An Application of Zero-Inflated Poisson Regression for Software Fault Prediction

Taghi M. Khoshgoftaar*
Kehan Gao
Florida Atlantic University
Boca Raton, Florida USA

Robert M. Szabo
IBM Corporation
Fort Lauderdale, Florida USA

Abstract

Poisson regression model is widely used in software quality modeling. When the response variable of a data set includes a large number of zeros, Poisson regression model will underestimate the probability of zeros. A zero-inflated model changes the mean structure of the pure Poisson model. The predictive quality is therefore improved. In this paper, we examine a full-scale industrial software system and develop two models, Poisson regression and zero-inflated Poisson regression. To our knowledge, this is the first study that introduces the zero-inflated Poisson regression model in software reliability. Comparing the predictive qualities of the two competing models, we conclude that for this system, the zero-inflated Poisson regression model is more appropriate in theory and practice.

Keywords*: software quality modeling, Poisson regression model, zero-inflated Poisson regression model, nested models, Vuong hypothesis test, program module*

1. Introduction

The objective of software quality modeling is to predict the quality of modules early in software development, and then direct an effective treatment to improve reliability of the modules that are predicted to need improvement. Software quality modeling may include prediction modeling and classification modeling. Prediction modeling may involve estimating the number of faults in software modules. Classification modeling may include identifying software modules as fault-prone or not fault-prone. Some software quality modeling techniques are suited for prediction or classification, but not both. For example: logistic regression method is only used for classification [11] and multiple linear regression method is only used for prediction. Whereas, other approaches can be applied to both, such as Case Based Reasoning (CBR) [12] and count modeling methods. In this paper, we focus on examining the count models for prediction. Count modeling methods for classification purposes will be presented in our future work.

Count models where the response variable (dependent variable) values are non-negative integers are common in many areas, such as economics, social science and software reliability engineering [3, 15, 17]. Various techniques have been proposed by many researchers for count modeling, especially in the economics field. These techniques include Poisson regression model [2, 3, 10, 17], mixed-Poisson model [2, 3, 7, 9], truncated model [3, 8], hurdle model [2, 3, 8, 16] and zero-inflated model [9, 14, 16]. In software quality count modeling, research studies are mainly focused on the basic method – Poisson Regression Model (PRM) [17], while other techniques are rarely involved. In this paper, empirical studies were performed using the Poisson regression modeling and the Zero-Inflated Poisson (ZIP) regression modeling methods to a full-scale industrial software system. It is concluded that as compared to the PRM, the ZIP regression model is more appropriate when the response variable of the data set contains a large proportion of zeros. Applying the ZIP regression technique to software quality modeling is one of the contributions of this paper. To our knowledge, this is the first study that introduces the ZIP regression model in software reliability.

The Poisson regression model is the most popular method in count modeling. It requires *equidispersion*, i.e., the expected value of the response variable equals its variance. When a data set has a Poisson distribu-

*Readers may contact the authors through Taghi M. Khoshgoftaar, Empirical Software Engineering Laboratory, Dept. of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA. Phone: (561)297-3994, Fax: (561)297-2800, Email: taghi@cse.fau.edu, URL: www.cse.fau.edu/esel/.

1071-9458/01 $17.00 © 2001 IEEE

tion, PRM is suitable to fit the data. However, we often have situations where zeros account for a large proportion of response variable values in a data set. Pure PRM fails to make an accurate prediction in these situations. In 1992, Lambert first introduced the ZIP regression model [14] which has been proven to be an effective way to solve the problem. In a ZIP regression model, we divide the data into two groups. One group consists of observations where the response variable values are zeros. The other group follows a standard Poisson distribution. Consequently, the mean structure is changed to the weighted combination of mean from each group.

