UNIVERSITY
of
OTAGO

SAPERE AUDE

Te Whare Wānanga o Otāgo

Bayesian Statistical Effort Prediction Models for Data-centred 4GL Software Development

Chikako van Koten
Andrew Gray

The Information Science Discussion Paper Series

Number 2005/09
November 2005
ISSN 1172-6024

University of Otago

Department of Information Science

The Department of Information Science is one of six departments that make up the School of Business at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in postgraduate research programmes leading to MCom, MA, MSc and PhD degrees. Research projects in spatial information processing, connectionist-based information systems, software engineering and software development, information engineering and database, software metrics, distributed information systems, multimedia information systems and information systems security are particularly well supported.

The views expressed in this paper are not necessarily those of the department as a whole. The accuracy of the information presented in this paper is the sole responsibility of the authors.

Copyright

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper, or for subsequent publication details. Please write directly to the authors at the address provided below. (Details of final journal/conference publication venues for these papers are also provided on the Department's publications web pages: http://www.otago.ac.nz/informationscience/pubs/). Any other correspondence concerning the Series should be sent to the DPS Coordinator.

Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND

Fax: +64 3 479 8311
email: dps@infoscience.otago.ac.nz
www: http://www.otago.ac.nz/informationscience/

Bayesian statistical effort prediction models for data-centred 4GL software development

C. van Koten[1] and A.R. Gray

Department of Information Science,University of Otago, P.O.Box 56, Dunedin, New Zealand

Abstract

Constructing an accurate effort prediction model is a challenge in Software Engineering. This paper presents three Bayesian statistical software effort prediction models for database-oriented software systems, which are developed using a specific 4GL tool suite. The models consist of *specification-based* software size metrics and development team's productivity metric. The models are constructed based on the subjective knowledge of human expert and calibrated using empirical data collected from 17 software systems developed in the target environment. The models' predictive accuracy is evaluated using subsets of the same data, which were not used for the models' calibration. The results show that the models have achieved very good predictive accuracy in terms of MMRE and pred measures. Hence it is confirmed that the Bayesian statistical models can predict effort successfully in the target environment. In comparison with commonly used multiple linear regression models, the Bayesian statistical models' predictive accuracy is equivalent in general. However, when the number of software systems used for the models' calibration becomes smaller than five, the predictive accuracy of the best Bayesian statistical models are significantly better than the multiple linear regression model. This result suggests that the Bayesian statistical models would be a better choice when software organizations/practitioners do not posses sufficient empirical data for the models' calibration. The authors expect those findings encourage more researchers to investigate the use of Bayesian statistical models for predicting software effort.

Key words: Effort prediction, 4GL, Bayesian statistics, Regression, Software metrics

[1] Corresponding author. Tel.: +64-3-479-8142; fax: +64-3-479-8311.
E-mail address: ckoten@infoscience.otago.ac.nz

1 Introduction

Accurate effort prediction at an early stage is often an important factor for successful software development. In order to predict software development effort, there exist a number of effort prediction models, including well-known COCOMO [3] and models using Function Points (FP) [1]. However, these existing models are, in general, empirical models constructed using historical data collected from a number of software systems developed in specific development environments. As a consequence, the applicability of those models is often limited to systems developed in those environments. On the other hand, the increasing number of software development tools available today enables software systems to be developed in very different environments. Some organizations use a data-centred fourth-generation-language (4GL) software development tool. Data-centred 4GL software development tools enable database-oriented transaction processing systems (TPSs) and/or management information systems (MISs) to be developed in a rapid manner. However, a number of studies have showed that traditional effort prediction models are not able to predict development effort accurately when a data-centred 4GL software development tool is used [16,8].

The situation described above has prompted researchers to construct new effort prediction models for data-centred 4GL software development [24,26,18,9]. Those effort prediction models are a linear regression model that consists of software size metrics collected in environments, where a specific development tool was used. The software size metrics chosen for each of those models are different but all *specification-based*, that is, derived from a software system's specifications such as Entity Relationship Diagrams (ERDs) and Function Hierarchy Diagrams (FHDs). Those models achieved good predictive accuracy in terms of mean magnitude of relative error (MMRE) and a measure called pred. Both MMRE and pred are commonly used predictive accuracy measures among researchers [11,20].

However, due to the very same reason as mentioned previously about empirical effort prediction models in general, the applicability of the above new effort prediction models is limited to a specific development environment in each case. Consequently, it is necessary to construct a new effort prediction model when a different development tool is used. The organization studied in this paper started using a data-centred 4GL software development tool suite, Oracle's *Designer 6i* and *Developer 6i*, in 2002. This created the need for a new effort prediction model and raised the following research question. How can an accurate development effort prediction model be constructed for this specific data-centred 4GL software development environment?

This paper presents three effort prediction models constructed using Bayesian statistics. Bayesian statistics are based on Bayesian probability theory and Bayesian inference [5]. Bayesian inference allows a predictive model to be constructed based on the subjective knowledge of human expert and calibrated using empirical data. The models presented in this paper are calibrated using subsets of data, which were collected from 17 software systems developed in the target environment. The models' predictive accuracy is evaluated using magnitude of relative error (MRE), pred and measures related to the absolute residuals. The models' predictive accuracy is then, compared with multiple linear regression models, since regression has been a popular and often a successful technique for software effort prediction. The term *predictive accuracy* in this paper means how well a predictive model constructed from known cases can predict the outcome of an unknown case. Hence the models are calibrated using subsets of the data and evaluated using other subsets, which do not contain the data already used for the calibration.

An application of Bayesian statistics to Software Engineering is currently limited to a small number of studies of development effort prediction [2,4,21,23], defect prediction [10,19] and maintainability prediction [25]. Those studies used a technique known as Bayesian network [7,15]. Bayesian network is a probabilistic network constructed based on the concepts of Bayesian statistics. Hence Bayesian network models are a Bayesian statistical model. However, the existing Bayesian network models unfortunately have restrictions on use of continuous probability density functions. Consequently those restrictions limit the use of the existing models to the cases that involve discrete variables only. Hence in the above studies of development effort prediction, Bayesian statistics were used only for classifying software projects according to the various categorical attributes. However, use of Bayesian statistics in software development effort prediction should not be limited to classifying software projects, since Bayesian statistics are also capable of making numerical prediction of continuous variables. Hence this paper applies Bayesian statistics for making numerical prediction of software development effort in the target environment, and evaluates the models' predictive accuracy.

In comparison with other software effort prediction models, a Bayesian statistical effort prediction model is considered to have a unique advantage due to the ability to incorporate existing human expert's knowledge into empirical data. It is often the case that organizations and practitioners posses some knowledge about their software development practices through experiences in the past development. Hence it is only natural if this knowledge can be utilized for constructing an effort prediction model and then, the model can be calibrated using available empirical data. In addition, even when organizations and practitioners do not posses sufficient empirical data for model calibration, a Bayesian statistical model is still expected to predict effort better than the existing models, since the Bayesian model makes use of not only empirical data but also the expert's knowledge, while the existing models only make use of empirical data. Hence in this study, the authors are particularly interested in examining the predictive accuracy of the Bayesian statistical models when they are calibrated using a very small amount of data. Because of this, the Bayesian statistical models in this study are calibrated using four different sized subsets, varying from the largest consisting of all the 17 systems to the smallest consisting of only five systems.

