An Investigation of CK Metrics Thresholds

Raed Shatnawi
Computer Science Department
The University of Alabama in Huntsville
Huntsville, AL 35899
rshatnaw@cs.uah.edu

Abstract

There is a dearth of studies that identified thresholds values of Chidamber and Kemerer (CK) metrics. In an empirical study on open-source software, Eclipse project—Version 3.0, we identified the thresholds values for CBO, RFC and WMC at two levels of risks using a quantitative methodology based on the logistic regression curve. These threshold values can be used to identify the most error-prone classes.

1. Introduction

There were enormous empirical studies that validated Object-Oriented (OO) metrics as predictors of software quality [1][4][6]. However, there is still a dearth of studies that identified the threshold values of OO metrics. Benlarbi et al. [3] estimated the threshold values of OO metrics using Ulm's statistical method [9]. Benlarbi et al. found that the estimated threshold values were not significant [3]. Ulm's method assumes that the error probability of a class is flat below the threshold level and increases according to the logistic function above the threshold level, but Bender pointed out that these thresholds should only be considered as valid threshold values if the prediction model's assumption, i.e. a constant risk below the threshold, is plausible [2]. Bender proposed a quantitative method for identifying threshold values from a logistic regression curve. In this paper, we identify the threshold values of CK [5] metrics using Bender methodology [2].

2. CK metrics

In this research we studied the suite of six object-oriented metrics [5]:

- Coupling between Objects (CBO): the number of couplings with other classes.
- Response for Class (RFC): the size of the response set for the class.
- Weighted Methods Complexity (WMC): the sum of the complexity of all methods for a class.
- Depth of Inheritance Hierarchy (DIT): measures the length of the inheritance chain starting from the root.
- Number of Child Classes (NOC): counts the number of children classes.
- Lack of Cohesion of Methods (LCOM): takes each pair of methods in the class and determines the set of fields they each access. If they have disjoint sets of field accesses, increase the count P by one. If they share at least one field access, then increase Q by one. After considering each pair of methods: LCOM = (P > Q) ? (P – Q): 0.

3. Data Collection

We collected and analyzed the error data for Eclipse project, version 3.0, from the change log of the Eclipse project [7]. We separated the errors fixed in Eclipse 3.0 by cross-referencing with the data logged on Bugzilla.

Table 1. Eclipse 3.0 Descriptive Statistics

Metrics	Mean	Std. Dev.	Percentiles		
			25%	50%	75%
CBO	8.31	10.11	2	5	11
RFC	40.02	71.53	7	18	45
WMC	24.86	41.57	5	12	28
DIT	1.59	1.25	1	1	2
NOC	1.00	6.71	0	0	0
LCOM	101.78	652.44	0	4	40

From the descriptive statistics shown in Table 1, we noticed that 50% of classes had low LCOM values (less than four) and 25% of the classes had zero in the LCOM metrics value; Basili et al. [1] and Briand et al. [4] also noted some problems in the original LCOM definition. Therefore, we excluded the LCOM metric. We also

ISSRE 2006 Supplementary Conference Proceedings

noticed that NOC values were zeros in at least 75% of the classes, the highest DIT value was eight, and 75% of classes had at most three levels of inheritance. This low variability in the DIT and NOC metrics agrees with the findings of Chidamber and Kemerer [6], therefore we excluded the DIT, and NOC metrics.

4. The Statistical Models

We used the Univariate Binary Logistic Regression (UBR) to examine whether there was any significant association between a metric and the error-proneness in the classes. We used 0.05 as the P-value to identify the significant metrics [8]. For the metrics that were significant we investigated the possible threshold values using Bender method [2]. The general logistic regression model is as follows:

$$P(X) = \frac{e^{g(x)}}{1 + e^{g(x)}},$$

Where:

$g(x) = \alpha + \beta * X$ is the logit function; X: is a metric; β: is the estimated coefficients from maximizing the log-likelihood; α: is the estimated constant.

P: the probability of a class being faulty.

Bender proposed a method for identifying possible threshold values based on the logistic curve [2]. The method defined as the Value of an Acceptable Risk Level (VARL). The acceptable risk level is given by a suggested probability p_0 (e.g. $p_{01} = 0.05$ or $p_{02} = 0.01$). For classes with metrics values below a threshold VARL, the risk of an error occurrence is lower than p_0. The VARL calculated as follows:

$$VARL = p^{-1}(p_0) = \frac{1}{\beta}\left(\log\left(\frac{p_0}{1 - p_0}\right) - \alpha\right)$$

This equation will be applied in the next section to calculate the threshold values for CK metrics.

5. Threshold Calculations

Table 2 shows the P-values for the univariate logistic regression. We noticed that the CBO, RFC, and WMC metrics were significant predictors of error-proneness (P-value <0.001), therefore we calculated the threshold values for these metrics. We noticed from Table 2 that for different values of p_0 (p_{01}=0.10 and p_{02}=0.075), we got different VARL's for each metric, since p_0 was a factor in calculating the threshold values. However, the two VARL's for each metric can be used at different levels of identifying the problematic parts of software that requires more testing.

Table 2. VARL's for Eclipse 3.0

Metrics	P-value	β	α	VARL1 (p_{01})	VARL2 (p_{02})
CBO	<0.001	0.029	-2.75	24	10
RFC	<0.001	0.003	-2.67	161	55
WMC	<0.001	0.005	-2.66	102	33

6. Conclusion

We identified the thresholds values at two levels, VARL1 and VARL2. Based on software testing resources availability, testers can use the first or the second level to identify the group of classes that are more prone to errors.

7. References

[1] V.L. Basili, L. Briand, and W.L. Melo, "A Validation of Object-Oriented Metrics as Quality Indicators," *IEEE Transactions Software Engineering*, vol. 22, no. 10, pp. 751-761, 1996

[2] R. Bender, "Quantitative Risk Assessment in Epidemiological Studies Investigating Threshold Effects," *Biometrical Journal*, vol. 41, No. 3, pp. 305-319, 1999

[3] S. Benlarbi, K. El Emam, N. Goel, S. Rai, "Thresholds for Object-Oriented Measures," In *11th International Symposium on Software Reliability Engineering (ISSRE'00)*, p. 24, 2000.

[4] L.C. Briand, J. Wust, J.W. Daly, and D.V. Porter, "Exploring the Relationship between Design Measures and Software Quality in Object Oriented Systems," *Journal Systems and Software*, vol. 51, no. 3, pp. 245-273, 2000.

[5] S.R. Chidamber and C.F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476-493, 1994.

[6] S.R. Chidamber, D.P. Darcy, and C.F. Kemerer, "Managerial Use of Metrics for Object Oriented Software: An Exploratory Analysis," *IEEE Transactions on Software Engineering*, vol. 24, no. 8, pp. 629-639, 1998.

[7] The homepage, http://www.eclipse.org, 2004.

[8] Hosmer, D., and Lemeshow, S., Applied Logistic Regression, Wiley Series in Probability and Statistics, second edition, 2000.

[9] K. Ulm, "A Statistical Method for Assessing a Threshold in Epidemiological Studies", In *Statistics in Medicine,* vol. 10, pp341 -349, 1991.

ISSRE 2006 Supplementary Conference Proceedings