Calibrating Software Cost Models Using Bayesian Analysis

Sunita Chulani*, Barry Boehm, Bert Steece
University of Southern California
Henry Salvatori, Room 330
Los Angeles, CA 90089-0781
***sdevnani@sunset.usc.edu**

1 Abstract

The COCOMO II research effort started in 1994 with the aim of updating software cost estimation models, such as the 1981 COnstructive COst MOdel and its 1987 Ada update. Both the earlier models experienced difficulties in estimating software projects of the 90s due to challenges such as non-sequential and rapid-development process models; reuse-driven approaches involving commercial-off-the-shelf (COTS) packages, reengineering, applications composition, and application generation capabilities; object-oriented approaches supported by distributed middleware; software process maturity effects and process-driven quality estimation. The COCOMO II research effort aims at alleviating these problems and is concentrated on developing a model well-suited for the 1990s and then annually updating it for the forthcoming years.

The initial definition of the COCOMO II model and its rationale are described in [Boehm95]. The model uses Source Lines of Code and/or Function Points for the sizing parameter, adjusted for reuse and breakage; a set of 17 multiplicative effort multipliers and a set of 5 exponential scale factors. The first calibration was based on a dataset of 83 actual projects collected from Commercial, Aerospace, Government and FFRDC organizations using a 10% weighted average multiple regression approach. It was published in 1997 and became popular as COCOMO II.1997 [Clark98]. Since then, the COCOMO II database has grown to 161 datapoints. This paper focuses on the Bayesian approach used to calibrate the 1998 version of the model to 161 datapoints. It compares and contrasts the two successive versions, i.e. the 1997 and 1998 calibrations; and concludes that the Bayesian approach used in the 1998 calibration is better and more robust than the multiple regression approach used in the 1997 calibration.

Bayesian analysis is a mode of inductive reasoning that has been used in many scientific disciplines. A distinctive feature of the Bayesian approach is that it permits the investigator to use both sample (data) and prior (expert-judgement) information in a logically consistent manner in making inferences. This is done by using Bayes' theorem to produce a 'post-data' or posterior distribution for the model parameters. Using Bayes' theorem, prior (or initial) values are transformed to post-data views. This transformation can be viewed as a learning process. The posterior distribution is determined by the variances of the prior and sample information. If the variance of the prior information is smaller than the variance of the sampling information, then a higher weight is assigned to the prior information. On the other hand, if the variance of the sample information is smaller than the variance of the prior information, then a higher weight is assigned to the sample information causing the posterior estimate to be closer to the sample information.

We note that the predictive performance of the Bayesian approach (i.e. COCOMO II.1998) is significantly better than that of the multiple regression approach (i.e. COCOMO II.1997). COCOMO II.1998 gives predictions that are within 30% of the actuals 71% of the time where as COCOMO II.1997 gives predictions within 30% of the actuals only 52% of the time.

Although this paper gives a synopsis of the COCOMO II model structure, the reader is urged to read [Boehm95] to attain a better understanding of COCOMO II and the differences from its predecessors.

Keywords: Software estimation, Bayesian analysis, multiple regression, model calibration, prediction accuracy, empirical modeling, COCOMO, measurement, metrics, cost estimation, project management.

2 Calibration Techniques of Successive Versions of COCOMO II

This section discusses the two statistical approaches used to calibrate the successive versions of COCOMO II, namely the 'Multiple Regression approach and the Bayesian approach.

Multiple Regression Approach
Multiple Regression expresses the response (e.g. Person Months) as a linear function of k predictors (e.g. Source Lines of Code, etc.). This linear function is estimated from the data using the ordinary least squares approach discussed in numerous books such as [Judge93, Weisberg85]. A multiple regression model may be written as

$$y_t = \beta_0 + \beta_1 x_{t1} + ... + \beta_k x_{tk} + \varepsilon_t \qquad \text{Eq. 1}$$

where x_{t1} … x_{tk} are the values of the predictor (or regressor) variables for the t_{th} observation, β_0 . . . β_κ are the coefficients to be estimated, ε_t is the usual error term, and y_t is the response variable for the t_{th} observation.

