsupervised models is difficult and the unsupervised models significantly underperform supervised models. In [3, 4], Neural-Gas and K-means algorithms were used. The performance of Neural-Gas model is better than that of K-means model in overall error rate but not in speed. A model using SOM network was proposed in [5] and compared with back-propagation neural network model. However the SOM model was not able to be verified on real data set. These approaches are dependent on experts who label clusters as fault-prone or not and have to select the number of clusters heuristically. As referred to earlier, determining the number of clusters has been on the rise a serious issue. On the contrary, models of [6, 7] did not require to select the number of clusters. A model using X-means algorithm was proposed in [6] and it outperforms the other models using Fuzzy C-means, K-means algorithms and Pure Metrics Thresholds methods. In [7], a model using QDK(Quad Tree-based K-means) algorithm based on K-means was proposed and the performance of the model is better than that of K-means, GM(Global K-means), DD(SAS2004) initialization algorithms. In our study, we focus on the models determining the number of clusters automatically. Recently the other direction related the prediction model is about semi-supervised models [8, 9]. The models are useful when training data exist partially because they are trained by a small set of labeled data and a large set of unlabeled data. This means the semi-supervised model can remedy the weakness of the supervised and unsupervised model.

3. Model Construction

3.1. Clustering Algorithms

The aim of our study is building fault prediction models using clustering algorithms that decide the number of clusters for itself and evaluating these models. We use WEKA data mining tool [10]. WEKA is a popular tool that supports lots of data mining algorithms. WEKA provides clustering algorithms that do not need to input the number of clusters (thus, the number of clusters is determined automatically) - CLOPE, Cobweb, DBSCAN, EM, OPTICS and X means. However, only EM and X means algorithms are used for constructing models in this study because the selection of other parameter values has a large impact on the number of clusters for CLOPE, Cobweb, DBSCAN, OPTICS algorithms. Selecting the values seems like that the number of clusters is determined heuristically and this hinders our purpose.

3.2. Process of Model Construction

When constructing unsupervised models using clustering algorithms, data sets have no class label (fault or non-fault) of each module. Figure 2 shows the process of constructing a prediction model using a clustering algorithm from data preprocessing to fault prediction phase. After preprocessing input data, a clustering algorithm is performed for input data in
clustering phase and the clusters are analyzed by experts in analysis phase. According to the analysis, experts label as fault-prone cluster or non fault-prone cluster on each cluster. These labeled clusters are estimated in accordance with characteristics and statistics of each cluster. Like Figure 2, the model is completed through clustering phase to analysis phase repeatedly until the best result come out (in most cases, the process is repeated by changing the number of clusters).

![Figure 2. The Process of Constructing Unsupervised Model](image)

Mostly, clustering algorithms do not determine the number of clusters. If the number of clusters is too small, the probability of grouping fault prone modules and non fault prone modules into same clusters is higher. In the opposite case, experts should take a lot of effort in analysis phase. For eliminating these problems and selecting a reasonable number, we use clustering algorithms deciding the number of clusters for itself. The phases in gray area are not performed in our study. That means it can remove inconvenient phases. There is no necessity to repeat from clustering phase to analysis phase for the selection of the cluster number making the best result. Thus, experts only determine fault-prone or not once without repetition. Furthermore it is no need for experts to evaluate the clusters in analysis phase.

4. Experimental Study

4.1. Experimental Setting

The data sets we used for experiments are three PROMISE repository data, AR3, AR4 and AR5, collected from a Turkish white-goods manufacturer [11]. Each data set includes 29 metrics and they consist of traditional procedural metrics (LOC, Halstead and McCabe metrics). The AR3 has 63 modules consisting of 55 non-fault modules and 8 fault modules. The AR4 has 107 modules consisting of 87 non-fault modules and 20 fault modules and the AR5 has 36 modules consisting of 28 non-fault modules and 8 fault modules. The reason why we select these data sets is that models of [6, 7] compared with our models used AR3, AR4 and AR5.

After normalization of data sets ranging from 0 to 1, the data are applied to all methods of attribute selection supported by WEKA. It is necessary to reduce the dimensionality of input data by removing redundant attributes for better performance of clustering algorithms. As a result of attribute selection, the most significant method is CfsSubsetEval. Therefore we experiment two cases: all attributes and attributes selected by CfsSubsetEval. CfsSubsetEval evaluates a subset of attributes by considering the predictive ability of each attribute along with the degree of redundancy between them [12].

