Defect Data Analysis and Experiments for Defect Projection

Prudhvi Ratna Badri Satya
A02057243
Computer Science Department
Utah State University
Logan, Utah.

***Abstract*—Software development is a complex task which requires good understanding and sound knowledge on the software system. Research Analysts have proposed different measurements in view of measurable parts of the source code entities such as techniques, classes, documents, or modules, and the social structure of a product that extend with an end goal to clarify the connections between software development and software defects. Notwithstanding, these measurements to a great extent disregard the real usefulness, i.e., the calculated concerns, of a software system framework, which are the principle technical concepts that mirror the business rationale or area of the system. The success of a software is always on par with the number of bugs that are to be found and rectified, in order to enhance the quality of the software product. The maintainers of the software system play a very crucial role in the maintaining the bug reports which contains details of a particular software failure, and detailed description on how these failures have been regenerated in the system. Defect projection is necessary to reduce the risks of the software defects that may lead to the failure of the software product developed. In this paper, I conducted the extensive data analysis and conducted experiments for the defect projection on the four publicly available defect datasets of software products namely Eclipse, OpenBSD, JetSpeed2 and Tomcat.**

***Keywords*—Defect Projection, Jensen-Shannon Divergence, Weibull, Rayleigh, and Gamma models.**

I. Introduction

Software analytics speaks to the base segment of the software analysis that generally aims at generating findings, conclusions, and evaluations about software systems and their implementation, composition, behavior, and evolution. Software analytics uses and consolidates methodologies and strategies from statistics, prediction analysis, data mining, and scientific visualization.

Software analytics is used to explore the data in order to predict the defects in the software system which is used to improve the quality of the software. The process of finding the bugs in the software system is one of the cost and time consuming tasks. Reliability models are developed to predict the defects and the failure rates in the software. Large complex multivariate statistical models have been proposed to find a single complexity metric that will account for defects. For the purpose of this project, we have performed study on the publicly available defect data sets. We are interested the defect-occurrence pattern, which is the rate of defect occurrence as a function of time over the lifetime of a release. We define the lifetime of a release as the duration of time between when a release becomes generally available and when there are no defect occurrences reported to the software development organization for three consecutive time intervals.

The analysis has been carried out on the defect data sets of four different projects named Eclipse, OpenBSD, JetSpeed2 and Tomcat. The In-depth and relevant information cannot be obtained easily by scrutinizing the raw data as such without the sound support of the software analytic technologies. The information obtained by using software analytics is the most relevant information that conveys the proper understanding or knowledge towards performing the given target assignment.

In this paper, I focus at answering the following research questions by performing data analysis and machine learning:

1.How are defect curves of the same product similar to each other?

2. What are the modeling quality (i.e. the goodness of fit) of the defect curves using Weibull, Rayleigh, and Gamma models? What is the best model?

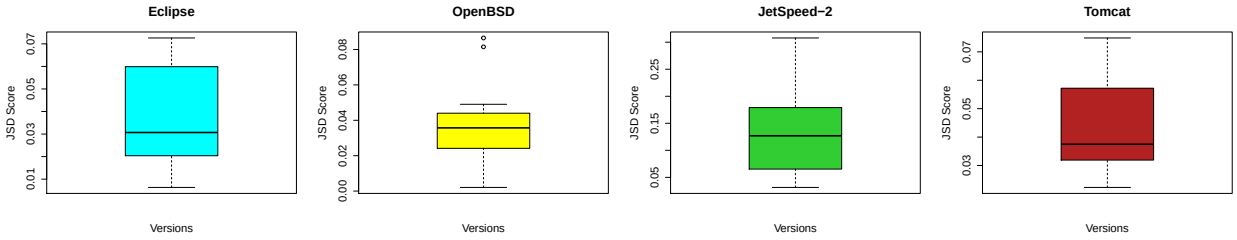
3. What is the prediction accuracy of the defect volume (for the whole reporting time) and defect count (for each time unit)?

The organization of the paper is as follows: Section 2 explains about the methodology for the defect Projection. Datasets and how defect curves of the same product similar to each other is discussed in Section 2A. Section 2B discusses about the modeling quality (i.e. the goodness of fit) of the defect curves using Weibull, Rayleigh, and Gamma models. Section 2C discusses about the prediction accuracy of the defect volume (for the whole reporting time) and defect count (for each time unit). Results and Discussion are discussed in Section 3. Finally, Section 4 contains the conclusion.