The COCOMO II Post Architecture model has the following form

$$Effort = A \times [Size]^{1.01+\sum_{i=1}^{5} SF_i} \times \prod_{i=1}^{17} EM_i \qquad \text{Eq. 2}$$

where,
A = Multiplicative Constant
Size = Size of the software project measured in terms of KSLOC (thousands of Source Lines of Code [Park92], Function Points [IFPUG94] or Object Points [Banker92])
SF = Scale Factor
EM = Effort Multiplier [refer to Boehm81, Boehm95 for further explanation of COCOMO '81 and COCOMO II respectively]

The COCOMO II model equation, can be linearized by taking logarithms on both sides of the equation as shown:

$$\begin{aligned} \ln(PM) = \beta_0 + \beta_1 \cdot 1.01 \cdot \cdot \ln(Size) + \beta_2 \cdot SF_1 \cdot \ln(Size) + \Lambda + \beta_6 \cdot SF_5 \cdot \ln(Size) + \\ \beta_7 \cdot \ln(EM_1) + \beta_8 \cdot \ln(EM_2) + \Lambda + \beta_{22} \cdot \ln(EM_{16}) + \beta_{23} \cdot \ln(EM_{17}) \end{aligned} \qquad \text{Eq. 3}$$

The 1997 calibration used a dataset consisting of 83 completed projects [Clark98]. The regression estimated the β coefficients associated with the 5 scale factors and 17 effort multipliers; with some of the estimates being counter intuitive.

As an example, lets consider the 'Develop for Reuse' (RUSE) effort multiplier. This multiplicative parameter captures the additional effort required to develop components intended for reuse on current or future projects. As shown in table 1a, if the RUSE rating is Extra High (XH), i.e. developing for reuse across multiple product lines, it will cause an increase in effort by a factor of 1.56. On the other hand, if the RUSE rating is Low (L), i.e. developing with no consideration of future reuse, it will cause effort to decrease by a factor of 0.89. This rationale is consistent with the results of twelve published studies of the relative cost of developing for reuse compiled in [Poulin97]. But, the regression results produce a negative coefficient for the β coefficient associated with RUSE. This negative coefficient results in the counter intuitive rating scale shown in table 1b, i.e. an XH rating for RUSE causes a decrease in effort and a L rating causes an increase in effort. Note the opposite trends followed in tables 1a and 1b.

Table 1a: RUSE – Expert-determined a-priori rating scale; consistent with 12 published studies

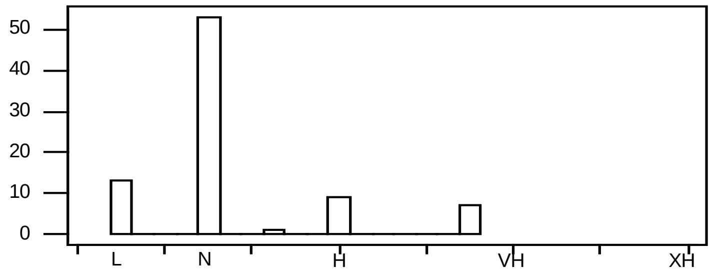
Develop for Reuse (RUSE)	Low (L)	Nominal (N)	High (H)	Very High (VH)	Extra High (XH)
Definition	None	Across project	Across program	Across product line	Across multiple product lines
1997 A-priori Values	0.89	1.00	1.16	1.34	1.56

Table 1b: RUSE – Data-determined rating scale; contradicting 12 published studies

Develop for Reuse (RUSE)	Low (L)	Nominal (N)	High (H)	Very High (VH)	Extra High (XH)
Definition	None	Across project	Across program	Across product line	Across multiple product lines
1997 Data-Determined Values	1.05	1.00	0.94	0.88	0.82

A possible explanation [discussed in a study by Mullet76 on "Why regression coefficients have the wrong sign"] for this contradiction may be the distribution of the data used to calibrate RUSE. Unfortunately, the data collection effort resulted in several responses such as, "I don't know" or "It does not apply", being recorded as 'Nominal' (since a rating of 'Nominal' on a multiplicative cost driver is always coded as 1.0 causing it to have no impact on effort). A reason for the lack of clarity of RUSE is that RUSE is a relatively new cost factor and the respondents did not have enough information to report its rating accurately during the data collection process. Hence, the data does not exhibit enough dispersion along the entire range of possible values for RUSE. Note (in figure 1) that a little over 50 of the 83 datapoints have RUSE = Nominal and we had no observations with RUSE = XH.