In practice, we usually do not know what distribution the data set follows. We need to try different methods, compare them and finally choose an appropriate one. To evaluate the quality of the models, we often split the data set into two subsets, i.e., a fit data set and a test data set. The *fit* data set is used to build the model and the *test* data set is used to evaluate the model. Generally, there are two ways, theoretical and practical, to compare the qualities of models. The theoretical approach refers to hypothesis testing. This method compares models based on the fit data set. Since the PRM and ZIP regression model are *non-nested* [10], the general methods such as likelihood ratio test, Wald test or Langrange multiplier test are not suitable. In this paper, we applied a special test proposed by Vuong [18]. In comparison to the theoretical approach, the practical approach calculates quality of the models based on performance statistics such as average absolute error and average relative error [17] of the test data set.

In this paper, we investigated software measures collected from a commercial software system and applied the PRM and the ZIP regression model to it. Comparing the predictive qualities of the two competing models, we concluded that for this system, the ZIP model is more appropriate than the PRM model both in theory and practice.

The remainder of this paper is organized as follows: In section 2 and 3, we present the Poisson regression modeling and ZIP regression modeling methods respectively. Hypothesis testing is discussed in section 4. In section 5, we examine the case study. Finally, we present our conclusions.

2. Poisson Regression Model

The Poisson regression model is derived from the Poisson distribution by allowing the intensity parameter μ to be a function of covariates (sometimes also called regressors or independent variables).

2.1 Poisson Distribution

Let y be a random variable with Poisson distribution having parameter μ, that is,

$$\Pr(y|\mu) = \frac{e^{-\mu}\mu^{y}}{y!} \qquad \text{for} \quad y = 0, 1, 2 \cdots . \qquad (1)$$

The Poisson distribution has the property that the expected value of y equals the variance of y, which is known as *equidispersion.*

$$\mathrm{E}(y) = \mathrm{Var}(y) = \mu , \qquad (2)$$

where $\mathrm{E}(\cdot)$ and $\mathrm{Var}(\cdot)$ represent the expected value and the variance respectively. The Poisson distribution can be derived from a stochastic process known as Poisson process [4].

2.2 - Building Poisson Regression Model

The Poisson regression model assumes that the response variable is a count and has a Poisson distribution with mean μ, which is dependent on the covariates. Let $(y_i,\mathbf{x}_i)$ be an observation in the data set. Assume y_i given $\mathbf{x}_i$ has Poisson distribution with a density function:

$$\Pr(y_i|\mu_i, \mathbf{x}_i) = \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!} \qquad \text{for} \quad y_i = 0, 1, 2 \cdots , \qquad (3)$$

where μ_i is the mean value of the response variable y_i. Since μ_i is always positive, the link function, which demonstrates the relationship between the expected value of response variable and the covariates, generally has the logarithm form shown below.

$$\ln(\mu_i) = \ln(\mathrm{E}(y_i|\mathbf{x}_i)) = \mathbf{x}_i'\beta , \qquad (4)$$

where ln means natural logarithm, $\mathbf{x}_i$ represents the covariates, $\mathbf{x}_i'$ represents the transpose of the vector $\mathbf{x}_i$ and β is a vector of the unknown parameters. Here, both $\mathbf{x}_i$ and β are vertical vectors. The expected value $\mathrm{E}(\cdot)$ is equivalent to the mean value denoted by μ throughout this paper. Sometimes, the link function is also written in an exponential form:

$$\mu_i = \mathrm{E}(y_i|\mathbf{x}_i) = e^{\mathbf{x}_i'\beta} . \qquad (5)$$

According to the equidispersion property – Equation (2) of the Poisson distribution, it follows that

$$\mathrm{Var}(y_i|\mathbf{x}_i) = e^{\mathbf{x}_i'\beta} , \qquad (6)$$

Equation (3) and Equation (4) jointly define the Poisson regression model.