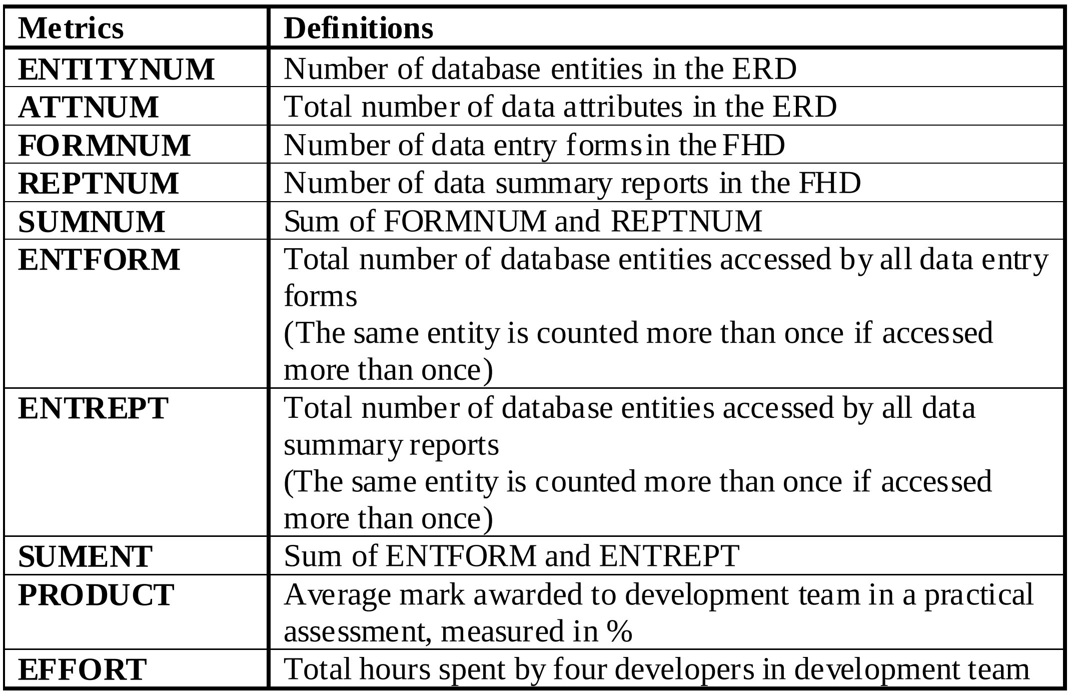
The findings of this study should be able to justify the use of Bayesian statistical models for predicting software effort. This should also encourage more researchers to engage themselves in investigating the potential of Bayesian statistical models further. The results from those further studies should reveal the full potential of Bayesian statistical models. The authors expect Bayesian statistical models to take a leading role among the other models due to the above advantages. Software organizations and practitioners will benefit from Bayesian statistical models particularly, since the models are applicable even when the system of interest does not have sufficient historical data.

The structure of the reminder of this paper is as follows. Section 2 describes data-centred 4GL software systems studied, software metrics chosen, and the characteristics of data collected. Section 3 briefly explains Bayesian probability theory and Bayesian inference, which Bayesian statistical models are based on. Section 4 describes the new Bayesian statistical software development effort prediction models. This is followed by Section 5 that describes the multiple linear regression models. Section 6 describes the predictive accuracy measures used. Section 7 evaluates the Bayesian statistical models' predictive accuracy and compares them with the multiple linear regression models. Finally Section 8 presents conclusions and a direction of future studies.

2 Data collection

2.1 Software systems

The empirical data was collected from a total of 17 small or medium sized TPSs and/or MISs, which were developed using Oracle's *Designer 6i* and *Developer 6i*. Each system was developed by a team of four developers, who were final year undergraduate students taking computer and/or information science as a major at a university in New Zealand. The development was undertaken as the final part of their university course and the systems were built for business clients outside the university in a manner similar to commercial development. The development teams followed the same development methodology, which was taught in the course.



Metrics	Definitions
ENTITYNUM	Number of database entities in the ERD
ATTNUM	Total number of data attributes in the ERD
FORMNUM	Number of data entry formsin the FHD
REPTNUM	Number of data summary reports in the FHD
SUMNUM	Sum of FORMNUM and REPTNUM
ENTFORM	Total number of database entities accessed by all data entry forms (The same entity is counted more than once if accessed more than once)
ENTREPT	Total number of database entities accessed by all data summary reports (The same entity is counted more than once if accessed more than once)
SUMENT	Sum of ENTFORM and ENTREPT
PRODUCT	Average mark awarded to development team in a practical assessment, measured in %
EFFORT	Total hours spent by four developers in development team

Table 1
Software Metrics

2.2 Software metrics

This paper uses *specification-based* software size metrics as predictor variables for effort prediction. This is because *specification-based* software size metrics are available at an early stage of development, and as was mentioned in Section 1, the models consisting of those metrics have achieved good predictive accuracy in other data-centred 4GL software development environments. The software systems studied were built according to the specifications, which were described in a form of ERD and FHD. The ERD was used to show the system's database entities and their relationships. Under the development methodology employed, the FHD was specifically used to show the system's user interface structure, that is, what data entry forms and data summary reports are required for the system and how those forms and reports are organized.

Our study initially listed eight candidate *specification-based* software size metrics, which are shown in Table 1. However, some of those *specification-based* software size metrics are clearly highly correlated with the other(s) and it is known that multicollinearity causes a problem of a large standard error of the predicted variable in multiple regression. Thus, we selected only the metrics that were expected not to be highly correlated. This resulted in using four *specification-based* software size metrics as the final candidates. They are FORMNUM, REPTNUM, ENTFORM and ENTREPT.

We also use an additional predictor variable, PRODUCT. The PRODUCT metric measures the productivity of a development team. It is measured as the average mark of four developers in each team, which was awarded in a practical development skill assessment. The assessment was carried out at the time of the development. In the assessment, each developer was required to built an identical software system individually using the same development tool suite within a specific time. The specifications of the system were given in a form of ERD and FHD. This assessment environment was very similar to that of the development. Hence, the PRODUCT metric is expected to reflect the development team's average productivity adequately. The PRODUCT metric is expressed in % by taking the full mark of the assessment as 100%. The inclusion of PRODUCT is expected to improve the effort prediction model's predictive accuracy, because it is known that the productivity of highly skilled developers is up to 30 times higher than low-skilled developers [13].

The EFFORT metric measures the total effort spent by a development team in hours. That is, the sum of the effort spent by four individual developers in the team. The EFFORT data were collected from log books of individual developers. Each developer was required to record their individual actual effort in hours each day into their own log book during the development. The recorded individual effort was then checked by the team leader for accuracy. The developer's academic achievement in the course was not subject to the amount of the recorded effort. Hence, we have a confidence in the accuracy of the recorded EFFORT data.

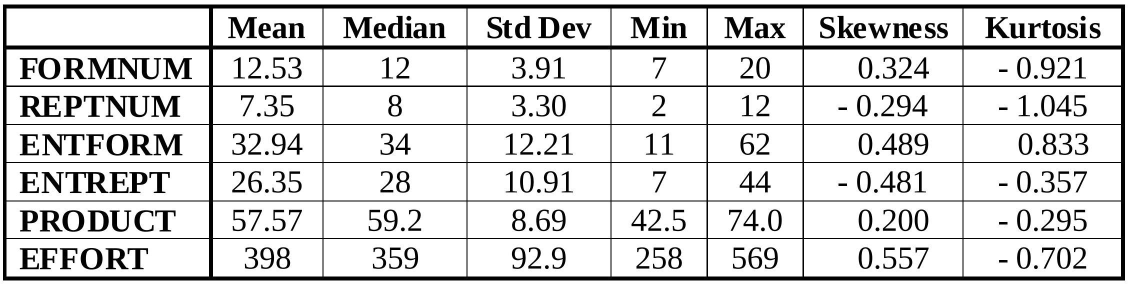
2.3 Characteristics of data

As was mentioned in the previous subsection, the software metric data collected in this study contains the PRODUCT metric, which is the average mark of each four student developers from a practical assessment in a university course. In order to protect those students' privacy, the authors are not allowed to publish the data. However, the data can be obtained from the authors by request by making an agreement about the confidentiality.