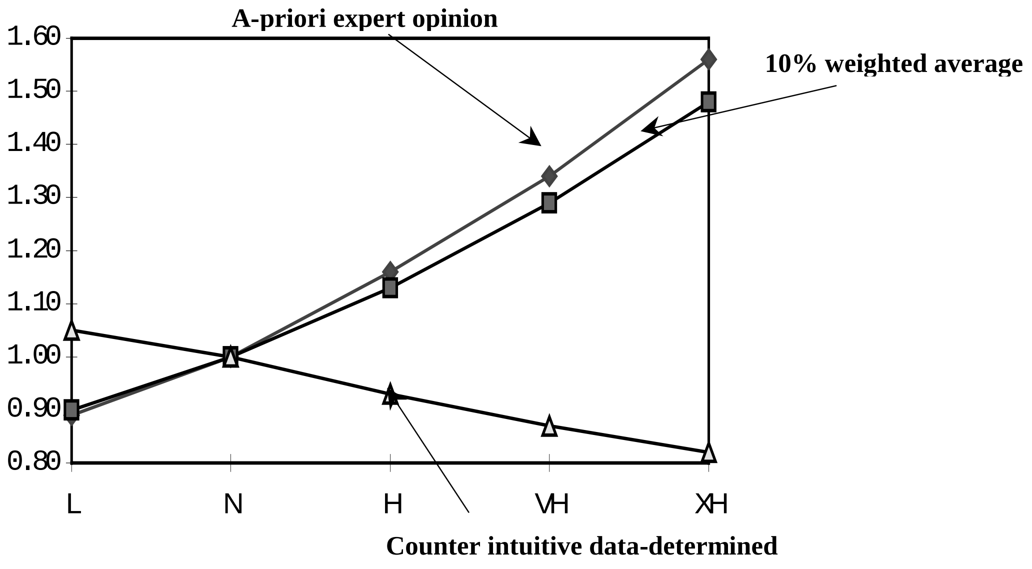
Figure 1: Distribution of RUSE



Other reasons for the counter intuitive results include the violation of some of the restrictions imposed by multiple regression [Briand92, Chulani98]:
(i) The number of datapoints should be large relative to the number of model parameters (i.e. there are many degrees of freedom). Unfortunately, collecting data has and continues to be one of the biggest challenges in the software estimation field. This is caused primarily by immature processes and management reluctance to release cost-related data.
(ii) There are no outliers. Extreme cases frequently occur in software engineering data because there is lack of precision in the data collection process.
(iii) The predictor variables (cost drivers and scale factors) are not highly correlated. Unfortunately, because cost data is historically rather than experimentally collected, correlations among the predictor variables are unavoidable.

The above restrictions are violated to some extent by the COCOMO II dataset. To resolve some of the counter intuitive results produced by the regression analysis (e.g. the negative coefficient for RUSE as explained above), we used a weighted average of the expert-judgement results and the regression results, with only 10% of the weight going to the regression results. We selected the 10% weighting factor because models with 40% and 25% weighting factors produced less accurate predictions. This pragmatic calibrating procedure moved the model parameters in the direction suggested by the sample data but retained the rationale contained within the apriori values. An example of the 10% application using the RUSE effort multiplier is given in figure 2. As shown in the graph, the trends followed by the a-priori and the data-determined curves are opposite. The data-determined curve has a negative slope and as shown above in table 1, it violates expert opinion, whose curve has a positive slope.

Figure 2: Example of the 10% weighted average approach: RUSE Rating Scale



The resulting calibration of the COCOMO II model using the 1997 dataset of 83 projects produced estimates within 30% of the actuals 52% of the time for effort. The prediction accuracy improved to 64% when the data was stratified into sets based on the source of the data. The constant, A, of the COCOMO II equation was recalibrated for each of these sets resulting in an improvement in prediction accuracy results.

Table 2: Prediction Accuracy of COCOMO II.1997

COCOMO II.1997	**Before Stratification by Organization**	**After Stratification by Organization**
PRED(.20)	46%	49%
PRED(.25)	49%	55%
PRED(.30)	52%	64%

While the 10% weighted average procedure produced a workable initial model, it is desirable to have a more formal framework for combining expert judgement and sample information. A Bayesian analysis with an informative prior provides such a framework.

