Data Analyses on Defect Data and Experiments with Defect Prediction Methods using Topic Metrics

Sarat Kiran Andhavarapu, *A01960723*

***Abstract*—In spite of diligent planning, good documentation and proper process control during software development, appearance of defects are inevitable. These software defects may lead to degradation of the quality and even sometimes extremely serious problems. it is important to make conscious efforts to control and minimize these defects by using techniques to allow in-process quality monitoring and control. In this paper, I compared Weibull, Rayleigh and Gamma models to check the goodness of the fit for the defect curses in the taken datasets and prediction accuracy with defect curves of prior release and using defect curves of other projects. [2]**

***Keywords*—Defect data analysis, Software metrics, Evaluation, Machine Learning Algorithms, Weibull model, Rayleigh model, Gamma model, defect curves and projection.**

I. Introduction

Due to increasing extent and complexity of software systems, it is common to have large teams, communities of developers working on the same project at the same time this may cause bugs. One example that burned up a 327.6 million project in minutes is in 1998, when the Mars Climate Orbiter built by NASA's Jet Propulsion Laboratory approached the Red Planet at the wrong angle. The biggest problem was that different parts of the engineering team were using different units of measurement. One group working on the thrusters measured in English units of pounds-force seconds; the others used metric Newton-seconds and this is not been checked, Even these small bugs can collapse the whole system.

Quality assurance activities, such as tests or code reviews, are an expensive, but vital part of the software development process as we can see from the above mentioned example. Any support that makes this phase more effective may thus improve software quality or reduce development costs. [1]

Defect-occurrence projection is necessary for the development of methods to mitigate the risks of software defect occurrences. The occurrence of defects is not only problem with the users, but also cause problems in maintenance planning for software makers. The costly consequences of defect occurrences have increased interest in insuring software consumers against the associated risks. [3]

In this experiment we aim to answer three questions:.

Question 1: What is the correlation of topic metrics to BUG? You should answer this question by plotting as boxplots the correlation of BUG and topic metrics for the number of topics of 10, 20, 50, and 100.

Question 2: Do topic metrics provide better and additional explanatory power and predictive power to base metrics? You could answer this question by plotting:
- Explanatory power and predictive power of 3 base metrics as baseline.
- Explanatory power and predictive power when only topic metrics are used. You will plot for K = 5, 10, ... 100 topics. For each K, you vary P (number of selected pricipal components) and choose what provides the best explanatory/predictive power.
- Explanatory power and predictive power when both topic metrics and base metrics are used. You will also plot for K = 5, 10, ... 100 topics, each with the best P.

So we will look at each question, discuss the method we tried to solve the problem and finally results that we obtained.

II. Datasets

A. What I understand from the datasets ?

This project contains five datasets namely Eclipse, Equinox, lucene, mylyn, pde. In turn these 5 datasets contain multiple "csv" files that contains each of following items
The explanatory power of a prediction model is measured with AIC score (the lower the better). The predictive power of a prediction model is measured with Spearman correlation (the higher the better).