The descriptive statistics of the data are shown in Table 2. Table 2 indicates that all the variables are approximately normally distributed.

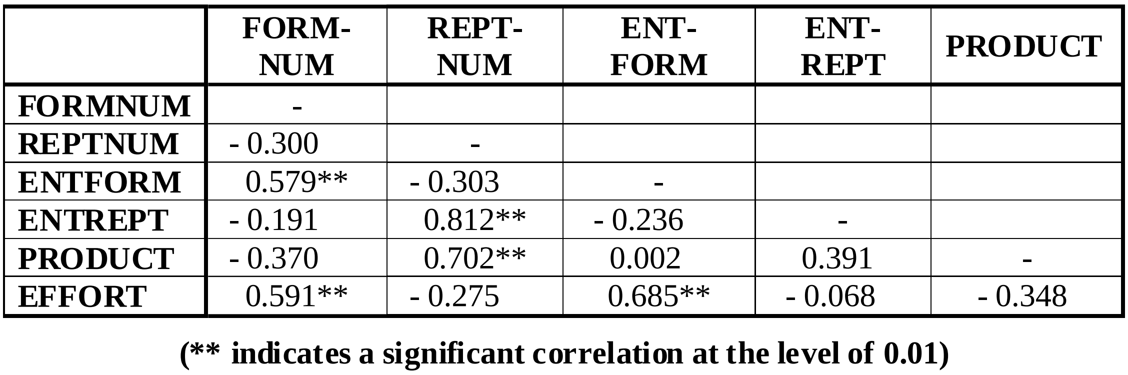
The Spearman's rank correlation coefficients between each pair of the metric variables are shown in Table 3. This paper presents the Spearman's rank correlation coefficients instead of the Pearson's correlation coefficients. This is because Pearson's correlation coefficient may be unstable when the size of data is as small as ours.

Table 3 shows that there is a significant bivariate correlation between EF-



	Mean	Median	Std Dev	Min	Max	Skewness	Kurtosis
FORMNUM	12.53	12	3.91	7	20	0.324	- 0.921
REPTNUM	7.35	8	3.30	2	12	- 0.294	- 1.045
ENTFORM	32.94	34	12.21	11	62	0.489	0.833
ENTREPT	26.35	28	10.91	7	44	- 0.481	- 0.357
PRODUCT	57.57	59.2	8.69	42.5	74.0	0.200	- 0.295
EFFORT	398	359	92.9	258	569	0.557	- 0.702

Table 2
Descriptive Statistics



	FORM-NUM	REPT-NUM	ENT-FORM	ENT-REPT	PRODUCT
FORMNUM	-				
REPTNUM	- 0.300	-			
ENTFORM	0.579**	- 0.303	-		
ENTREPT	- 0.191	0.812**	- 0.236	-	
PRODUCT	- 0.370	0.702**	0.002	0.391	-
EFFORT	0.591**	- 0.275	0.685**	- 0.068	- 0.348

(indicates a significant correlation at the level of 0.01)**

Table 3
Correlations of the Metrics

FORT and some of the metric variables. This indicates that those variables are possibly able to predict EFFORT.

3 Bayesian inference

3.1 Joint probability

In the probability theory, the *joint probability* distribution of a set of random variables $\mathbf{X}$ is defined as:

$$P(\theta, \mathbf{X}) = P(\mathbf{X} \mid \theta)P(\theta) \quad (1)$$

where θ denotes a set of parameters that describe the *joint probability* distribution. The set of random variables $\mathbf{X}$ can include both predictor variables and the variable(s) to be predicted. In Equation 1, $P(\theta)$ is called the *prior* probability distribution of θ, and $P(\mathbf{X} \mid \theta)$ is called the *likelihood* of data. The *likelihood* of data is the amount proportional to the probability of observing a given set of data, which is a set of specific values of $\mathbf{X}$. The *likelihood* of data is calculated for the given *prior* probability distribution. Hence, Equation 1 shows that the *joint probability* distribution of $\mathbf{X}$ can be found by multiplying the *prior* probability distribution of θ by the *likelihood* of data, provided that the *prior* probability distribution of θ is known. If the *joint probability* distribution of $\mathbf{X}$ is found, the *marginal* probability distribution of an individual variable in the set $\mathbf{X}$ is also found, by integrating the *joint probability* distribution function by the other variables in the set. For example, the *marginal* probability distribution of X_1 in the set $\mathbf{X} : X_1, \cdots, X_n$ is found by integrating the *joint probability* distribution function by $X_2, \cdots, X_n$. The term *marginal* means 'unconditional'. That is, the *marginal* probability of X_i is the probability of a value of X_i being observed regardless of the values of any other variables in the set $\mathbf{X}$.

3.2 Conditional probability and Bayes' theorem

On the other hand, in Bayesian probability theory, a relationship of two events X and Y is defined in the *conditional probability*, $P(Y \mid X)$, which is the probability of event Y conditional on a given outcome of event X. The *conditional probability* is calculated using Bayes' Theorem:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \tag{2}$$

where $P(X \mid Y)$ is the *conditional probability* of event X given event Y, and $P(X)$ and $P(Y)$ are the *marginal* probability of events X and Y respectively. In the Bayes' Theorem, $P(Y)$ and $P(Y \mid X)$ are also called the *prior* probability distribution and the *posterior* probability distribution of a random variable Y, respectively. This is because in the Bayes' Theorem, $P(Y)$, the *prior* probability distribution of Y, is updated to $P(Y \mid X)$, the *posterior* probability distribution of Y, using the given information of $P(X)$ and $P(X \mid Y)$.

Using the Bayes' Theorem 2, it is shown that:

$$P(\theta \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \theta)P(\theta)}{P(\mathbf{X})} \tag{3}$$

Considering that $P(\mathbf{X})$ is a constant for a given set of specific values of $\mathbf{X}$, the above Equation 3 means

$$P(\theta \mid \mathbf{X}) \propto P(\mathbf{X} \mid \theta)P(\theta) \tag{4}$$

That is, $P(\theta \mid \mathbf{X})$ is 'proportional to' $P(\mathbf{X} \mid \theta)P(\theta)$. From Equation 1, this means

$$\begin{aligned}&Posterior\ distribution\ of\ \theta\ \propto\\ &\ Likelihood\ of\ data\ \ \times\ \ Prior\ distribution\ of\ \theta\end{aligned} \tag{5}$$

Hence the *posterior* distribution of θ is actually 'proportional to' the *likelihood* of data × the *prior* distribution of θ. Similarly,

$$P(\theta \mid \mathbf{X}) \propto P(\theta, \mathbf{X}) \tag{6}$$

$$\begin{aligned}&Posterior\ distribution\ of\ \theta\ \propto\\ &\ Joint\ probability\ distribution\ of\ \mathbf{X}\end{aligned} \tag{7}$$

That is, $P(\theta \mid \mathbf{X})$, the *posterior* distribution of θ is actually 'proportional to' $P(\theta, \mathbf{X})$, the *joint probability* distribution of a set of random variables $\mathbf{X}$, which is described by θ.

3.3 Prediction

The above relationships 5 and 7 provide the principle of Bayesian inference. In Bayesian inference, the *prior joint probability* distribution of a set of random variables $\mathbf{X}$ is specified and updated using the likelihood of the observed data. Then the *posterior marginal* probability distribution of the variable of interest in the set $\mathbf{X}$ is obtained by integrating the *posterior joint probability* distribution of $\mathbf{X}$ by the other variables in the set. For prediction, both predictor variables and the variable(s) to be predicted are in $\mathbf{X}$. Let us assume that $\mathbf{X}$ consists of multiple predictor variables: $X_1, \cdots, X_n$ and a single predicted variable, Y. Then, the *posterior marginal* probability distribution of Y is obtained by integrating the *posterior joint probability* distribution of $\mathbf{X}$ by $X_1, \cdots, X_n$.