2.3 PRM Maximum Likelihood Estimation

The standard estimation technique for PRM is Maximum Likelihood Estimation (MLE). Based on Equation (3), we have the likelihood function of the PRM as follows:

$$\mathcal{L}(\beta|y_i,\mathbf{x}_i) = \prod_{i=1}^{n} \Pr(y_i|\mu_i,\mathbf{x}_i) = \prod_{i=1}^{n} \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!} \ . \qquad (7)$$

Substituting for μ_i based on Equation (5), one can get the log-likelihood function of the PRM which is presented below:

$$\begin{aligned} \ln \mathcal{L}(\beta|y_i,\mathbf{x}_i) &= \sum_{i=1}^{n} \ln \Pr(y_i|\mu_i,\mathbf{x}_i) \\ &= \sum_{i=1}^{n} \{y_i \mathbf{x}_i'\beta - e^{\mathbf{x}_i'\beta} - \ln y_i!\}. \qquad (8) \end{aligned}$$

If $\hat{\beta}$ is considered as an estimator of the actual β, then the Poisson maximum likelihood estimator $\hat{\beta}$ is the solution to the first-order condition:

$$\sum_{i=1}^{n} (y_i - e^{\mathbf{x}_i'\beta})\mathbf{x}_i = \mathbf{0} \ . \qquad (9)$$

The commonly used method to solve Equation (9) is Newton-Raphson iteration. Convergence is guaranteed, because the log-likelihood function is globally concave [3].

2.4 PRM Prediction

When the model is built, one can utilize this model to predict the response variable for each module. A prediction model for the PRM can be written as

$$\hat{y}_i = \hat{\mu}_i = e^{\mathbf{x}_i'\hat{\beta}} \qquad \text{for } i = 1, 2, \cdots \ . \qquad (10)$$

3. Zero-Inflated Poisson Regression Model

In software quality modeling, it is often seen that a data set has excess zeros for the response variable. As pointed out before, a pure PRM is not suitable for this situation [9]. A zero-inflated count model is an effective way of dealing with this problem. A zero-inflated count model assumes that zeros can be generated by a different process than positive counts. Thus, this kind of model modifies the mean structure, so that the conditional variance and the probability of zero count increase. Mullahy used a With-Zeros (WZ) model in his application [16]. In 1992, Lambert first introduced the zero-inflated Poisson regression model [14].

3.1 Definition

In a zero-inflated model, we group all zeros into two parts. One part comes from the perfect modules, i.e., modules with zero faults. The other part comes from the non-perfect modules where the number of faults follows some standard distribution. In a ZIP regression model, we introduce a parameter ψ which represents the probability of a module being perfect. Hence, the probability of the module being non-perfect is $1-\psi$. Also, we assume that in non-perfect modules, the number of faults follows a Poisson distribution.

Let the response variable $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ be independent and

$$\begin{aligned} y_i &= \textit{perfect} && \text{probability } \psi_i \ , \\ &= \textit{non-perfect} \sim \text{Poisson}(\mu_i) && \text{probability } 1-\psi_i \ . \end{aligned}$$

The probability density function (pdf) of the ZIP regression model therefore is

$$\Pr(y_i|\mathbf{x}_i,\psi_i) = \begin{cases} \psi_i + (1-\psi_i)e^{-\mu_i}, & y_i = 0, \\ (1-\psi_i)\frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!}, & y_i = 1, 2, \cdots \ . \end{cases} \qquad (11)$$

The ZIP regression model is obtained by adding the next two link functions:

$$\begin{aligned} \ln(\mu_i) &= \mathbf{x}_i'\beta && (12) \\ \text{logit}(\psi_i) &= \ln \frac{\psi_i}{1-\psi_i} = \mathbf{x}_i'\gamma \ , && (13) \end{aligned}$$

where both β and γ represent the coefficient vectors of the covariates $\mathbf{x}_i$, but one serves as the log function of mean μ_i and the other one serves as the logit function of probability ψ_i instead.