Bayesian Approach: Basic Framework - Terminology and Theory

The Bayesian approach provides a formal process by which a-priori expert-judgement can be combined with sampling information (data) to produce a robust a-posteriori model. Using Bayes' theorem, we can combine our two information sources as follows:

$$f(\beta|Y) = \frac{f(Y|\beta)\, f(\beta)}{f(Y)} \qquad \text{Eq. 4}$$

where ß is the vector of parameters in which we are interested and Y is the vector of sample observations from the joint density function $f(\beta|Y)$. In equation 4, $f(\beta|Y)$ is the posterior density function for ß summarizing all the information about ß, $f(Y|\beta)$ is the sample information and is algebraically equivalent to the likelihood function for ß, and $f(\beta)$ is the prior information summarizing the expert-judgement information about ß. Equation 4 can be rewritten as

$$f(\beta|Y) \propto l(\beta|Y)\, f(\beta) \qquad \text{Eq. 5}$$

In words, equation 5 means

$$Posterior \propto Sample \times Prior$$

In the Bayesian analysis context, the "prior" probabilities are the simple "unconditional" probabilities to the sample information; while the "posterior" probabilities are the "conditional" probabilities given sample and prior information.

The Bayesian approach makes use of prior information that is not part of the sample data by providing an optimal combination of the two sources of information. As described in many books on Bayesian analysis [Leamer78, Box73], the posterior mean, b**, and variance, Var(b**), are defined as

$$b^{**} = [\frac{1}{s^2} X'X + H^*]^{-1} \times \left[\frac{1}{s^2} X'Xb + H^* b^*\right] \text{ and } Var(b^{**}) = \left[\frac{1}{s^2} X'X + H^*\right]^{-1} \quad \text{Eq. 6}$$

where X is the matrix of predictor variables, s is the variance of the residual for the sample data; and H* and b* are the precision (inverse of variance) and mean of the prior information respectively.

From equation 6, it is clear that in order to determine the Bayesian posterior mean and variance, we need to determine the mean and precision of the prior information and the sampling information. The next two subsections describe the approach taken to determine the prior and sampling information, followed by a subsection on the Bayesian a-posteriori model.

Bayesian Approach: Prior Information

To determine the prior information for the coefficients (i.e. b* and H*), one of the authors conducted a Delphi exercise [Helmer66]. Eight experts from the field of software estimation were asked to independently provide their estimate of the numeric values associated with each COCOMO II cost driver. The author summarized the results of the first round and distributed them back to the participants to get further consensus. Each of the participants got a second opportunity to independently refine his/her response based on the responses of the rest of the participants in round 1. The a-priori mean and variance for each of the 22 parameters were computed from the round 2 results.

Table 3 provides the a-priori set of values for an example effort multiplier, Required Software Reliability, RELY. This multiplicative parameter is the measure of the extent to which the software should perform its intended functions satisfactorily over its execution time. If the effect of a failure is simply slight inconvenience, then the RELY rating is set to Very Low (VL) resulting in a decrease in effort by a factor of 0.75. On the other hand, if the effect of a software failure can be the loss of human life, then the corresponding RELY rating is set to Very High (VH) resulting in an increase in effort by a factor of 1.41. The resulting range of productivity for RELY is 1.91 where a change in the rating from VL to VH will cause an increase in effort by a factor of 1.91.

Table 3: COCOMO II.1998 "A-Priori" Rating Scale for Required Software Reliability (RELY)

Required Software Reliability (RELY)	**Productivity Range**	**Very Low**	**Low**	**Nominal**	**High**	**Very High**
Definition	Highest Rating/ Lowest Rating	Slight inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High financia l loss	Risk to human life
1998 A-priori Values	1.41/0.74 = 1.91	0.74	0.88	1.0	1.16	1.41

Similarly the a-priori rating scale for each of the COCOMO II parameters was determined by the outcome of the 2-round Delphi.