When the *prior joint probability* distribution of $\mathbf{X}$ is specified based on the subjective knowledge of human experts, Bayesian inference provides a framework in which human experts' knowledge can be incorporated into a predictive model together with empirical data. Bayesian statistical models are based on Bayesian inference. Hence Bayesian statistical models allow the subjective knowledge of human experts, which is expressed as the *prior joint probability* distribution, to be updated using empirical data. In addition, a Bayesian statistical model can be re-calibrated every time additional data become available. Another characteristic of Bayesian statistical models is that they provide an interval estimate of the variable of interest, since Bayesian inference outputs its entire *posterior marginal* probability distribution. This would be an advantage when an interval estimate is preferred to a point estimate. In the case of software development effort prediction, an interval estimate would be particularly useful for risk management. When a point estimate is required, an appropriate summary statistic of the probability distribution should be used, according to the need of the estimator. This study uses the mode statistic as the predicted value, since the mode is the value of EFFORT having the highest probability. In other words, the mode is the EFFORT value that most likely occur. In the case of the Bayesian statistical models proposed in this paper, the mode coincides with the mean of the distribution, which is usually more convenient to calculate.

4 Bayesian statistical models

4.1 Model description

Three different Bayesian statistical effort prediction models are constructed in this study. The first model is based on Bayesian linear regression [5]. We refer to this model as the Bayesian regression model in the reminder of this paper. A Bayesian linear regression model, in general, has a parametric form similar to the linear regression counterpart. However, it is different since the Bayesian model can specify the *prior joint probability* distribution of the variables, which includes the *prior marginal* probability distributions of the linear regression coefficient parameters. The Bayesian regression model is specified as follows:

$$\mathsf{EFFORT} \sim N(\mu, \tau) \tag{8}$$

$$\begin{aligned} \mu = {} & \beta_0 + \beta_1(X_1 - \bar{X}_1) + \beta_2(X_2 - \bar{X}_2) \\ & + \cdots + \beta_k(X_k - \bar{X}_k) \end{aligned} \tag{9}$$

$$\begin{aligned} \beta_0 &\sim N(0, 0.0001) && (10) \\ \beta_j &\sim N(0, 0.001) \quad for \ \ j = 1, \cdots, k && (11) \\ \tau &\sim Gamma\,(0.001, 0.001) && (12) \end{aligned}$$

The notation 8 means that EFFORT has a normal distribution with mean μ and variance $1/\tau$. In Equation 9 there are k predictor variables. The specific value of k is set according to the number of predictor variables actually used.

The notations 10, 11 and 12 specify the *prior marginal* probability distributions of the specific parameters in the model. These probability distributions are also updated to the corresponding *posterior* distributions using the given data in Bayesian inference. The notation 10 means that the linear regression coefficient parameter β_0 has a normal distribution initially with mean 0 and variance $1/0.0001 = 10000$. The initial mean is 0 because prior to calibration, it is equally possible for this coefficient parameter to take either a positive or negative value. On the other hand, for the initial variance, any number relatively larger than the anticipated β_0 would be sufficient. The value of 10000 is chosen in this study. Similarly the notation 11 means that each of the remaining linear regression coefficient parameter β_j has a normal distribution initially with mean 0 and variance $1/0.001 = 1000$. The initial mean is 0 from the same reason as was mentioned above. Similarly the initial variance of 1000 is chosen, although any relatively large number would be sufficient. The notation 12 means that τ, which is the reciprocal of the variance of the *prior* distribution of EFFORT, has a gamma distribution with scale parameter $1/0.001 = 1000$ and shape parameter 0.001. These parameter values are often recommended by experienced researchers of Bayesian statistics [5]. Using those specifications, the Bayesian regression model predicts EFFORT of the software system of interest as follows:

$$\mathsf{EFFORT}_{new} \sim N\left(\mu_{new}, \tau\right) \tag{13}$$

$$\begin{aligned}\mu_{new} = &\beta_0 + \beta_1(X_{1,new} - \bar{X_1}) + \beta_2(X_{2,new} - \bar{X_2}) \\ &+ \cdots + \beta_k(X_{k,new} - \bar{X_k})\end{aligned} \tag{14}$$

where $X_{1,new}, \cdots, X_{k,new}$ are the specific values of the k predictor variables, which are observed in the system. That is, the Bayesian regression model predicts EFFORT by taking the *posterior marginal* probability distribution of EFFORT_{new}.

The second Bayesian statistical model uses a multivariate normal distribution, which is a normal distribution defined under multiple variables. The mean of the multivariate normal distribution is the set of the means of those multiple variables, while the variance is the variance-covariance matrix of the variables. We refer to this model as the Bayesian MVN model. The Bayesian MVN model's specifications are as follows:

$$\begin{aligned}&\mathsf{EFFORT}, X_1, \cdots, X_k \sim MVN\left(\mu_j, \tau_{jm}\right) \\ &\quad for \quad j = 1, \cdots, k+1 \\ &\qquad\quad\;\; m = 1, \cdots, k+1\end{aligned} \tag{15}$$

$$\begin{aligned} &\mu_j \sim N\,(0, 0.000001) \qquad (16)\\ &\tau_{jm} \sim Wishart\,(C_{jm}, k+1)\\ &\qquad where \quad C_{jm} = \begin{cases} 1 & \text{if } j = m \\ 0 & \text{otherwise} \end{cases} \qquad (17) \end{aligned}$$

The specification 15 means that EFFORT and k predictor variables $X_1, \cdots, X_k$ together form a multivariate normal distribution with $mean = \mu_1, \cdots, \mu_{k+1}$ and $variance$ =:

$$\begin{pmatrix} 1/\tau_{1,1} & \cdots & 1/\tau_{1,k+1} \\ \vdots & \ddots & \vdots \\ 1/\tau_{k+1,1} & \cdots & 1/\tau_{k+1,k+1} \end{pmatrix}$$

where μ_1 = the mean of EFFORT, and $\mu_2, \cdots, \mu_{k+1}$ = the mean of each of the predictor variables $X_1, \cdots, X_k$ respectively. The notations 16 and 17 specify the *prior marginal* probability distributions of two specific parameters in the model. The notation 16 means that each mean μ_j has a normal distribution initially with mean 0 and variance $1/0.000001 = 1000000$. The initial mean is chosen to be 0, assuming that prior to calibration there is absolutely no knowledge about the mean values of the variables. The initial variance is chosen to be 1000000, although any number larger than the anticipated variances of EFFORT and the k predictor variables would be sufficient. The notation 17 means that τ_{jm} has a Wishart distribution whose initial scale matrix is a $(k+1) \times (k+1)$ identity matrix. Using those specifications, the Bayesian MVN model predicts EFFORT by taking the *posterior marginal* probability distribution of EFFORT.