II. Methodology

A. Datasets and Analyzing similarity of defect curves for the software Products

The datasets that have been used in this paper are the publicly available projects from the Apache and Eclipse foundations. Each of the projects consists of the various versions i.e. Eclipse (E1-E6), OpenBSD (B1-B7), JetSpeed2 (J1-J4) and Tomcat (T1-T4) and each



(a) Eclipse (b) OpenBSD (c) JetSpeed2 (d) Tomcat

Fig. 1: Boxplots of Jensen-Shannon Divergence Score of all defect curves for the software products

version file contains a defect curve, i.e. a sequence of numbers of post-release defects per quarter reported for the corresponding version (release) of those software products. The defect data for the project JetSpeed2 has been extracted manually from the raw data present in the Apache foundation.
In order to answer the research question *Q1.How are defect curves of the same product similar to each other?*, the Jensen-Shannon Divergence (JSD) of all pairs of normalized (i.e. having sum of 1) defect curves of each product is computed. The Jensen–Shannon divergence is a popular method of measuring the similarity between two probability distributions. Then boxplots are plotted as shown in Fig.[1] for the JSD scores computed for Eclipse, OpenBSD, JetSpeed2 and Tomcat projects.

B. Analyzing models that are best fit for the defect curves.

The research question *Q2.What are the modeling quality (i.e. the goodness of fit) of the defect curves using Weibull, Rayleigh, and Gamma models? What is the best model?*, is answered by computing the fitness of the curves using the Weibull, Rayleigh and Gamma models.
Weibull Distribution model: The probability density function of a Weibull random variable is:

$$f(x;\lambda,k) = \begin{cases} \left(\frac{k}{\lambda}\right)\left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^{k}} & x \geq 0, \\ 0 & x < 0, \end{cases}$$

where k > 0 is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution.
Rayleigh Distribution model: A Rayleigh distribution is often observed when the overall magnitude of a vector is related to its directional components. The probability density function of the Rayleigh distribution is

$$f(x;\sigma) = \left(\frac{x}{\sigma^2}\right) e^{\frac{-x^2}{(2\sigma^2)}} \geq 0,$$

where σ is the scale parameter of the distribution.
Gamma Distribution model: The gamma distribution is a two-parameter family of continuous probability distributions.The probability density function using the shape-scale parametrization is:

$$f(x;k,\theta) = \frac{x^{k-1}e^{-\frac{x}{\theta}}}{\theta^{k}\Gamma(k)} \quad for \quad x > 0 \quad and \quad k,\theta > 0$$

Here $\Gamma(k)$ is the gamma function evaluated at k.
Then R^2 and AIC scores are computed in order to analyze the quality of the fitted models.The Akaike information criterion (AIC) is a measure of the relative quality of a statistical model, for a given set of data. As such, AIC provides a means for model selection. AIC deals with the trade-off between the goodness of fit of the model and the complexity of the model. In the general case, the AIC is:

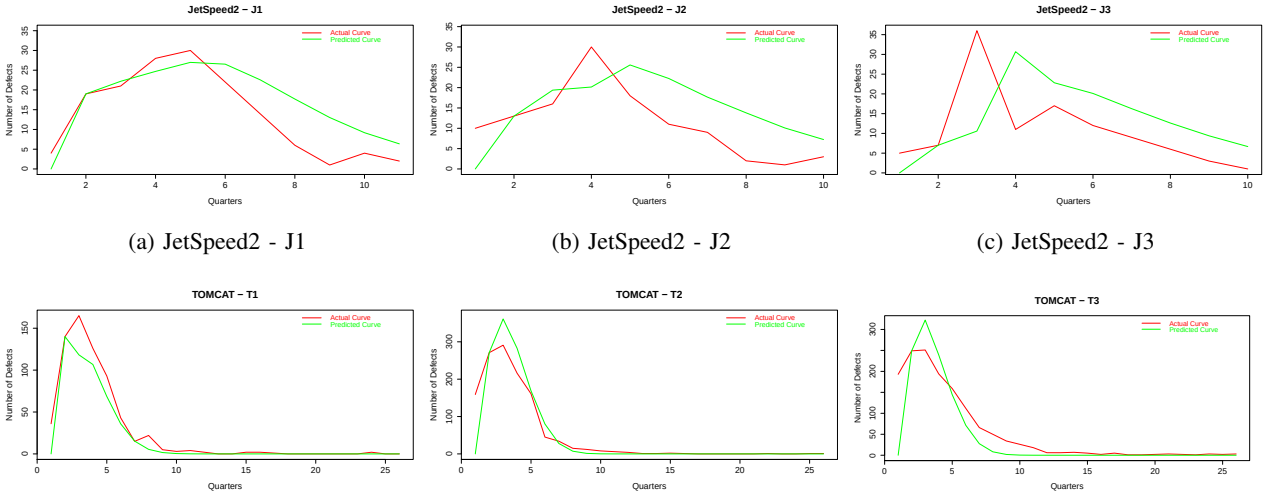
$$AIC = 2k - 2ln(L)$$

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

R^2 Score 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.It is computed as

$$R^2 = 1 - var(y' - y)/var(y)$$

where y is the actual defect counts and y' is the corresponding fitted values.
We used Non-Linear Least Squares method to find out the best fit of the defect curve to the distribution. The fitness is given by R^2 and AIC values. R^2 is the similarity between the two curves. So higher the value of R^2 higher the fit and lower the AIC values highest is the fit. In order to determine the best model among Weibull, Rayleigh, and Gamma, paired t-test is computed on the obtained R^2 and AIC scores.T-Tests can be used to determine if two sets of data are significantly different from each other, and is applied when the test statistic would follow a normal distribution.



