Comparison of Outlier Detection Methods in Fault-proneness Models

Shinsuke MATSUMOTO, Yasutaka KAMEI, Akito MONDEN, Ken-ichi MATSUMOTO
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
$\{$*shinsuke-m, yasuta-k, akito-m, matumoto*$\}$@ *is.naist.jp*

Abstract

In this paper, we experimentally evaluated the effect of outlier detection methods to improve the prediction performance of fault-proneness models. Detected outliers were removed from a fit dataset before building a model. In the experiment, we compared three outlier detection methods (Mahalanobis outlier analysis (MOA), local outlier factor method (LOFM) and rule based modeling (RBM)) each applied to three well-known fault-proneness models (linear discriminant analysis (LDA), logistic regression analysis (LRA) and classification tree (CT)). As a result, MOA and RBM improved F1-values of all models (0.04 at minimum, 0.17 at maximum and 0.10 at mean) while improvements by LOFM were relatively small (-0.01 at minimum, 0.04 at maximum and 0.01 at mean).

1 Introduction

Various fault-prone detection models (fault-proneness models) have been used to identify modules that may need rework and/or comprehensive testing [2]. These models are constructed from a fit dataset consisting of module metrics (SLOC, cyclomatic number, etc.) as predictor variables and a module status (faulty or not faulty) as an objective variable.

This paper focuses on the problem of “noisy” modules, e.g., very large and complex but having no fault, which generally exist in the fit dataset [5]. Since such noisy data points often degrade the performance of a constructed model, it is desirable to construct the model after removing the noise from the fit dataset. To identify the noise in a dataset, various outlier detection methods have been proposed in a wide variety of research areas (e.g., knowledge discovery [6]). However, to our knowledge, no study has reported which outlier detection method is the most appropriate for fault-proneness models. This paper experimentally evaluates the effects of three outlier detection methods (MOA, LOFM and RBM) each applied to three well-known fault-proneness models (LDA, LRA and CT). In the experiment, we used a module dataset from NASA IV&V Facility Metrics Data Program (MDP) [7].

2 Outlier Detection Method

2.1 Mahalanobis Outlier Analysis

Mahalanobis outlier analysis (MOA) is a distance-based approach, which uses Mahalanobis distance as “outlying degree” of each data point [4]. Mahalanobis distance is computed on the basis of the variance of data points. It describes the distance between each data point and the center of mass. When one data point is on the center of mass, its Mahalanobis distance is 0, and when one data point is distant from the center of mass, its Mahalanobis distance is more than 0. Therefore, data points far away from the center of mass are considered outliers. We apply MOA to each group of modules (fault-prone (*fp*) and not fault-prone (*nfp*)).

2.2 Local Outlier Factor Method

Local outlier factor (LOF) was also proposed to quantitatively measure the outlying degree of data points [1]. LOF is measured based on the density with k-nearest neighbors. Figure 2.2 shows an example of LOF. Each vertical bar indicates the LOF of a data point that has two metrics (x-axis and y-axis). When a data point (α in the figure) keeps a certain distance from k-nearest neighbors, its LOF becomes 1. On the other hand, when a data point (β in the figure) takes varied distances from k-nearest neighbors, its LOF becomes greater than 1. Therefore, data points isolated from k-nearest neighbors are treated as outliers. As with MOA, we apply LOF method (LOFM) to each group of modules (*fp* and *nfp*).

2.3 Rule Based Modeling

Rule based modeling (RBM) was proposed to improve the performance of fault-proneness models [5]. This ap-

0-7695-2886-4/07 $20.00 © 2007 IEEE
DOI 10.1109/ESEM.2007.34

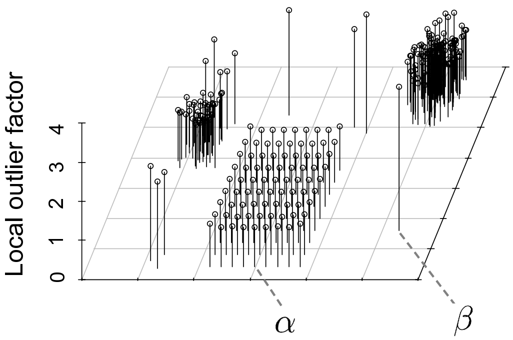

Figure 1. Example of local outlier factor.