3.2 ZIP Maximum Likelihood Estimation

The standard estimation technique for ZIP regression model is also based on the maximum likelihood estimation. The joint likelihood function for a ZIP regression model is

$$\begin{aligned} \mathcal{L}(\beta,\gamma|y_i,\mathbf{x}_i) &= \prod_{i=1}^{n} \Pr(y_i|\mathbf{x}_i,\psi_i) \\ &= \prod_{y_i=0} \{\psi_i + (1-\psi_i)e^{-\mu_i}\} \cdot \\ &\quad \prod_{y_i>0} \left\{(1-\psi_i)\frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!}\right\} \ . \qquad (14) \end{aligned}$$

Let $1(y_i = 0)$ denote a function that its value is 1 when $y_i = 0$ and 0 otherwise. The joint log-likelihood function for the ZIP regression model therefore is

$$\begin{aligned} & \ln \mathcal{L}(\beta, \gamma | y_i, \mathbf{x}_i) \\ = & \sum_{i=1}^{n} 1(y_i = 0) \ln(\exp(\mathbf{x}_i{}'\gamma) + \exp(-\exp(\mathbf{x}_i{}'\beta))) \\ & + \sum_{i=1}^{n} (1 - 1(y_i = 0))(y_i \mathbf{x}_i{}'\beta - \exp(\mathbf{x}_i{}'\beta)) \\ & - \sum_{i=1}^{n} \ln(1 + \exp(\mathbf{x}_i{}'\gamma)) \; . \qquad (15) \end{aligned}$$

Since $y_i!$ does not affect parameters estimation, terms involving factorials have been dropped from the computation of the log-likelihood function.

The Newton-Raphson algorithm can be used to maximize the log-likelihood function (15). Since the number of parameters in a ZIP model is twice as many as it in a PRM, the computation therefore becomes more complex. Other methods such as EM algorithm [14] also perform successfully.

3.3 ZIP Regression Model Prediction

When the parameters (β, γ) are estimated, one can utilize the ZIP regression model to predict the response variable for each module, and find the probability for each module being perfect. The probabilities of the response variable being various counts can be obtained by the pdf - Equation (11) of the ZIP regression model.

The mean prediction for the ZIP regression model is

$$\mathrm{E}(y_i | \mathbf{x}_i) = (1 - \hat{\psi}_i) e^{\mathbf{x}_i{}'\hat{\beta}} \; . \qquad (16)$$

where $\hat{\psi}_i$ is the predicted probability of module i being perfect. $\hat{\psi}_i$ is estimated by the following:

$$\hat{\psi}_i = \frac{e^{\mathbf{x}_i'\hat{\gamma}}}{1 + e^{\mathbf{x}_i'\hat{\gamma}}} \; . \qquad (17)$$

4. Performance Metrics

There are many ways to evaluate the quality of the prediction. Average Absolute Error (AAE) and Average Relative Error (ARE) [17] are the common ways. The definition of AAE and ARE are:

$$\begin{aligned} \mathrm{AAE} &= \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \\ \mathrm{ARE} &= \frac{1}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{y_i + 1} \; , \end{aligned}$$

where n is the number of modules in the test data set, y_i and $\hat{y}_i$ represent the actual and predicted value of the response variable. In ARE, since actual response variable may be zero, we use one more than the actual response variable as the denominator to make the definition always well-defined [13]. Lower values of ARE and AAE indicate better prediction quality.

5. Hypothesis Testing

Most hypothesis tests on the regression coefficient and dispersion parameters in count models can be obtained by applying the three classic hypothesis testing methods: Likelihood Ratio (LR) test, Wald test and Langrange Multiplier (LM) test. But there are still some hypothesis tests which can not be obtained by directly using these methods. The main reason is that the relationship between the two models are different.

Given two conditional models, we can group the relationship between the two models into two categories: nested and non-nested models. Roughly speaking, two models are *nested* if one model is a special case of the other, whereas *non-nested* models imply that neither model can be represented as a special case of the other. Further distinction can be made in the non-nested case: overlapping and stricted non-nested models. *Overlapping* means that the two models have some common specifications and *stricted non-nested* means that the two models have no common specifications between them. LR, Wald and LM tests are suitable for the nested models [10]. But for the non-nested case, we may need to use different techniques. Many approaches were proposed such as [5, 6, 18]. In this paper, we use a method developed by Vuong [18].