Bayesian Approach: Sample Information

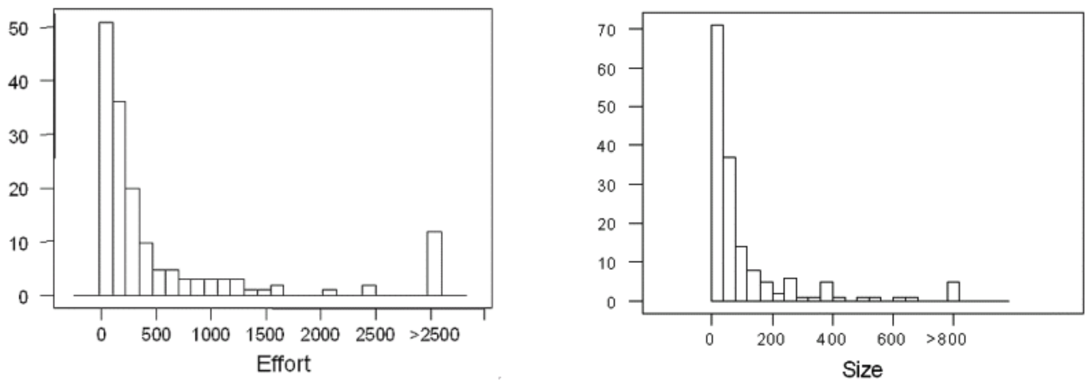
The sampling information is the result of a data collection activity initiated in September 1994, soon after the initial publication of the COCOMO II description [Boehm95]. Affiliates of the Center for Software Engineering at the University of Southern California provided most of the data. These organizations represent the Commercial, Aerospace, and FFRDC (Federally Funded Research and Development Center) sectors of software development.

Data of completed software projects is recorded on a data collection form that asks between 33 and 59 questions depending on the degree of source code reuse [USC-CSE94]. A question asked very frequently is the definition of software size, i.e., what defines a line of source code? Appendix B in the Model Definition Manual [USC-CSE97] defines a logical line of code using the framework described in [Park92]. In spite of the definitions, the data collected to date exhibits local variations caused by differing interpretations of the counting rules. These variations result in local calibration producing better estimates. The local calibration results are discussed in the following subsection (see table 4).

The data collected includes the actual size in KSLOC (thousands of Source Lines of Code adjusted for breakage and reuse) which is a predictor variable; and the actual effort in PM (Person Months; 1 PM = 152 hours), which is the response variable. The database has grown from 83 datapoints in 1997 to 161 datapoints in 1998.

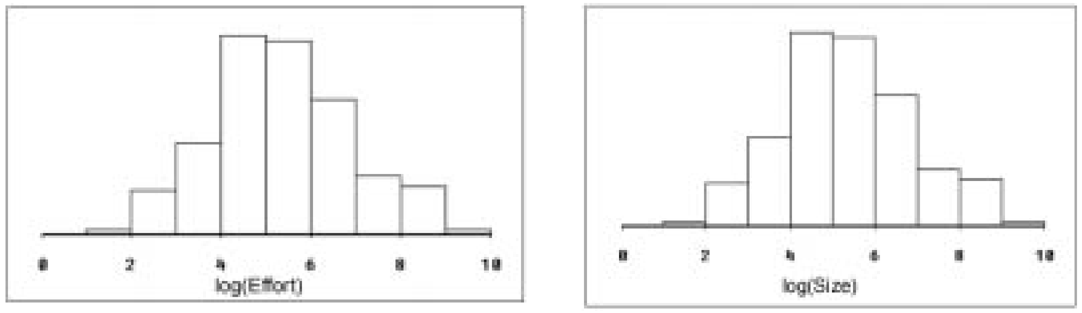
The distributions of effort and size for the 1998 database of 161 datapoints are as shown in figure 3. It was important to check the distributions to ensure that the log transformations (taken to go from eq. 2 to eq. 3) are indeed valid and they yield the normality essential for multiple regression.

Figure 3: Distribution of Effort and Size: 1998 dataset of 161 observations



As can be noted, both the histograms are positively skewed with the bulk of the projects in the database with effort less than 500 PM and size less than 150 KSLOC. Since the multiple regression approach based on least squares estimation assumes that the sample is from a multivariate normal population, the positively skewed histograms indicate the need for a transformation. As depicted in figure 4, log transformations achieve the desired normality in the data. This is consistent with the transformations we took to go from eq. 2 to eq. 3.

Figure 4: Distribution of log transformed Effort and Size: 1998 dataset of 161 observations