proach can be applied to two group classification problems. First, for all j, a threshold to divide j^{th} predictor variable (c_j) into two groups is determined by using Kolmogorov-Smirnov two-sample test based on the percentage of *fp* modules and *nfp* modules. Second, Boolean rules are constructed to divide the high dimensional space of predictor variables into many groups, each assigned to either of two classes (*fp* or *nfp*). For example, assuming two predictor variables (x_1, x_2), rules would be $\{(x_1 \le c_1) \wedge (x_2 \le c_2) \Rightarrow fp\}$, $\{(x_1 \le c_1) \wedge (x_2 > c_2) \Rightarrow nfp\}$, and so on. Finally, modules that do not follow the rules, e.g., a *fp* module that satisfies the antecedent part of a rule $\{(x_1 \le c_1) \wedge (x_2 \le c_2) \Rightarrow nfp\}$, are removed as outliers. As a result, RBM separates completely *fp* modules from *nfp* modules in each separated group.

3 Experiment

3.1 Overview

In the experiment, we applied each of three outlier detection methods to a fit dataset (*fit*). Then we built fault-proneness models by using a *fit* that has no outliers, and evaluated the prediction performance of the models by using test dataset (*test*).

We experimentally determined a threshold of MOA and LOFM using only *fit* in advance. Note that the threshold was determined for each combination of two outlier detection methods (MOA and LOFM) and three fault-proneness models. The value k of the k-nearest neighbor used by LOFM was determined according to Breunig's determination method [1], and was set to 30 to 50.

As evaluation criteria, we used recall, precision and F1-value [3]. F1-value is the harmonic average of precision and recall.

3.2 Dataset

In the experiment, we used a module dataset called KC1, from NASA IV&V Facility Metrics Data Program [7]. It contained 2107 modules, and each module had 20 kinds of source code metrics and the number of faults detected via IV&V. The percentage of *fp* modules was about 15% of all modules. For the construction of fault-proneness models, 20 metrics were used as predictor variables and the existence of fault (no fault or more than one fault) was used as an objective variable. We randomly divided the dataset into two equally sized datasets. One of the dataset is used as *fit* and the other is used as *test*.

3.3 Experimental Procedure

The experimental procedure is as follows.

Step 1. Divide the dataset into *fit* and *test* randomly.

Step 2. Apply an outlier detection method to *fit* and remove outliers. The resultant dataset is *fit'*.

Step 3. Construct a fault-proneness model by using *fit'*.

Step 4. Evaluate the fault-proneness model using *test*.

Step 5. Repeat steps 1 to 4 ten times and evaluate the average prediction performance.

Step 6. To conduct an "unprocessed experiment" that is not applied outlier detection method, repeat steps 1 to 5, omitting step 2.

4 Result and Discussion

Table 1 shows the average prediction performance of each combination. Bold letters in a cell indicate improvements of performance compared with unprocessed experiment. As shown in Table 1, MOA improved F1-values of all models (0.04 at minimum, 0.17 at maximum and 0.10 at mean). Similarly, RBM improved F1-values of all models (0.04 at minimum, 0.16 at maximum and 0.10 at mean). On the other hand, improvements by LOFM were relatively

Table 1. Prediction performance of each combination.

		LDA	LRA	CT
Recall	unprocessed	0.277	0.154	0.241
	MOA	**0.376**	**0.389**	**0.378**
	LOFM	0.276	**0.160**	**0.301**
	RBM	**0.348**	**0.336**	**0.323**
Precision	unprocessed	0.557	0.652	0.483
	MOA	0.452	0.452	0.440
	LOFM	0.515	0.625	0.460
	RBM	0.463	0.406	0.412
F1-value	unprocessed	0.367	0.247	0.317
	MOA	**0.409**	**0.416**	**0.402**
	LOFM	0.356	**0.252**	**0.357**
	RBM	**0.408**	**0.405**	**0.404**