4.2. Performance Measure

There are many performance measures of a prediction model such as Precision, Recall, F-Measure, and AUC. Prediction errors may also be symbolic of a performance measure. In this study, we use False Positive Rate (FPR), False Negative Rate (FNR) and Total Error Rate (TER) to evaluate models because models of [6, 7] compared with our models used TER,
FPR and FNR for model evaluation. TER is the percentage of mislabeled modules and FPR is the percentage of non fault-prone modules mislabeled as fault-prone. FNR is the percentage of fault-prone modules mislabeled as non fault-prone. Fault modules which pretended non fault-prone modules are difficult to be detected and cause heavy cost when detected after finishing a project. That is why FNR is much more serious than FPR. Figure 3 describes TER, FPR, FNR and confusion matrix.

\[ \text{TER} = \frac{B+C}{A+B+C+D} \]
\[ \text{FPR} = \frac{B}{B+D} \]
\[ \text{FNR} = \frac{C}{A+C} \]

Figure 3. Confusion Matrix and Definition of TER, FPR, FNR

4.3. Experimental Result

For the convenience of experiment, we automate analysis phase according to an algorithm in [13]. Automation is possible here because AR3, AR4 and AR5 have class labels unlike usual data used in unsupervised models. Clusters in which the rate of fault-prone modules is higher than that of non fault-prone modules are labeled as fault-prone clusters. Similarly, clusters in which the rate of non fault-prone modules is higher than that of fault-prone modules are labeled as non fault-prone clusters.

Table 2 shows the results of Xmeans and EM models. A model with NotReduction is a model using all attributes without reducing dimension. A model with CfsSubsetEval is a model using attributes selected by CfsSubsetEval method. The number of clusters K is determined automatically, not determined by experts. When TER is low but FNR is high, it is not to say the model has a good performance because FNR is also important as much as TER. Thus the model should be evaluated in terms of TER and FNR. As shown in Table 2, all models with NotReduction show higher performance than models with CfsSubsetEval. Therefore we compare other models with models with NotReduction.

<table>
<thead>
<tr>
<th>Table 2. Results of Xmeans and EM Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Xmeans NotReduction</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>CfsSubsetEval</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>EM NotReduction</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>CfsSubsetEval</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Table 3 is a summary of results from earlier studies using Xmeans[6] and QDK[7]. Xmeans[6] and QDK show almost similar prediction performances on the AR3, AR5 except on the AR4. On the AR4, QDK outperforms in TER and Xmeans outperforms in FNR. It is to say that QDK is slightly better than Xmeans because TER has higher precedence than FNR.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TER</td>
<td>FPR</td>
<td>FNR</td>
<td>TER</td>
</tr>
<tr>
<td>AR3</td>
<td>33.33</td>
<td>34.55</td>
<td>25</td>
<td>33.33</td>
</tr>
<tr>
<td>AR4</td>
<td>37.38</td>
<td>44.83</td>
<td>5</td>
<td>12.14</td>
</tr>
<tr>
<td>AR5</td>
<td>13.89</td>
<td>14.29</td>
<td>12.5</td>
<td>13.88</td>
</tr>
</tbody>
</table>

Figure 4 shows a graph comparing two models we constructed and two models of earlier studies, Xmeans, EM, Xmeans[6] and QDK. The Xmeans model uses the same clustering algorithm with Xmeans[6] but these are distinguished by input data and automating analysis phase. Xmeans[6] did not automate analysis phase and used only 6 metrics among 29 metrics (LoC, CC, UOp, UOpnd, TOp, and TOpnd).

In terms of TER, the Xmeans is superior to Xmeans[6], QDK and slightly better than EM on the AR3. On the AR4, all models show similar results except Xmeans[6]. For Xmeans[6], TER is extremely high and FNR is extremely low on the AR4. On the AR5, EM is worse than the others. Therefore the Xmeans model significantly outperforms Xmeans[6], QDK and is better than or equal to EM in TER. As shown the results of FNR, all of the models show similar results on the AR3. On the AR4, QDK is better than Xmeans and EM. EM shows worse results and the others show similar results to each other on the AR5. These experimental results show that the Xmeans model which shows good performance in all parts outperforms Xmeans[6], QDK and EM except in term of FNR on the AR4 dataset.

Figure 4. Comparison of Clustering Results in (a) TER and (b) FNR

5. Conclusions

There is a need for unsupervised fault prediction models which do not require training data. However the unsupervised models using clustering algorithms should select the number of clusters which have influence on the performance of prediction. We built unsupervised models using various clustering algorithms that decide the number of clusters automatically. It means experts do not need to do laborious work changing the number of clusters over
and over. We compared Xmeans, EM model we constructed with Xmeans[6], QDK model of the earlier studies. According to the experimental results, the performance of Xmeans model is better than that of the other models except FNR on the AR4.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0021902).

References


Authors

Mikyeong Park, she received the B.S. degree in information technology from Sungshin Women’s University in 2014. She is currently a M.S. student at Sungshin Women’s University. Her research interests include software engineering and data mining.

Euyeok Hong, he received the Ph.D. degree from Seoul National University in 1999. He joined the faculty of the School of Information Technology at Sungshin Women’s University since 2002, and currently is a Professor. His research interests include software quality and prediction models in software engineering.