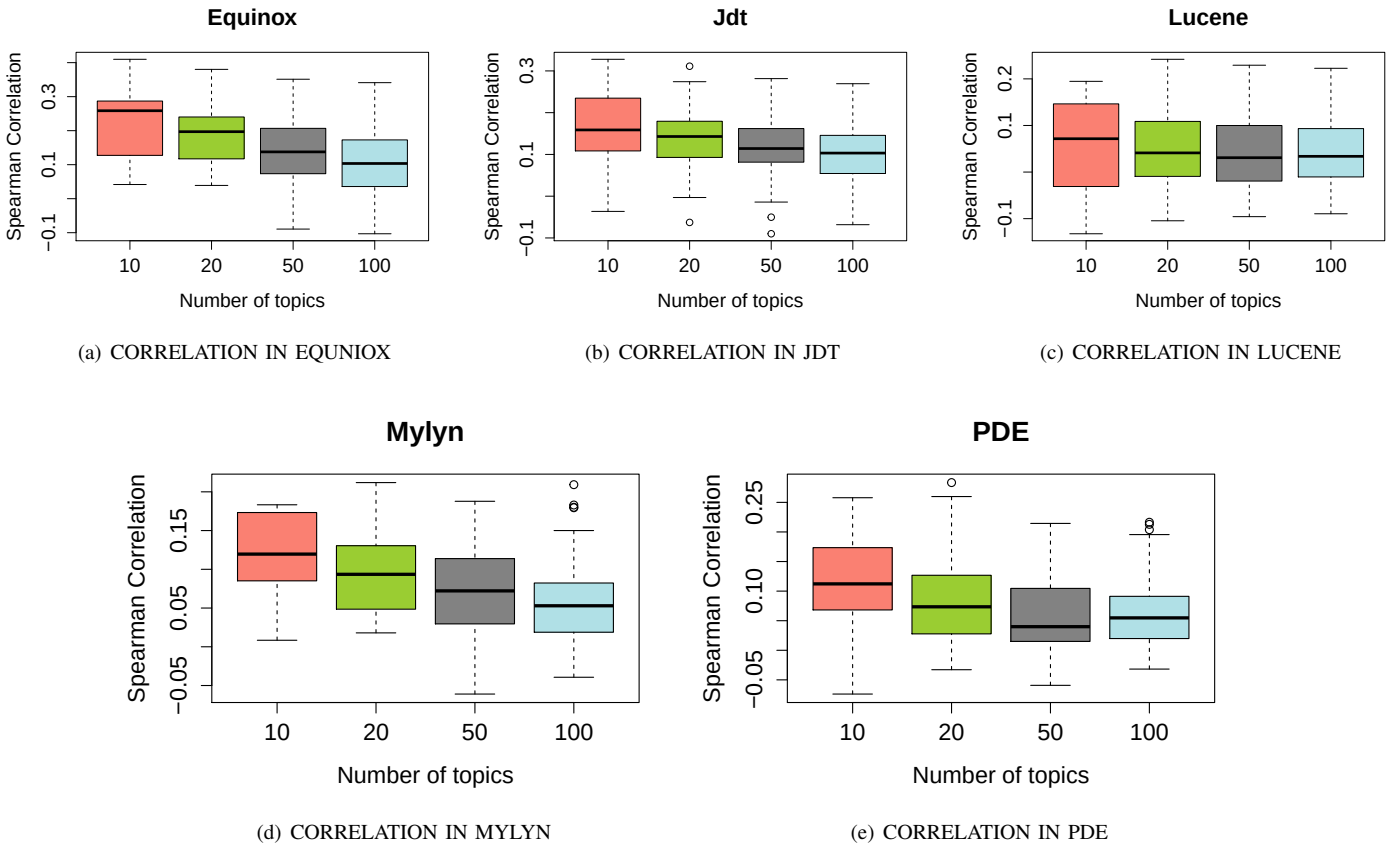
1) LOC: number of lines of code (measuring the code complexity)
2) BF: number of prior bug fixes (measuring the defect history)
3) HCM: the entropy of code changes (measuring the complexity of code changes)
4) V1-VK: the log transformation of the number of words assigned for each topic (from 1 to K). For example, topic5log.csv will have 5 metrics V1-V5, each for a topic.

we should include the 3 base metrics into the topics and perform Principal Component analysis if we want to find PCA for topics + base metrics , BUG is the actual number of post-release bugs, which is predicted not used as a predictor.

III. Analysis

A. Question 1: Correlation of topic metrics to BUG

To answer the first question we took topic metrics for the number of topics of 10, 20, 50, and 100 and found the correlation between the topics and the "bugs", the results for



(a) CORRELATION IN EQUNIOX

(b) CORRELATION IN JDT

(c) CORRELATION IN LUCENE

(d) CORRELATION IN MYLYN

(e) CORRELATION IN PDE

Fig. 1. correlation of topics with bugs for different projects

these are plotted in boxplots in figure 1. Things I observed from the graph

1) As the number of topics increase the boxplot area decreases, which is very intresting as topics will increase and the area is consolidated in small area. We observe this may be beacuase as number of topics increase there are more number of similar topics.
2) Only in few files (PDE,Mylyn, JDT) there are dots that are seen way out of the areas we plotted, we ignore the same.