Greene [10] pointed out that PRM and ZIP models are not nested. $\psi_i = 0$ is necessary for the ZIP model to reduce to the PRM. However, this does not occur when setting the parameter γ in Equation (13) to $\mathbf{0}$. In fact, when $\gamma = \mathbf{0}$, it is always true that $\psi_i = 0.5$. Consequently, Greene used a test proposed by Vuong [18] for the non-nested models. Let $\hat{\Pr}_1(y_i|\mathbf{x}_i)$ be the predicted probability of the first model, and $\hat{\Pr}_2(y_i|\mathbf{x}_i)$ be the predicted probability of the second one. Let m_i be defined by:

$$m_i = \ln \left(\frac{\hat{\Pr}_1(y_i|\mathbf{x}_i)}{\hat{\Pr}_2(y_i|\mathbf{x}_i)} \right) \; . \qquad (18)$$

Then, the Vuong's statistic for testing the nested hypothesis of model 1 versus model 2 is:

$$v = \frac{\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} m_i\right)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i - \bar{m})^2}} = \frac{\sqrt{n}\bar{m}}{s_m} \; , \qquad (19)$$

where $\bar{m}$ denotes the mean of m_i, for $i = 1, 2, 3, \cdots$, and s_m denotes the standard deviation of m_i, for

$i = 1, 2, 3, \cdots$. Vuong's statistic v is used to test the hypothesis that $E(m_i) = 0$, i.e., for each module, the two models have the same predicted probabilities that the response variable equals its actual value y_i. Vuong showed that statistic v is bidirectional and asymptotically normal. To compare the PRM and the ZIP regression model, we just replace $\hat{\Pr}_1(y_i|\mathbf{x}_i)$ by the pdf of the ZIP regression model and replace $\hat{\Pr}_2(y_i|\mathbf{x}_i)$ by the pdf of the PRM in Equation (18). For a given significance level p, if $v \geq Z_{1-p/2}$, then mode 1, ZIP, is chosen; if $v \leq -Z_{1-p/2}$, then model 2, PRM, is selected; otherwise, i.e., $|v| \leq Z_{1-p/2}$, either model can be selected. In this situation, we prefer the simpler model, PRM, for our study.

6. Case Study

6.1 System Description

In this case study, we used data collected from two very large Windows©-based software applications. These applications were very similar and contained common software code. Data collected from both applications, was analyzed simultaneously. These computer applications were written in C++ with 1211 source code files, and over 27.5 million lines of code in each application. Source code files were considered as modules in this case study. The metrics were collected using a combination of several tools and databases. Table 1 lists five software metrics used in this data set. *Ins* is a process metric and the other four are product metrics. These five metrics were used as independent variables. The response variable in this case study was *Fault*, the number of faults discovered in a source file during system test. In the system, about 67 percent of modules had no fault. The maximum number of the faults in one module was 97. Figure 1 shows the distribution of *Fault*.

6.2 Preprocess Data

We studied all 1211 modules and applied the data splitting technique to the data set, so the predictive quality of the models could be evaluated. The data set was impartially partitioned into fit and test data sets. Two thirds of the modules (807) were assigned to the fit data set and the remaining third (404) were assigned to the test data set. Table 2 shows the statistics for the *Fault*.

Table 1. Software product metrics

Symbol	Description
Ins	Number of times the source file was inspected prior to system test release.
BCode	Number of lines of code for the source file prior to the coding phase. This represents auto-generated code.
SCode	Number of lines of code for the source file prior to system test release.
Bcomm	Number of lines of commented code for the source file prior to the coding phase. This represents auto-generated code.
Scomm	Number of lines of commented code for the source file prior to system test release.