The third Bayesian statistical model is a hybrid of the previous two Bayesian statistical models. We refer to this model as the Bayesian MVN-R model. The Bayesian MVN-R model's specifications are as follows:

$$\text{EFFORT} \;\sim\; N\,(\mu_y, \tau_y) \qquad (18)$$

$$\begin{aligned} &X_1 - \bar{X}_1, \cdots, X_k - \bar{X}_k \;\sim\; MVN\,(\mathbf{0}, \tau_{xjm})\\ &\quad for \quad j = 1, \cdots, k\\ &\qquad\qquad m = 1, \cdots, k \qquad (19) \end{aligned}$$

$$\begin{aligned} \mu_y = {} & \beta_1(X_1 - \bar{X}_1) + \beta_2(X_2 - \bar{X}_2)\\ & + \cdots + \beta_k(X_k - \bar{X}_k) \qquad (20) \end{aligned}$$

$$\begin{aligned} \tau_y &\sim Gamma\,(0.0001, 0.0001) && (21)\\ \beta_j &\sim N\,(0, 0.0001) \quad for \ \ j = 1, \cdots, k && (22)\\ \tau_{xjm} &\sim Wishart\,(C_{jm}, k) \\ & \qquad where \quad C_{jm} = \begin{cases} 0.00001 & \text{if } j = m \\ 0 & \text{otherwise} \end{cases} && (23) \end{aligned}$$

The notation 18 means that EFFORT has a normal distribution with mean μ_y and variance $1/\tau_y$. The notation 19 means that the variances of the k predictor variables $X_1, \cdots, X_k$ together form a multivariate normal distribution with $mean = 0, \cdots, 0$ and $variance =$:

$$\begin{pmatrix} 1/\tau_{x,1,1} & \cdots & 1/\tau_{x,1,k} \\ \vdots & \ddots & \vdots \\ 1/\tau_{x,k,1} & \cdots & 1/\tau_{x,k,k} \end{pmatrix}$$

The notations 21, 22 and 23 specify the *prior marginal* probability distributions of some specific parameters in the model. The notation 21 means that τ_y has a gamma distribution with scale parameter $1/0.0001 = 10000$ and shape parameter 0.0001, which are recommended by experienced researchers of Bayesian statistics [5]. The notation 22 means that each of the linear regression coefficient parameter β_j has a normal distribution initially with mean 0 and variance $1/0.0001 = 10000$. The initial mean of 0 and the initial variance of 10000 are chosen from the similar reasons to the Bayesian regression model. The notation 23 means that τ_{xjm} has a Wishart distribution whose initial scale matrix is a $k \times k$ diagonal matrix with the $(i, i) - th \ \ element = 0.00001$ for $i = 1, \cdots, k$. This value is also recommended by experienced researchers of Bayesian statistics [5]. Using those specifications, the Bayesian MVN-R model predicts EFFORT as follows:

$$\mathsf{EFFORT}_{new} = \mu_y + mean \ \ \mathsf{EFFORT} \qquad (24)$$

where mean EFFORT is the average of the EFFORT values in the calibration data. Hence, the Bayesian MVN-R model predicts EFFORT by taking the *posterior marginal* probability distribution of EFFORT_{new}.

4.2 Model construction

The above three models are constructed using a software tool called WinBUGS, the Windows version of BUGS (Bayesian analysis Using Gibbs Sampling). A version of WinBUGS can be obtained from Imperial College and

Medical Research Council in the U.K. under the license agreement. This study used the version 1.4.1. WinBUGS is an implementation of a numerical method of Bayesian inference. This tool allows users to build a complex Bayesian statistical model using a wide range of known distribution functions, including the distributions appeared in the models' specifications above. Then Bayesian inference is made by approximating the *posterior marginal* probability distribution of the variable(s) of interest, using a large number of random samples drawn from the *posterior joint probability* distribution by the Gibbs sampler. The Gibbs sampler is a Markov chain Monte Carlo (MCMC) algorithm [14].

For the above three models, the effort estimation procedure works as follows. The Gibbs sampler successively samples a value from the *posterior marginal* probability distribution of interest, given the values of all the predictor variables, which are observed in the system of interest. After a number of iterations, this sampling process is known to settle into a dynamic equilibrium, in which the sampling distribution is exactly proportional to the *posterior marginal* probability distribution. In other words, the sampling distribution can approximate the *posterior marginal* probability distribution. Hence, once the equilibrium is reached, the predicted value is obtained by calculating the mean of the sampling distribution. The mean is used since our estimate of EFFORT, the mode statistic, coincides with the mean for the three models.

The number of iterations required to reach the dynamic equilibrium generally depends on the complexity of the model and the initial values chosen for the sampling algorithm to start with. Hence setting appropriate initial values often reduces the number of iterations. This study set the initial value of τ to 1 in the Bayesian regression model. This study also set the initial values of τ_{jm} in the Bayesian MVN model equal to those of a $(k+1) \times (k+1)$ identity matrix. The initial values of μ_j were also set to the corresponding mean values, which were calculated from the calibration data. Similarly, in the Bayesian MVN-R model, the initial value of τ_y was set to 1, the initial values of β_j's were set to 0, and the initial values of τ_{xjm} were set to the values equal to those of a $k \times k$ identity matrix. As the result, it was observed that all the three models reached the equilibrium almost instantly. However, in order to ensure the equilibrium is reached, this study discarded the first 75000 samples and used only the following 75000 samples to approximate the *posterior marginal* probability distribution.

4.3 Model calibration

Calibration subsets

One of the difficulties in software effort prediction is that organizations and practitioners often have only a very small amount of available data when constructing an empirical model. Hence, the authors consider it particularly important that an accurate software effort prediction model can be constructed using as a small amount of empirical data as possible. As was mentioned in Section 1, Bayesian statistical models are expected to satisfy this requirement. In order to examine if our Bayesian statistical models satisfy this requirement, we calibrate the models using four different-sized subsets, which were sampled from the original dataset consisting of the 17 systems. The first subset is the original dataset. We refer to this as the original dataset in the reminder of this paper. The second subset consists of data of 13 systems, which is approximately 75% of the original dataset. Those 13 systems were randomly sampled from the original dataset without replacement, using a function provided in a statistical software package SPSS 11.0. The sampling was repeated four times and four different subsets were drawn. We refer to those subsets as the 75% subsets. The third subset consists of data of nine systems, which is approximately 50% of the original dataset. Those nine systems were sampled in the same way as the 75% subsets, resulting in four different subsets. We refer to those as the 50% subsets. The forth subset consists of data of five systems, which is approximately 25% of the original dataset. Those five systems were also sampled in the same way as the previous subsets, resulting four different subsets. We refer to those as the 25% subsets.

Calibration method

The three Bayesian statistical models are first calibrated using the original dataset. The models consist of EFFORT and all the five predictor variables: FORMNUM, REPTNUM, ENTFORM, ENTREPT and PRODUCT. We refer to those models as the full models in the reminder of this paper. That is, they are the full Bayesian regression model, the full Bayesian MVN model, and the full Bayesian MVN-R model. Then, three other models, which consist of EFFORT and only two predictor variables: ENTFORM and PRODUCT, are also calibrated using the original dataset. We refer to those models as the reduced models. That is, they are the reduced Bayesian regression model, the reduced Bayesian MVN model, and the reduced Bayesian MVN-R model. The reduced models are required since the variable selection procedures in multiple linear regression have identified those two predictor variables alone can explain the variation of EFFORT sufficiently. In other words, the other three predictor variables are redundant. Including those redundant predictor variables into the models often causes a larger standard error of the predicted value, in addition to making the models more complicated and difficult to understand. Another advantage of the reduced models is that they require less effort for data collection.

The above three reduced models are also calibrated separately using the 75%, 50% and 25% subsets. The full models are not used here since the reduced models have the advantages mentioned above over the full models.

4.4 Model validation

This study uses a *leave-one-out* validation method in order to evaluate and compare the models' predictive accuracy. The *leave-one-out* method is commonly used when the size of data is as small as ours. In the method, one case in data is reserved and a model is calibrated using the remaining cases. Then, the model's predictive accuracy is tested using the reserved case, which is unknown to the model. This procedure is repeated until all cases in the data are chosen to be the reserved case in turn.