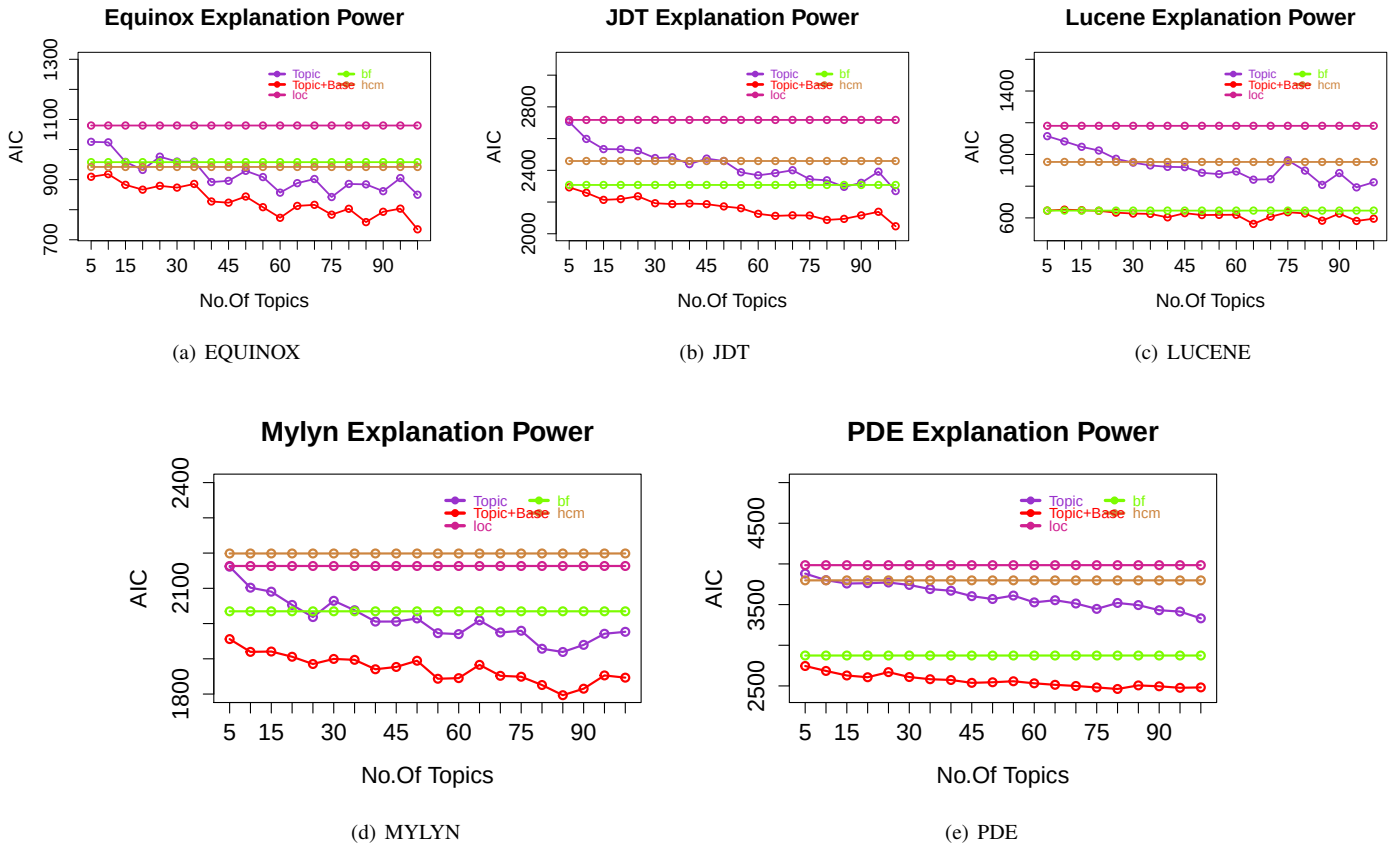
B. Question 2: Topics that provide the best Predictive and Explanative powers

Few terms we need to know before we perform analysis:

1) Principal component analysis (PCA) - is a statistical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables.
2) AIC score - The Akaike information criterion (AIC) is a measure of the relative quality of a statistical model, for a given set of dataset.AIC deals with the trade-off between the goodness of fit of the model and the complexity of the model. It is founded on information entropy: it offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. AIC does not provide a test of a model in the sense of testing a null hypothesis; i.e. AIC can tell nothing about the quality of the model in an absolute sense. If all the candidate models fit poorly, AIC will not give any warning of that. In the general case, the AIC is:

$$AIC = 2k - 2ln(L)$$

where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood



(a) EQUINOX (b) JDT (c) LUCENE

(d) MYLYN (e) PDE

Fig. 2. Explanatory power of different projects with AIC scores for each. (lower the better)

function for the estimated model.