(a) JetSpeed2 - J1 (b) JetSpeed2 - J2 (c) JetSpeed2 - J3

(d) TOMCAT - T1 (e) TOMCAT - T2 (f) TOMCAT - T3

Fig. 2: Actual and Predicted Defect Curves for JetSpeed-2 and Tomcat Projects

C. *Predicting the accuracy of defect curves*

To answer the research question *Q3. Prediction accuracy of the defect volume (for the whole reporting time) and defect count (for each time unit)* we need to first determine the model. If the software product is an operating system, we use weibull model otherwise we use rayleigh model. In order to determine the parameters of the model for the initial release we need to predict using the generic prediction model. If it is not the first release we can use the shape and scale values of the previous version to predict the defect curve.Regression models like Linear Regression is used to make the estimates. If it is the latest versions then the averages of the shape and scale of previous versions are taken in to account for prediction and prediction accuracy is determined. To predict the future versions of the JetSpeed2 project we would use the shape and scale parameters from other projects.Then defect volume N and defect count N(t) is predicted sequentially over time.The prediction accuracy of these results is determined by the relative absolute errors (RAE) over time. The RAE is also plotted in order to draw conclusions.In the next section the various results are analysed and discussed.

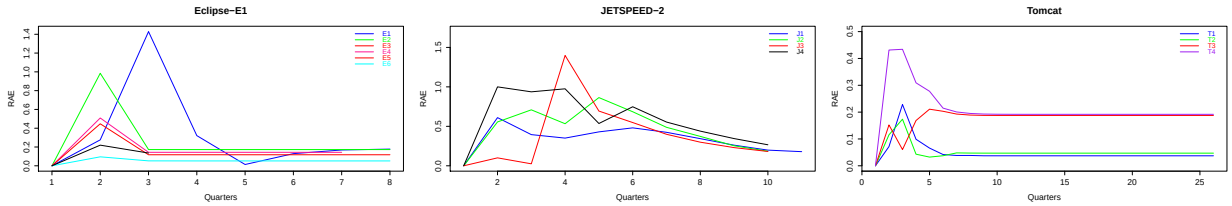
III. Results and Discussion

Defect Curves Similarity : The box plots as shown in Fig.[1] clearly shows that the lower the JSD scores higher the similarity, therefore from the Fig.[1](a) Eclipse project has the lowest JSD Score when compared to the other projects, which clearly states that the defect curves of various versions in the Eclipse project are more similar to each other.

Model that best fits to the Defect curve : From the results present in Table-I which represents the AIC and R^2 scores of the various models. The model that has the lower AIC score and the higher R^2 Score clearly depicts the model that best fits the curve. In order to know which model best fit the curve,. From Table-I, we can say that among the various distribution models to predict the best fit to the curve, Gamma model has the lower AIC scores and higher R^2 Scores when compared to other models for the projects.

Table II represents the p-values obtained from the paired T-Tests conducted among the models for the various projects that are conisdered in this paper. The Paired T-Tests values are computed for AIC and R^2 values among Weibull-Rayleigh, Rayleigh-Gamma and Weibull-Gamma models for the projects. The P-values obtained are less than 5% which represents the obtained results are valid.

prediction accuracy of the Defect Curves: The Fig.[2] shows the plots of the actual and the predicted curves of three versions in the JetSpeed-2 and Tomcat Projects. The linear regression model is applied and the estimates that are choosen are Software type, Cycle and the Cost of the License. We predicted the shape and scale parameters of the all the software projects namely Eclipse, OpenBSD, Jetspeed2 and Tomcat. The curve is plotted based on the results that are obtained by computing the defect count over time. The accuracy of the defect volume is measured by computing the Relative absolute error over time. The Fig.[3] shows the RAE plots of the Eclipse, JetSpeed2 and Tomcat projects.From the Fig.[3], we can say that when the number of quarters increases the prediction accuracy increases as the RAE value decreases.The accuracy become linear after some quarters in all the 3 versions,Thus we can say that using the other products as the training data, we can predict the defect curves of the future versions of products. The final prediction accuracy is measured for the prediction and actual defect curve by computing the R^2 values. The Table-III shows the R^2 values of the predicted defect curves. The R^2 values are computed for all the versions of the projects that are mentioned in the paper.