The regression analysis done in RCode (statistical software developed at University of Minnesota, [Cook94]) on the log transformed COCOMO II parameters using a dataset of 161 datapoints yield the following results:

```
Data set = COCOMOII.1998
Response      = log[PM]
Coefficient Estimates
Label             Estimate         Std. Error    t-value
Constant_A        0.961552         0.103346        9.304
log[SIZE]         0.921827         0.0460578      20.015
PMAT*log[SIZE]    0.684836         0.481078        1.424
PREC*log[SIZE]    1.10203          0.373961        2.947
TEAM*log[SIZE]    0.323318         0.497475        0.650
FLEX*log[SIZE]    0.354658         0.686944        0.516
RESL*log[SIZE]    1.32890          0.637678        2.084
log[PCAP]         1.20332          0.307956        3.907
log[RELY]         0.641228         0.246435        2.602
log[CPLX]         1.03515          0.232735        4.448
log[TIME]         1.58101          0.385646        4.100
log[STOR]         0.784218         0.352459        2.225
log[ACAP]         0.926205         0.272413        3.400
log[PEXP]         0.755345         0.356509        2.119
log[LTEX]         0.171569         0.416269        0.412
log[DATA]         0.783232         0.218376        3.587
log[RUSE]        -0.339964         0.286225       -1.188
log[DOCU]         2.05772          0.622163        3.307
log[PVOL]         0.867162         0.227311        3.815
log[AEXP]         0.137859         0.330482        0.417
log[PCON]         0.488392         0.322021        1.517
log[TOOL]         0.551063         0.221514        2.488
log[SITE]         0.674702         0.498431        1.354
log[SCED]         1.11858          0.275329        4.063
```

The above results provide the estimates for the β coefficients associated with each of the predictor variables (see eq. 3). The t-value (ratio between the estimate and corresponding standard error; where standard error is the square root of the variance) may be interpreted as the signal-to-noise ratio associated with the corresponding predictor variables. Hence, the higher the t-value, the stronger the signal (i.e. statistical significance) being sent by the predictor variable.

While the regression produced intuitively reasonable estimates for most of the predictor variables; the negative coefficient estimate for RUSE (as discussed earlier) and the magnitudes for the coefficients on AEXP (Applications Experience), LTEX (Language and Tool Experience), FLEX (Development Flexibilit), and TEAM (Team Cohesion), violate our prior opinion about the impact of these parameters on Effort (i.e. PM). The quality of the data probably explains some of the conflicts between the prior information and sample data.

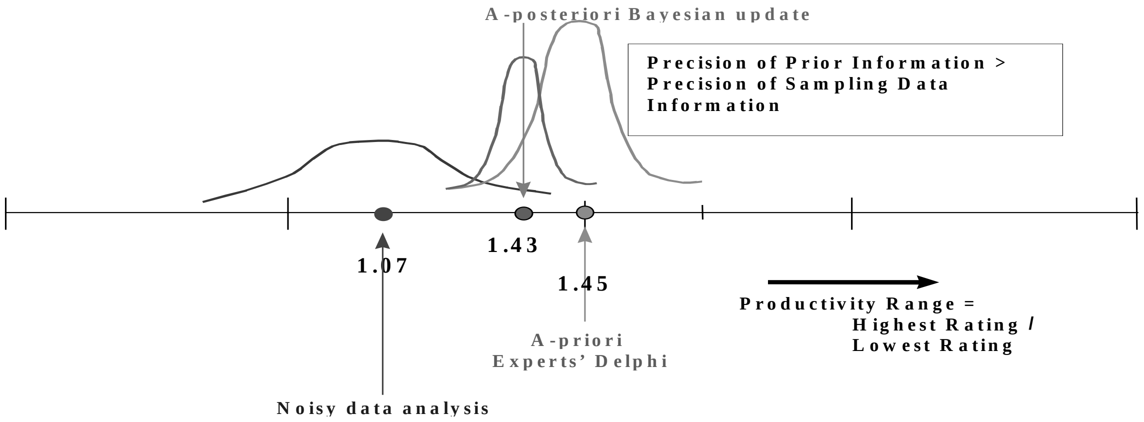
Bayesian Approach: Combining Prior and Sampling Information

As a means of resolving the above conflicts, we will now use the Bayesian paradigm as a means of formally combining prior expert judgment with our sample data.

Equation 6 reveals that if the precision of the a-priori information (H*) is bigger (or the variance of the a-priori information is smaller) than the precision (or the variance) of the sampling information $\left(1/s^2\ X'X\right)$, the posterior values will be closer to the a-priori values. This situation can arise when the gathered data is noisy as depicted in figure 5 for an example cost factor, Language and Tool Experience. Figure 5 illustrates that the degree-of-belief in the prior information is higher than the degree-of-belief in the sample data. As a consequence, a stronger weight is assigned to the prior information causing the posterior mean to be closer to the prior mean. On the other hand (not illustrated), if the precision of the sampling information