In the case of the original dataset, the metric data from one of the 17 software systems is reserved and a model is calibrated using the data from the remaining 16 systems. This procedure is repeated 17 times, by using a different system as a test case in turn. Then, the model's overall predictive accuracy is evaluated by averaging the predictive accuracy of the 17 test cases.

In the case of the 75%, 50% and 25% subsets, the predictive accuracy of each subset is first evaluated using the above *leave-one-out* method. Then, the model's overall predictive accuracy is evaluated by averaging the predictive accuracy of the four subsets. That is, the model's predictive accuracy is evaluated using four independent samples, which were randomly drawn from the original dataset. Using four validation samples instead of one should reduce the possibility of getting a result by chance.

5 Multiple linear regression models

For a comparison, a multiple linear regression model consisting of all the five predictor variables is constructed using the original dataset. We refer to this model as the full regression model in the reminder of this paper. In addition, other multiple linear regression models, which correspond to the reduced Bayesian statistical models, are constructed using different variable selection procedures. The software package used is SPSS 11.0.

In multiple linear regression, a variable selection procedure enables only important predictor variables to be included in the model. Two commonly used variable selection procedures are backward elimination and stepwise selection. Backward elimination eliminates predictor variables whose contribution is less significant according to a specified criterion, while stepwise selection enters predictor variables whose contribution is more significant. In addition, stepwise selection re-assesses the contribution of the variables that were already entered, every time a new variable is entered. The re-assessment is performed in the same way as backward elimination using a specified elimination criterion. Backward elimination and stepwise selection often produce different models. Hence, we construct one model using backward elimination and the other using stepwise selection. The elimination criterion used in backward elimination is the p-value of the F statistic being larger than or equal to 0.1. The entering criterion used in stepwise selection is the p-value of the F statistic being smaller than or equal to 0.05. The eliminating criterion used in stepwise selection is the same as the one used in backward elimination.

In our study, the stepwise selection has produced the same model as the backward elimination for the original dataset. The model consists of only two predictor variable: ENTFORM and PRODUCT. Both parameter coefficients are significant as indicated by the associated p-values smaller than 0.05. The residual plots of the model show no sign of violation of the assumptions in linear regression. We refer to this model as the reduced regression model. Other reduced regression models are constructed for the 75%, 50% and 25% subsets.

6 Predictive accuracy measures

This paper evaluates and compares the software effort prediction models quantitatively, using the following predictive accuracy measures: absolute residual (Ab.Res.), the magnitude of relative error (MRE) and pred measures.

The Ab.Res. is the absolute value of residual given by:

$$Ab.Res. = \mid actual\ value - predicted\ value \mid \tag{25}$$

In this paper, the sum of the absolute residuals (Sum Ab.Res.), the median of the absolute residuals (Med.Ab.Res.) and the standard deviation of the absolute residuals (SD Ab.Res.) are used. The Sum Ab.Res. measures the total residuals over the dataset. The Med.Ab.Res. measures the central tendency of the residual distribution. The Med.Ab.Res. is chosen to be a measure of the central tendency because the residual distribution is usually skewed in software datasets. The SD Ab.Res. measures the dispersion of the residual distribution.

MRE is a normalized measure of the discrepancy between actual values and predicted values, given by [17]:

$$MRE = \frac{\mid actual\ value - predicted\ value \mid}{actual\ value} \tag{26}$$

In this paper, the maximum value of MRE (Max.MRE) is used. The Max.MRE measures the maximum relative discrepancy, which is equivalent to the maximum error relative to the actual effort in the prediction. The mean of MRE, the mean magnitude of relative error (MMRE):

$$MMRE = \frac{1}{n}\sum_{i=1}^{i=n} MRE_i \tag{27}$$

is also used. MMRE measures the average relative discrepancy, which is equivalent to the average error relative to the actual effort in the prediction. Sometimes MMRE is expressed in %. However, this paper follows the difinition given in Equation 27 and does not express MMRE in %.

Pred is a measure of what proportion of the predicted values have MRE less than or equal to a specified value, given by [11]:

$$Pred(q) = \frac{k}{n} \tag{28}$$

where q is the specified value, k is the number of cases whose MRE is less than or equal to q, and n is the total number of cases in the dataset. In this paper, pred(0.25) and pred(0.30) are used because those two pred measures are commonly used in the software effort prediction literature.

In order for an effort prediction model to be considered accurate, $MMRE \leq 0.25$ [6] and/or either $pred(0.25) \geq 0.75$ [6] or $pred(0.30) \geq 0.70$ [18] is suggested in the literature. On the other hand, there is a concern about MRE because MRE is biased [22] and not always reliable as a predictive accuracy measure [12]. However, MRE has been the de facto standard in the software effort prediction literature and no alternative standard exists at present. Thus, we still use MRE in this paper. However, in addition to MRE, we also use the absolute residual measures because it has shown that the absolute residual measures, in particular the SD Ab.Res., are a better measure than MRE for model comparison [12].

7 Model evaluation and comparison

7.1 Results from the original dataset

Table 4 shows the values of the predictive accuracy measures achieved by each of the effort prediction models for the original dataset. The values in

Model	Max. MRE	MMRE	Pred (0.25)	Pred (0.30)	Sum Ab.Res.	Med. Ab.Res.	SD Ab.Res.
Full Bayesian regression	0.354	0.144	0.706	0.824	939.2	42.6	46.6
Full Bayesian MVN	0.382	0.148	0.765	0.824	941.5	49.0	47.7
Full Bayesian MVN-R	0.359	0.147	0.824	0.824	945.1	54.0	45.9
Full regression	0.393	0.148	0.765	0.824	943.7	47.1	48.4
Reduced Bayesian regression	0.372	0.135	0.882	0.882	865.2	36.4	37.9
Reduced Bayesian MVN	0.404	0.136	0.882	0.941	861.1	42.4	37.1
Reduced Bayesian MVN-R	0.415	0.134	0.882	0.941	845.4	46.2	38.0
Reduced regression	0.405	0.136	0.882	0.941	861.1	42.6	37.0

Table 4
Predictive accuracy for the original dataset

this table are obtained using the *leave-one-out* validation method described in Section 4. Table 4 shows that the three full Bayesian models have achieved the MMRE values of 0.144, 0.148 and 0.147. This means that the average discrepancy between actual EFFORT and predicted EFFORT of the models is within 14.4 and 14.8% of actual EFFORT. Those MMRE values satisfy the suggested criterion of MMRE for an accurate model, that is, $MMRE \leq 0.25$. Table 4 also shows that the three full Bayesian models have achieved the pred(0.25) values of 0.706, 0.765 and 0.824, and the pred(0.30) value of 0.824. This means that more than 70% of the systems in the original dataset has a MRE value less than or equal to 0.25, and 82.4% has a MRE value less than or equal to 0.30. The pred(0.25) values of the full Bayesian MVN model and MVN-R model satisfy the suggested criterion of pred(0.25), that is, $pred(0.25) \geq 0.75$, while the pred(0.30) values of all the three full Bayesian models satisfy the criterion of pred(0.30), that is, $pred(0.30) \geq 0.70$. Those MMRE and pred values are indeed very good. Hence, it is concluded that each full Bayesian model is able to predict software development effort in the target environment quite successfully.

When comparing the models, Table 4 shows that the values achieved by the three full Bayesian models are similar to each other, and similar to those of the full regression model. In order to test if there is a significant difference between those four models, the Wilcoxon signed-rank tests are performed both on the MRE values and the absolute residuals. The results have confirmed no evidence of a significant difference between the four models. Hence, it is concluded that each full Bayesian model is able to predict software development effort in the target environment equally well to the others and the full regression model.