(a) Eclipse (b) JetSpeed2 (c) Tomcat

Fig. 3: RAE Curves for Eclipse, JetSpeed-2 and Tomcat Projects

Versions		AIC			R^2	
	Weibull	Rayleigh	Gamma	Weibull	Rayleigh	Gamma
E1	126.81	72.30	**69.70**	-0.0322	0.9993	**0.9995**
E2	96.63	54.14	**53.93**	0.02583	0.9953	0.9947
E3	72.84	47.96	**49.05**	0.08493	0.9696	0.9620
E4	85.82	55.42	**55.85**	0.18319	0.9787	0.9774
E5	99.37	63.63	**62.37**	0.01345	0.9896	**0.9898**
E6	43.38	13.29	**19.24**	-0.2748	0.9999	0.9995
J1	98.39	66.23	**60.57**	-0.1769	0.8749	**0.9213**
J2	85.71	61.52	**61.88**	-0.2876	0.7824	0.7728
J3	87.24	72.26	**70.76**	-0.1858	0.5135	**0.5804**
J4	87.44	79.07	**65.00**	-0.1327	0.3089	**0.8281**
B1	79.86	68.56	**53.68**	0.5164	0.8573	**0.9770**
B2	66.25	57.28	**38.35**	0.8871	0.9510	**0.9954**
B3	82.43	73.31	**53.32**	0.5573	0.8322	**0.9856**
B4	51.68	45.17	**39.65**	0.9403	0.9678	**0.9874**
B5	43.74	41.63	**27.49**	0.7447	0.7998	**0.9938**
B6	61.29	47.90	**49.59**	0.5130	0.9323	0.9099
B7	41.18	38.71	**23.05**	0.6573	0.7432	**0.9947**
T1	278.56	201.27	**151.69**	0.2247	0.9525	**0.9929**
T2	306.78	188.53	**197.78**	0.3233	0.9916	0.9879
T3	304.54	223.39	**153.18**	0.3448	0.9689	**0.9975**
T4	298.01	212.59	**211.33**	0.0566	0.9546	0.9542

TABLE I: Measuring Model Quality using AIC and R^2 Scores

		P-Values of Paired T-test				
	AIC			R2		
	Wei-Ray	Ray-Gam	Wei-Gam	Wei-Ray	Ray-Gam	Wei-Gam
Eclipse	0.0004	0.657	0.001	2.40E-05	0.2651135	2.58E-05
JetSpeed2	0.0314	0.202	0.011	0.021	0.2923	0.001
OpenBSD	0.003	0.0054	0.0003	0.0241	0.0274	0.0031
Tomcat	0.0023	0.2384	0.0032	0.0012	0.2311	0.0009

TABLE II: P-Values from Paired T-Test on AIC and R^2 Scores

R^2 VALUES							
	File1	File-2	File-3	File-4	File-5	File-6	File-7
OpenBSD	0.5057	0.3267	0.9073	-0.0099	0.4792	0.3908	0.6794
Eclipse	-0.7526	0.0381	0.0836	0.1843	0.0225	-0.2989	
JetSpeed2	0.7145	0.1623	-0.3714	-0.2022			
Tomcat	0.9361	0.8201	0.7238	0.8655			

TABLE III: R^2 Values of the Predicted Defect Curves.

IV. Conclusion

This paper discusses about the Defect Projection model which uses the defect curves of various datasets such as Eclipse, OpenBSD, JetSpeed2 and Tomcat . The similarity between the curves of various versions in a product is determined. The curves are then fit into a statistical distribution (like Rayleigh, Weibull, Gamma models) and the best fit to the curve is determined.In this project for the given scale and shape factorrs of the projects Gamma distribution was determined to be the best fit to the curve. Then using the shape and scale parameters of the best fit, we can predict the defect curves of the future versions using the previous version trends. The projection of defects done using this methodolgy gave not so good accuracy results for the projects considered in this paper.

Acknowledgements

I sincerely thank Dr.Tung Nguyen for his valuable support through out this project. I also acknowledge other students in this class for their extended help in this project.

References

[1] Tse-Hsun Chen, Stephen W. Thomas, Meiyappan Nagappan, Ahmed E. Hassan: Explaining Software Defects Using Topic Models.

[2] Tim Klinger P. Santhanam Tung Thanh Nguyen, Evelyn Duesterwald and Tien N. Nguyen: Characterizing defect trends in software support.

[3] T Patrick Knab,Martin Pinzger, Abraham Bernstein: Predicting defect densities in source code files with decision tree learners.