3) Coefficient of determination (R^2) : the coefficient of determination, denoted R2 and pronounced R squared, indicates how well data points fit a statistical model – sometimes simply a line or curve. It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, as the proportion of total variation of outcomes explained by the model.

Figures 2 and Figures 3 summarizes the Explanatory power and predictive power for the topics from K = 5, 10 till 100 with a increment of 5 for each level. Here we take five things to plot the graphs - Topics metrics, Topic + Base metrics, LOC, HCM, BF metrics correlated with bugs. Few observations we can make from the graphs we get are - AIC score is used to evaluate the explanatory power and I find that topic + base metrics have the highest in all the project files. We can conclude that the topic + base metrics will give us the best explanatory power.

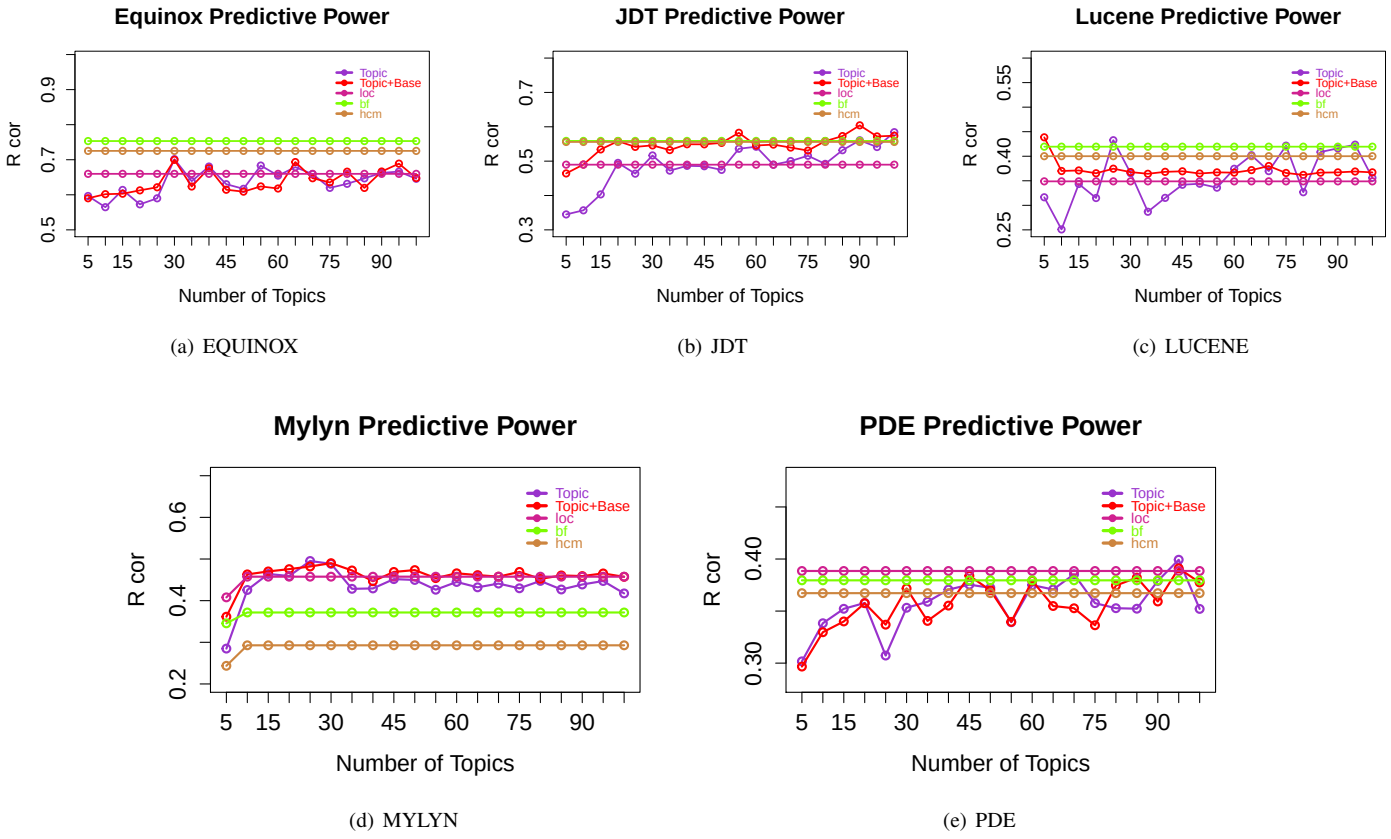
Figures 3 summarizes the predictive power using LM (Linear Regression Model) and. To evaluate the prediction power, we use the cross validation, by selecting randomly 80% data for training and 20% for predicting. we compute mean Spearman's correlation between the predicted PCA scores and bug number of post-release defects and the actual number. We repeat the above steps thirty times and find the mean Spearman ranked correlation.