Table 2. Descriptive Statistics for *Fault*

Data	Number of Obs.	Average	stdDev
Total	1211	1.53	5.33
Fit	807	1.58	5.60
Test	404	1.43	4.75

6.3 Building Count Models

Estimate Parameters of PRM. Based on the maximum likelihood estimation for the PRM, we estimated the Poisson regression parameters as shown in Table 3. The *Fault* predicted by the PRM is given as

$$Fault_{\mathrm{PRM}} = \exp(\hat{\beta}_0 + \hat{\beta}_1 Ins + \hat{\beta}_2 BCode + \hat{\beta}_3 SCode + \hat{\beta}_4 BComm + \hat{\beta}_5 SComm)\ . \quad (20)$$

Observing Table 3, we found that $\hat{\beta}_0$ is negative, while others are positive. This implies that the variable *Fault* increases as each independent variable increases.

Estimate Parameters of ZIP Model. In this system, two thirds of the modules had no fault. We built the zero-inflated Poisson model using the fit data set. The ZIP parameters were estimated by the MLE technique and the results are shown in Table 3. The *Fault* prediction for the ZIP model therefore is given as

$$Fault_{\mathrm{ZIP}} = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 Ins + \hat{\beta}_2 BCode + \hat{\beta}_3 SCode + \hat{\beta}_4 BComm + \hat{\beta}_5 SComm)}{1 + \exp(\hat{\gamma}_0 + \hat{\gamma}_1 Ins + \hat{\gamma}_2 BCode + \hat{\gamma}_3 SCode + \hat{\gamma}_4 BComm + \hat{\gamma}_5 SComm)} \quad (21)$$

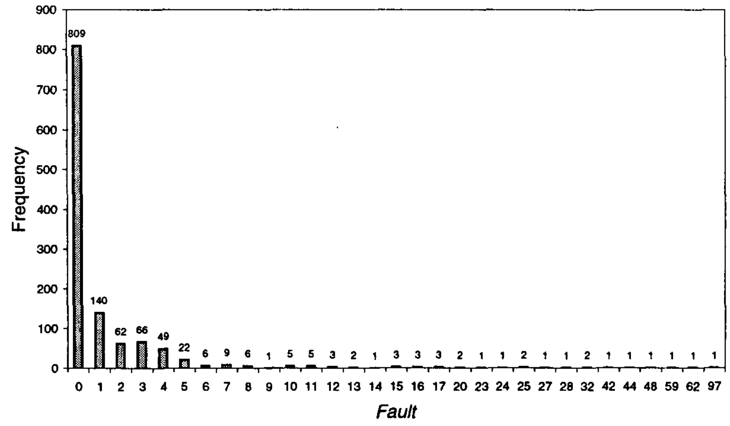

Figure 1. Distribution of *Fault*

Table 3. Regression Models

Parameter	PRM	ZIP
$\hat{\beta}_0$	-0.271365	1.58764
$\hat{\beta}_1$	0.035910	-0.27403
$\hat{\beta}_2$	0.000443	0.00192
$\hat{\beta}_3$	0.000197	-0.00633
$\hat{\beta}_4$	0.005174	-0.00705
$\hat{\beta}_5$	0.000674	0.00385
$\hat{\gamma}_0$	-	0.89976
$\hat{\gamma}_1$	-	0.02547
$\hat{\gamma}_2$	-	0.00013
$\hat{\gamma}_3$	-	0.00022
$\hat{\gamma}_4$	-	0.00250
$\hat{\gamma}_5$	-	0.00047

Table 4. Prediction Qualities

Models	Statistics	Total	Zero	NonZero
PRM	AAE	2.2848	0.9627	4.9488
	ARE	0.8193	0.9627	0.5304
	std AE	10.1928	0.1592	17.4371
	std RE	0.6391	0.1592	1.0299
ZIP	AAE	1.4207	0.7847	2.7023
	ARE	0.6759	0.7847	0.4565
	std AE	2.6933	0.4027	4.379
	std RE	0.4518	0.4027	0.467

6.4 Testing for Zero-Inflation

We applied Vuong's hypothesis test described in section 5 to this case study. The test statistic v is 7.25. This implies that the ZIP model is more appropriate at the significance level of less than 1% ($p < 0.01$).