With regard to the reduced models, Table 4 shows that the three reduced Bayesian models have achieved the MMRE, pred(0.25) and pred(0.30) values even better than the full models. However, the Wilcoxon signed-rank tests have found no evidence of a significant difference between the reduced and full Bayesian models. Hence, it is concluded that each reduced Bayesian model is able to predict software development effort in the target environment at least equally well to the full model. Although this finding does not confirm the expectation that removing redundant predictor variables would achieve better predictive accuracy, considering that the reduced models require less effort for data collection, we regard the reduced Bayesian models preferable for practical use.

When comparing the reduced models, Table 4 also shows that the values achieved by each reduced Bayesian model are very similar to the others, and similar to those of the reduced regression model. The results from the Wilcoxon signed-rank tests have confirmed no evidence of a significant difference between those four models. Hence, it is concluded that each reduced Bayesian model is able to predict software development effort in the target environment equally well to the others and the reduced regression model.

7.2 Results from the 75% subsets

Table 5 shows the results of the four reduced models obtained from the 75% subsets. The values in this table are the average of the four different subsets, which were randomly sampled from the original dataset in Section 4. Table 5 shows that the three reduced Bayesian models still have achieved good MMRE, pred(0.25) and pred(0.30) values, although they are not as good as those of the reduced models obtained from the original dataset. Hence, it is concluded that each reduced Bayesian model is able to predict software development effort in the target environment quite successfully even when the data from only 12 systems has been used for calibration.

Table 5 also shows that all the four models have very similar values, and the Wilcoxon signed-rank tests have confirmed no evidence of a significant

Model	Max. MRE	MMRE	Pred (0.25)	Pred (0.30)	Sum Ab.Res.	Med. Ab.Res.	SD Ab.Res.
Reduced Bayesian regression	0.332	0.140	0.846	0.885	698.5	47.1	40.1
Reduced Bayesian MVN	0.357	0.140	0.846	0.904	679.9	45.9	39.0
Reduced Bayesian MVN-R	0.366	0.137	0.865	0.923	667.6	46.4	38.8
Reduced regression	0.358	0.140	0.846	0.904	679.7	46.4	39.0

Table 5
Predictive accuracy for the 75% subsets

difference. Hence, it is concluded that each reduced Bayesian model is able to predict software development effort in the target environment equally well to the others and the reduced regression model, even when the data from only 12 systems has been used for calibration.

7.3 Results from the 50% subsets

Table 6 shows the results of the four reduced models obtained from the 50% subsets. Those are again the average values of the four different subsets. Table 6 shows that the three reduced Bayesian models still have achieved good MMRE, pred(0.25) and pred(0.30) values, although they are not as good as those obtained from the original dataset and the 75% subsets. In addition, similar to the previous results, no significant differences between the four models are observed here and confirmed in the Wilcoxon signed-rank tests. Hence, it is concluded that each reduced Bayesian model is still able to predict software development effort in the target environment quite successfully, and equally well to the others and the reduced regression model, even when the data from only eight systems has been used for calibration.

7.4 Results from the 25% subsets

Table 7 shows the results of the four reduced models obtained from the 25% subsets. Those are again the average values of the four different subsets. Table 7 shows that the reduced Bayesian MVN-R model still satisfies the sug-

Model	Max. MRE	MMRE	Pred (0.25)	Pred (0.30)	Sum Ab.Res.	Med. Ab.Res.	SD Ab.Res.
Reduced Bayesian regression	0.348	0.167	0.750	0.806	599.8	66.3	45.6
Reduced Bayesian MVN	0.357	0.147	0.778	0.861	489.9	53.7	37.6
Reduced Bayesian MVN-R	0.376	0.145	0.778	0.889	489.2	47.8	37.2
Reduced regression	0.358	0.147	0.778	0.861	489.8	54.2	37.7

Table 6
Predictive accuracy for the 50% subsets

Model	Max. MRE	MMRE	Pred (0.25)	Pred (0.30)	Sum Ab.Res.	Med. Ab.Res.	SD Ab.Res.
Reduced Bayesian regression	0.896	0.716	0.050	0.050	1401.9	288.0	84.2
Reduced Bayesian MVN	0.553	0.258	0.550	0.600	563.0	86.3	108.3
Reduced Bayesian MVN-R	0.448	0.204	0.650	0.750	416.8	72.9	79.4
Reduced regression	0.622	0.274	0.550	0.600	597.4	86.4	119.2

Table 7
Predictive accuracy for the 25% subsets

gested criteria of MMRE and pred(0.30). In addition, the reduced Bayesian MVN model also seems to perform well enough for practical use. On the other hand, the reduced Bayesian regression model performs poorly.

Table 7 also shows that the values of the three Bayesian models are different. Table 8 shows the p-values obtained from the Wilcoxon signed-rank tests for each pair-wise comparison. With regard to the reduced Bayesian regression model, Table 8 has confirmed strong evidence of a significant difference from the three other models as indicated by the p-values close to 0.000. Hence, it is concluded that the reduced Bayesian regression model is not able to predict

Model	Reduced Bayesian regression		Reduced Bayesian MVN		Reduced Bayesian MVN-R	
	Ab.Res.	MRE	Ab.Res.	MRE	Ab.Res.	MRE
Reduced regression	0.000	0.000	0.015	0.011	0.048	0.086
Reduced Bayesian regression	-	-	0.000	0.000	0.000	0.000
Reduced Bayesian MVN	0.000	0.000	-	-	0.067	0.108
Reduced Bayesian MVN-R	0.000	0.000	0.067	0.108	-	-

Table 8
Significance test results on the differences in Table 7

effort as accurately as the three other models when only four systems have been used for calibration.

With regard to the reduced Bayesian MVN model, the tests have confirmed evidence of a significant difference from the reduced regression model as indicated by the p-values less than 0.05. Hence, it is concluded that the reduced Bayesian MVN model can predict effort better than the regression model when only four systems have been used for calibration.

With regard to the reduced Bayesian MVN-R model, the test has confirmed evidence of a significant difference in the absolute residuals from the reduced regression model as indicated by a p-value less than 0.05. Another test has also confirmed weak evidence in the MRE values as indicated by the p-value of 0.086, which is not far from 0.05. Hence, it is concluded that the reduced Bayesian MVN-R model can predict effort better than the regression model when only four systems have been used for calibration. In comparison with the reduced Bayesian MVN model, the difference in the absolute residuals is weakly significant as indicated by the p-value of 0.067, while the difference in the MRE values is not significant as indicated by the p-value of 0.108. Considering that the absolute residuals seem to be a better measure for model comparison than MRE according to other studies, it is concluded that the reduced Bayesian MVN-R model can predict effort better than the reduced Bayesian MVN model when only four systems have been used for calibration.

When summarizing the results from those pair-wise comparisons, it is also concluded that the Bayesian MVN-R model is the best among the four models when only four systems have been used for calibration.

7.5 Discussion

For the original dataset and the 75% and 50% subsets, the three Bayesian statistical effort prediction models have achieved predictive accuracy that satisfy the suggested criteria for an accurate model in the literature, with an only exception of the pred(0.25) of the full Bayesian regression model. Although those criteria are based on MRE and MRE is not always a reliable measure, we still consider those models' predictive accuracy are very good, certainly appropriate for practical use. For the 25% subsets, the Bayesian MVN and MVN-R models still have achieved good predictive accuracy. Those findings confirm our expectation that Bayesian statistical models can achieve good predictive accuracy even when only a very small amount of empirical data is used for calibration.

In comparison with the multiple linear regression model, the accuracy of the three Bayesian statistical models are equivalent in general. However, when the calibration subsets consist of only four software systems, the accuracy of the Bayesian MVN and MVN-R models are significantly better. This finding suggests that those two models, in particular the Bayesian MVN-R model would be a better choice in the target environment, particularly when only a very small amount of historical data is available for calibration.