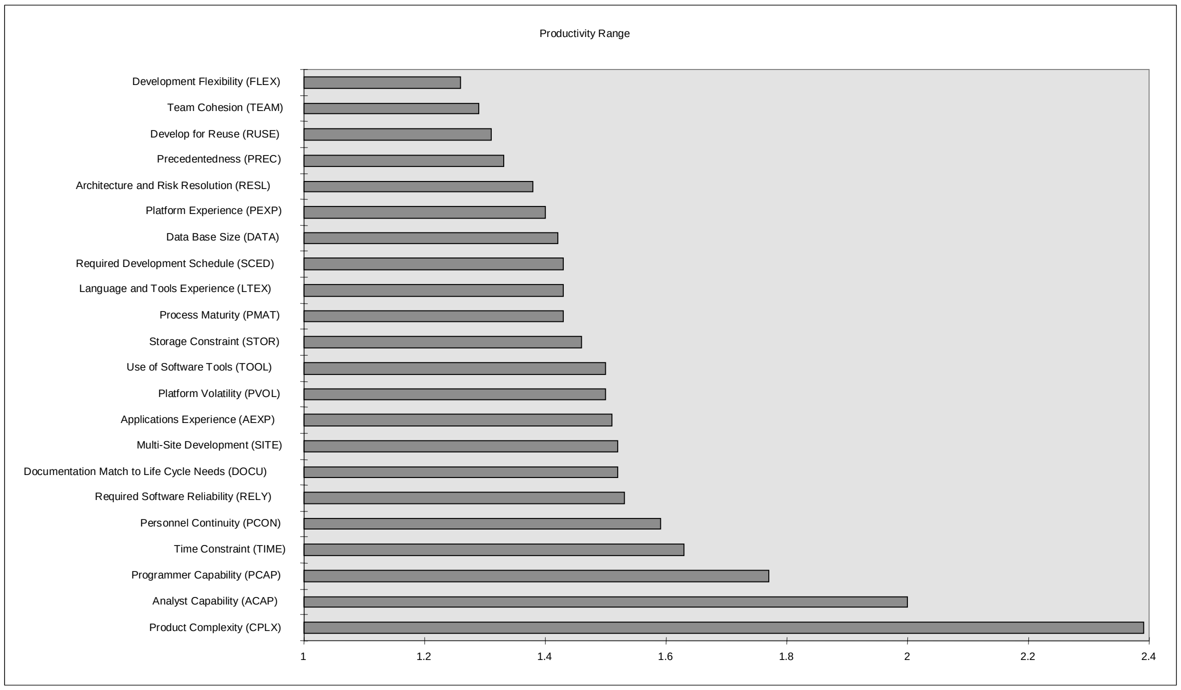
$(1/s^2\ X'X)$ is larger than the precision of the prior information (H*), then a higher weight is assigned to the sampling information causing the posterior mean to be closer to the mean of the sampling data. The resulting posterior precision will always be higher than the a-priori precision or the sample data precision.

Figure 5: A-Posteriori Bayesian Update in the Presence of Noisy Data (Language and Tool Experience, LTEX)


A-posteriori Bayesian update
Precision of Prior Information > Precision of Sampling Data Information
1.43
1.07
1.45
Productivity Range = Highest Rating / Lowest Rating
A-priori Experts' Delphi
Noisy data analysis

The complete Bayesian analysis on COCOMO II yields the Productivity Ranges (ratio between the least productive parameter rating, i.e. the highest rating, and the most productive parameter rating, i.e. the lowest rating) illustrated in figure 6. Figure 6 gives an overall perspective of the relative Software Productivity Ranges (PRs) provided by the COCOMO II parameters. The PRs provide insight on identifying the high payoff areas to focus on in a software productivity improvement activity. For example, CPLX (Product Complexity) is the highest payoff parameter and FLEX (Development Flexibility) is the lowest payoff parameter in the Bayesian Calibration of the COCOMO II model.

Figure 6: Bayesian A-Posteriori Productivity Ranges



The resulting COCOMO II.1998 model calibrated to 161 datapoints produces estimates within 30% of the actuals 71% of the time for effort. If the model's multiplicative coefficient is calibrated to each of the major sources of project data, the resulting model produces estimates within 30% of the actuals 76% of the time. It is therefore recommended that organizations using the model calibrate it using their own data to increase model accuracy and produce a local optimum estimate for similar type projects. From table 4 it is clear that the prediction accuracy of the COCOMO II.1998 model is better than the prediction accuracy of the COCOMO II.1997 model.