6.5 Comparisons

We used AAE and ARE to evaluate the predictive qualities of the models. We found out that AAE and ARE of the ZIP model are significantly less than those of its PRM counterpart. This demonstrates that the ZIP model on average has better predictive accuracy than the PRM. Table 4 summarizes prediction qualities of this system. Standard deviation of absolute error and relative error are denoted by std AE and std RE respectively. The upper half of the table lists the statistics of PRM and lower half of the table lists the statistics of the ZIP model. For each model, we report three values: Total, Zero and NonZero. Total means the entire test data set which consists of 404 modules. We further divided our test data set into Zero and NonZero parts. The modules whose response variable values were zeros were assigned to the Zero part. Conversely, those whose response variable values were not zeros were assigned to NonZero part. The number of modules in Zero and NonZero parts are 270 and 134 respectively. To compare the statistics of the two models, we used the Z-test [1]. The Z-test results are shown in Table 5. We summarize our findings below.

1. The ZIP model has better predictive accuracy than the PRM. This is indicated by the values of both AAE and ARE in Table 4.

2. AAE of the ZIP model is lower than its PRM coun-

Table 5. Z - Test Statistics

Statistics	Data	Z -value	p -value
AAE	Total	1.65	0.04947
	Zero	6.75	<0.00002
	NonZero	1.45	0.07353
ARE	Total	3.68	0.00012
	Zero	6.75	<0.00002
	NonZero	0.76	0.22363

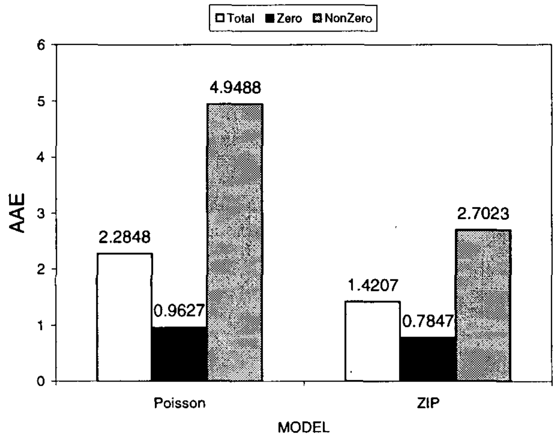

Figure 2. AAEs of Test Data Set

terpart at significance level of less than 5%.

3. ARE of the ZIP model is lower than its PRM counterpart at significance level of less than 1%.

4. When the data set are divided into two parts, zero and non-zero, we observe that for both parts, AAE and ARE of the ZIP model are much better than those of the PRM. Z-tests show that the AAE and ARE improvements of the ZIP model for zero part are larger than they are for non-zero part. Figure 2 and Figure 3 demonstrate the statistics of AAE and ARE for the different parts respectively.

5. For the zero part, both AAE and ARE of the ZIP model are lower than its PRM counterpart with a significance level of less than 1%.

6. For the non-zero part, AAE and ARE of the ZIP model are lower than its PRM counterpart with a significance level of 7% and 22% respectively.