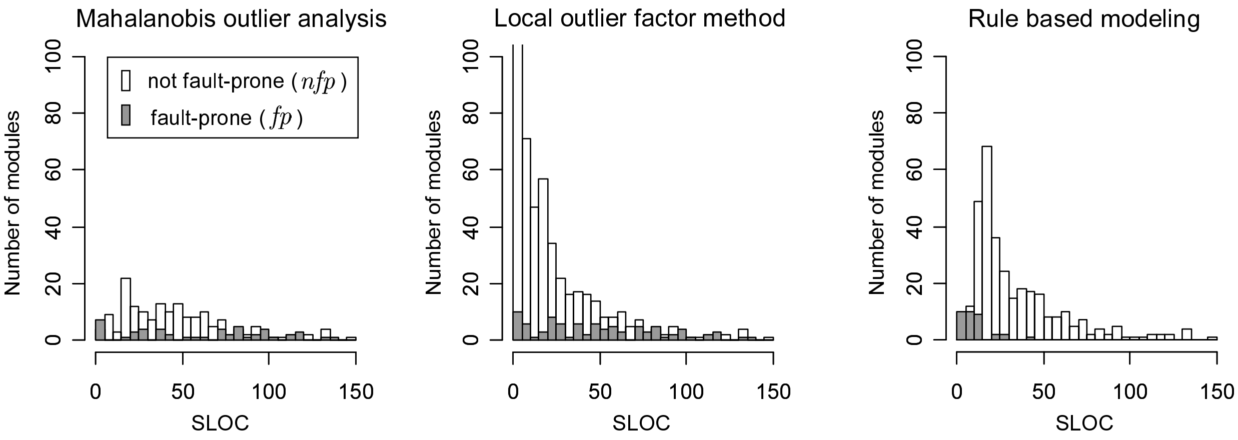

Figure 2. Modules removed by each outlier detection method.

Table 2. Percentage of removed modules of each combination.

	LDA	LRA	CT
MOA	20.3%	37.5%	27.7%
LOFM	55.5%	55.5%	55.6%
RBM	30.3%	30.3%	30.3%

small (-0.01 at minimum, 0.04 at maximum and 0.01 at mean).

Table 2 shows the percentage of removed modules of each combination. As shown in Table 1, LOFM removed too many modules (55.5% on average) while MOA and RBM did not (around 30%).

Figure 2 shows a histogram of the number of modules removed by each outlier detection method (in the case of LDA). The x-axis shows SLOC, which had the strongest correlation with the *fp* or *nfp*. The y-axis shows the number of removed modules. The white bar indicates *nfp* modules and the gray indicates *fp* modules. Since we are focusing on the *nfp* modules, we note that LOFM removed many very small (SLOCs are nearly zero) *nfp* modules, while other methods did not. It is possible that LOFM could not improve the prediction performance because it removed very small modules having no fault as outliers, which are generally not outliers.

5 Conclusion

In this paper, we experimentally evaluated the effect of outlier detection methods to improve the prediction performance of fault-proneness models. The result showed that MOA and RBM improved F1-values of all models (0.04 at minimum, 0.17 at maximum and 0.10 at mean) while improvements by LOFM were relatively small (-0.01 at minimum, 0.04 at maximum and 0.01 at mean).

The limitation of this paper is that we used only one dataset. Our future work will be to use other datasets to increase the validity of the results.

6 Acknowledgements

This work was partially supported by the EASE (Empirical Approach to Software Engineering) project, which is part of the Comprehensive Development of e-Society Foundation Software program of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc. SIGMOD*, pages 93–104, May 2000.

[2] A. R. Gray and S. G. MacDonell. Software metrics data analysis—exploring the relative performance of some commonly used modeling techniques. *Empirical Software Eng.*, 4(4):297–316, Dec 1999.

[3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, Jan 2004.

[4] S. A. Jimenez-Marquez, C. Lacroix, and J. Thibault. Statistical data validation methods for large cheese plant database. *J. Dairy Sci.*, 85(9):2081–2097, Sep 2002.

[5] T. M. Khoshgoftaar, N. Seliya, and K. Gao. Detecting noisy instances with the rule-based classification model. *Intell. Data Anal.*, 9(4):347–364, Jul 2005.

[6] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. VLDB*, pages 392–403, Aug 1998.

[7] NASA/WVU IV&V Facility, Metrics Data Program. http://mdp.ivv.nasa.gov/.