IV. Conclusion

TABLE I. Explanatory power Best values of AIC, P, K - topics, topics + base metrics for projects

Data set	Combined - Topics + base			Topics		
	AIC	Best P	Best K	AIC	Best P	Best K
Equinox	734.79	98	100	842.13	35	75
JDT	2047.23	93	100	2269.3	90	100
Lucene	562.09	54	65	792.54	84	95
Mylyn	1796.79	88	85	1919.41	85	85
PDE	2462.598	81	80	3330.93	95	100

In this Project, we would use topics and base metrics to get the explanatory power and use LM (linear regression model) to predict the bugs and to get predictor power of each metrics. First we used Principal Component Analysis to get correlation of metrics and observed that as topics increased, they tend to get close. As explanatory power and predictive power are not



(a) EQUINOX (b) JDT (c) LUCENE

(d) MYLYN (e) PDE

Fig. 3. Predictive power of different projects with rcor values for each. (higher the better)

comparable, we plot them in different tables as shown below. This table gives Best P (principal components) and Best K for topics and topics + base metrics for the explanatory powers.

TABLE II. PREDICTIVE POWER BEST VALUES OF RCOR, P, K - TOPICS, TOPICS + BASE METRICS FOR PROJECTS

Data set	Combined - Topics + base			Topics		
	R COR	Best P	Best K	R COR	Best P	Best K
Equinox	0.699	29	30	0.701	25	30
JDT	0.604	93	90	0.584	95	100
Lucene	0.438	2	5	0.432	22	25
Mylyn	0.490	6	30	0.495	15	25
PDE	0.391	36	95	0.399	11	95

This table gives Best P (principal components) and Best K for topics and topics + base metrics for the predictive powers. When we compare the 5 metrics we can see that topic + base metrics performs the best in the explanatory power and the predictor we used find similar results with high rcor values for topic + base metrics for most projects.

V. FUTURE WORK

This project can be extended to attain accurate results. Taking a much larger training set, using other models with accurate parameters can be helpful. The LM (Linear regression model) can be replaced with better models to get good predictor results as LM model did not perform accurately for two projects.

ACKNOWLEDGMENT

I would Like to thank Dr Tung for his excellent guidance for this project.

REFERENCES

[1] Anh Tuan Nguyen, Tung Thanh Nguyen, J. Al-Kofahi, Hung Viet Nguyen, and T.N. Nguyen. A topic-based approach for narrowing the search space of buggy files from a bug report. In *Automated Software Engineering (ASE), 2011 26th IEEE/ACM International Conference on*, pages 263–272, Nov 2011. doi: 10.1109/ASE.2011.6100062.

[2] Tung Thanh Nguyen, Tien N. Nguyen, and Tu Minh Phuong. Topic-based defect prediction (nier track). In *Proceedings of the 33rd International Conference on Software Engineering*. ACM, 2011. ISBN 978-1-4503-0445-0. doi: 10.1145/1985793.1985950.

[3] Qinbao Song, Martin Shepperd, Michelle Cartwright, and Carolyn Mair. Software defect association mining and defect correction effort prediction. *IEEE Trans. Softw. Eng.*, 32(2). ISSN 0098-5589. doi: 10.1109/TSE.2006.1599417. URL http://dx.doi.org/10.1109/TSE.2006.1599417.