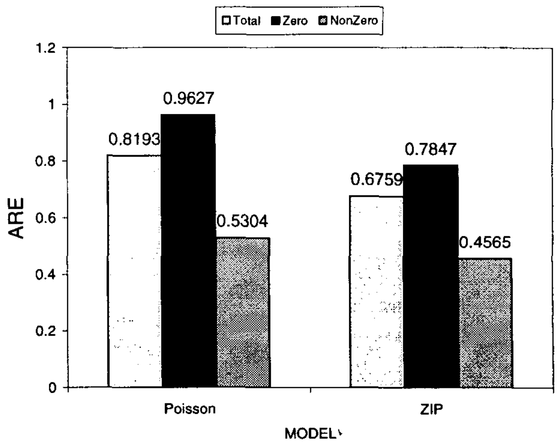

Figure 3. AREs of Test Data Set

7. Conclusions

In this paper, we discussed two count models, PRM and ZIP. PRM is the most commonly used count model in software reliability engineering, but it assumes equidispersion. When the response variable set has a large proportion of zeros, PRM is not appropriate. Zero-inflated models such as ZIP can be considered. This kind of model increases the conditional variance and probability of zero count by modifying the mean structure. One contribution of this paper is that to our knowledge, this is the first application of the ZIP technique for building software quality models. In our case study, we first developed two count models, PRM and ZIP, for a full-scale industrial software system and then performed a hypothesis test, PRM versus ZIP. Since the relationship between PRM and ZIP is not nested, we used a test proposed by Vuong instead of the generally used LR test. Vuong's test showed that the ZIP regression model fits better for this case study. Finally, we compared the predictive qualities of the two competing models and concluded that the ZIP regression model predicts better than the PRM, which agrees with our hypothesis testing result. Similar results were obtained for another case study. Future work will investigate other count modeling techniques and compare them with PRM and ZIP model.

Acknowledgments

We thank Michele Lim and Linda Lim for collecting the case study data.

References

[1] M. L. Berenson, D. M. Levine, and M. Goldstein. *Intermediate Statistical Methods and Applications: A Computer Package Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[2] A. C. Cameron and P. K. Trivedi. Count data models for finacial data. *Handbook of Statistics*, 14:363–391, 1996.

[3] A. C. Cameron and P. K. Trivedi. *Regression Analysis of Count Data.* Cambridge University Press, 1998.

[4] D. R. Cox and D. V. Hinkley. *Theoretical Statistics.* Chapman and Hall Ltd, London, 1974.

[5] R. D. Cox. Tests of seperate families of hypotheses. *Proceedings of the Fourth Berkeley Symposium on mathematical Statistics and Probablity*, pages 105–123, 1961.

[6] R. D. Cox. Further results on tests of seperate families of hypotheses. *Journal of the Royal Statistical Society*, B(24):406–424, 1962.

[7] C. Dean, J. F. Lawless, and G. E. Willmot. A mixed poissoon-inverse-gaussian regression model. *The Canadian Journal of Statistics*, 17(2):171–181, 1989.

[8] G. Dionne, M. Artis, and M. Guillen. Count data models for a credit scoring system. *Journal of Empirical Finance*, 3:303–325, 1996.

[9] W. H. Green. Accounting for excess zeros and sample selection in poisson and negative binomial regression models. Technical Report EC-94-10, Economics Department, New York University, 1994.

[10] W. H. Greene. *Econometric Analysis.* Prentice-Hall Inc. Upper Saddle River, New Jersity 07458, New York University, 4 edition, 2000.

[11] T. M. Khoshgoftaar and E. B. Allen. Logistic regression modeling of software quality. *International Journal of Reliability, Quality and Safety Engineering*, 6(4):303–317, Dec. 1999.

[12] T. M. Khoshgoftaar, E. B. Allen, and J. C. Busboom. Software quality modeling: The software measurement analysis and reliability toolkit. In *Proceedings of the Twelfth IEEE International Conference on Tools with Artificial Intelligence*, pages 54–61, Nov. 2000.

[13] T. M. Khoshgoftaar, J. C. Munson, B. B. Bhattacharya, and G. D. Richardson. Predictive modeling techniques of software quality from software measures. *IEEE Transactions on Software Engineering*, 18(11):979–987, Nov. 1992.

[14] D. Lambert. Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, Feb. 1992.

[15] S.-P. Miaou. The relationship between truck accidents and geometric design of road sections: Poisson versus negative binomial regression. *Accident analysis and prevention*, 26(4):471–482, 1994.

[16] J. Mullahy. Specification and testing of some modified count data models. *Journal of Econometrics*, 33:341–365, 1986.

[17] R. M. Szabo and T. M. Khoshgoftaar. Exploring a poisson regression fault model: A comparative study. Technical Report TR-CSE-00-56, Florida Atlantic University, 2000.

[18] Q. H. Vuong. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57(2):307–333, Mar. 1989.