8 Conclusions

8.1 Summary

Three new Bayesian statistical software development effort prediction models are constructed for database-oriented software systems, which are developed using a 4GL development tool suite, Oracle's *Designer 6i* and *Developer 6i*. They are a Bayesian regression model, a Bayesian multivariate normal distribution (MVN) model, and a model named Bayesian MVN-R since it is a hybrid of the previous two models. The models use four *specification-based* software size metrics and development team's productivity metric as predictor variables. The full models consist of all those five predictor variables, while the reduced models consist of only two, which are the total number of database entities accessed by all the data entry forms, and productivity metric. Both full and reduced models are first calibrated using the original dataset, which consists of data from 17 software systems developed in the target environment. Then, the reduced models are calibrated using smaller-sized subsets, which were randomly sampled from the original dataset, in order to examine if the predictive accuracy still remains good as the calibration subsets become smaller.

The models' predictive accuracy are evaluated using MRE, pred(0.25), pred(0.30) and the absolute residuals. The results show that each of the three Bayesian statistical models is able to predict software development effort in the target environment very successfully, and equally well to the other two models, as well as to the corresponding multiple linear regression model. In addition, the reduced models seem to be a better choice than the full models since they require less effort for data collection. However, when the calibration subsets become smaller than five systems, the Bayesian MVN and MVN-R models significantly outperform the Bayesian regression and multiple linear regression models. Overall, it is concluded that the reduced Bayesian MVN-R model is the best choice for predicting effort in the target environment. Those results confirm the authors' expectation that Bayesian statistical models can predict software development effort accurately in the target environment, even when only a very small amount of historical data is available for the models' calibration. Those results also should justify the use of Bayesian statistical models in software effort prediction and encourage other researchers to investigate the models further, so that the Bayesian statistical models' full potential in software effort prediction can be revealed and software organizations and practitioners can benefit from it.

8.2 Limitation of the study

All the models in this study were calibrated using the empirical data collected from software systems developed by non-professional, university undergraduate developers. This may imply that the applicability of the models to industrial settings would be limited. However, the authors consider the models still can achieve good predictive accuracy in industrial settings, provided that they are calibrated using historical data collected in the target setting. This is because our Bayesian statistical models' predictive accuracy mainly depend on the shapes of the probability distributions of the variables. And those shapes are expected to be not significantly different between an industrial setting and our setting, although the central tendency and variance could be different. However, the applicability of empirical effort prediction models is, in general, subject to the specific environment where the models' historical data were collected. Given that, the models presented in this paper would not be exempted from such limitations. This means that the models require separate calibration for each new environment, which includes choosing a new set of appropriate predictor variables when necessary.

Another limitation is that the results presented in this paper were obtained from a limited number of independent random samples drawn from a single dataset. This implies that the results would be different if a different number of samples were drawn and/or if more than one dataset were used. Although the authors are confident of the appropriateness of the statistical procedures performed in this study, our models would not be exempted from this limitation, either.

8.3 Future directions

One of the authors is currently investigating predictive accuracy of the Bayesian statistical models using the International Software Benchmarking Standards Group dataset, which contains data collected from a wide range of software systems developed in various industrial settings. The results from this investigation are expected to provide more information about predictive accuracy of the Bayesian statistical effort prediction models in general. The authors also expect more researchers to engage themselves in carrying out similar studies, since Bayesian statistical models can provide a significant benefit to software organizations and practitioners due to the better predictive accuracy than regression models, when the system of interest has only a very limited amount of historical data. The results from those future studies should reveal Bayesian statistical software effort prediction models' full potential and contribute toward the models taking a leading role among other effort prediction models.

References

[1] A.J. Albrecht and J.E. Gaffney. Software function, source lines of code, and development effort prediction: a software science validation. *IEEE Transactions on Software Engineering*, SE–9(6):639–648, 1983.

[2] J. Baik, B. Boehm, and B.M. Steece. Disaggregating and calibrating the CASE tool variable in COCOMO II. *IEEE Transactions on Software Engineering*, 28(11):1009–1022, 2002.

[3] B.W. Boehm. *Software Engineering Economics.* Prentice–Hall, 1981.

[4] S. Chulani, B. Boehm, and B.M. Steece. Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering*, 25(4):513–583, 1999.

[5] P. Congdon. *Bayesian Statistical Modelling.* John Wiley & Sons., 2001.

[6] S.D. Conte, H.E. Dunsmore, and V.Y. Shen. *Software Engineering Metrics and Models.* Benjamin/Cummings Publishing Company, 1986.

[7] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer–Verlag New York, 1999.

[8] J.J. Dolado. A study of the relationships among albrecht and mark ii function points, lines of code, 4gl and effort. *Journal of Systems Software*, 37:161–173, 1997.

[9] J.J. Dolado. A validation of the component-based method for software size estimation. *IEEE Transactions on Software Engineering*, 26(10):1006–1021, 2000.

[10] N. Fenton and M. Neil. A critique of software defect prediction models. *IEEE Transactions on Software Engineering*, 25(5):675–689, 1999.

[11] N.E. Fenton and S.L. Pfleeger. *Software Metrics:A Rigorous & Practical Approach.* PWS Publishing Company, second edition, 1997.

[12] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit. A simulation study of the model evaluation criterion mmre. *IEEE Transactions on Software Engineering*, 29(11):985–995, 2003.

[13] R.L. Glass. Frequently forgotten fundamental facts about software engineering. *IEEE Software*, May/June:110–112, 2001.

[14] P.J. Green. A primer on markov chain monte carlo. In O.E. Barndorff-Nielsen, D.R. Cox, and C. Klüppelberg, editors, *Complex Stochastic Systems*, chapter 1, pages 1–62. Chapman & Hall/CRC, 2001.

[15] F.V. Jensen. *Bayesian Networks and Decision Graphs.* Springer–Verlag New York, 2001.

[16] C.F. Kemerer. An empirical validation of software cost estimation models. *Communications of the ACM*, 30(5):416–429, 1987.

[17] B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, and M.J. Shepperd. What accuracy statistics really measure. *IEE Proceedings–Software*, 148(3):81–85, 2001.

[18] S.G. MacDonell. Establishing relationships between specification size and software process effort in case environment. *Information and Software Technology*, 39:35–45, 1997.

[19] M. Neil, N. Fenton, and L. Nielsen. Building large-scale bayesian networks. *The Knowledge Engineering Review*, 15(3):257–284, 2000.

[20] M.J. Shepperd, M. Cartwright, and G. Kadoda. On building prediction systems for software engineers. *Empirical Software Engineering*, 5:175–182, 2000.

[21] I. Stamelos, L. Angelis, P. Dimou, and E. Sakellaris. On the use of Bayesian belief networks for the prediction of software productivity. *Information and Software Technology*, 45:51–60, 2003.

[22] E. Stensrud, T. Foss, B.A. Kitchenham, and I. Myrtveit. An empirical validation of the relationship between the magnitude of relative error and project size. In *Proceedings of the 8th IEEE Symposium on Software Metrics (METRICS'02)*, pages 3–12, 2002.

[23] B. Stewart. Predicting project delivery rates using the Naive–Bayes classifier. *Journal of Software Maintenance and Evolution: Research and Practice*, 14:161–179, 2002.

[24] G. Tate and J.M. Verner. Approaches to measuring size of application products with case tools. *Information and Software Technology*, 33(9):622–628, 1991.

[25] C. van Koten and A.R. Gray. An application of bayesian network for predicting object-oriented software maintainability. *Information and Software Technology*, in press, 2005.

[26] J.M. Verner and G. Tate. A software size model. *IEEE Transactions on Software Engineering*, 18(4):265–278, 1992.