Table 4: Prediction Accuracies of COCOMO II.1997 and COCOMO II.1998

COCOMO II	Prediction Accuracy	Before Stratification	After Stratification
1997	PRED(.20)	46%	49%
	PRED(.25)	49%	55%
	PRED(.30)	52%	64%
1998	PRED(.20)	56%	66%
	PRED(.25)	65%	74%
	PRED(.30)	71%	76%

Cross Validation of the COCOMO II.1998 Calibrated Model

The COCOMO II.1998 Bayesian calibration discussed above uses the complete dataset of 161 datapoints. Thus, the prediction accuracies of COCOMO II.1998 (depicted in table 4) are based on the same dataset of 161 datapoints. That is, the calibration and validation datasets are the same. A natural question that arises in this context is how well will the model predict new software development projects? To address this issue, we randomly selected 121 observations for our calibration dataset with the remaining 40 becoming assigned to the validation dataset (i.e. " new" data). We repeated this process 15 times creating 15 calibration and 15 validation datasets each of size 121 and 40 respectively.

We then developed a prediction equation for each of the 15 calibration datasets. We used the resulting a-posteriori models to predict the development effort of the 40 "new" projects in the validation datasets. This validation approach, known as out-of-sample validation, provides a true measure of the model's predictive abilities. This out-of-sample test yielded an average PRED(.30) of 69%; indicating that on average, the out-of-sample validation results produced estimates within 30% of the actuals 69% of the time. Hence, we conclude that our Bayesian model has reasonably good predictive qualities.

3 Conclusions

The Bayesian approach discussed in this paper resolves one of the biggest problems faced by the software engineering community; the challenge of making good decisions using data that is usually scarce and incomplete. Most of the current software cost estimation models, including COCOMOII.1997, are calibrated using some form of the Multiple Regression approach, which doesn't resolve this and other problems of existing software engineering data. The Bayesian approach alleviates this problem by making use of experience-based expert judgement data along with sampling information in a logically consistent manner.

The Bayesian estimating procedure used to calibrate COCOMO II.1998 incorporated a formal process to merge prior information with existing software engineering data. In many models, such prior information is informally used to evaluate the "appropriateness" of results. An important aspect of formalizing the use of prior information is that when others know what prior production functions are being used they can repeat the calibration calculations (or can incorporate different prior information in a similar way).

Using the prior information obtained from a 2-Round Delphi (a consensus of experts in the software estimation field) and the sampling information obtained from the COCOMO II dataset of 161 observations, the Bayesian approach produced estimates within 30% of the actuals 71% of the time for effort. Cross validation of the posterior model on 15 randomly selected datasets yielded comparable results. This accuracy was further improved by locally calibrating the model's multiplicative coefficient to produce estimates within 30% of the actuals 76% of the time. It is therefore recommended that organizations using the model calibrate it using their own data to increase model accuracy and produce a local optimum estimate for similar type projects.

4 References

Banker92 - "An Empirical Test of Object-based Output Measurement Metrics in a Computer Aided Software Engineering (CASE) Environment", Rajiv D. Banker, Robert J. Kauffman and Rachna Kumar, Journal of Management Information Systems, Vo1. 8, No. 3, 1992.

Boehm81 - Software Engineering Economics, Barry W. Boehm, Prentice-Hall, 1981.

Boehm95 - "Cost Models for Future Software Live Cycle Processes: COCOMO 2.0", Annals of Software Engineering Special Volume on Software Process and Product Measurement, J.D. Arthur and S.M. Henry (Eds.) , J.C. Baltzer AG, Science Publishers, Amsterdam, The Netherlands, Vol 1, 1995, pp. 45-60.

Box73 - Bayesian Inference in Statistical Analysis, George Box and George Tiao, Addison Wesley, 1973.

Briand92 - "A Pattern Recognition Approach for Software Engineering Data Analysis", Lionel C. Briand, Victor R. Basili and William M. Thomas, IEEE Transactions on Software Engineering, Vol. 18, No. 11, November 1992.

Chulani98 - "Incorporating Bayesian Analysis to Improve the Accuracy of COCOMO II and Its Quality Model Extension", Sunita Chulani, Qualifying Exam Report, Computer Science Department, USC Center for Software Engineering, February 1998.

Clark98 - "Calibration Results of COCOMOII.1997", Brad Clark, Sunita Chulani, Barry Boehm International Conference on Software Engineering, April 1998.

Cook94 - An Introduction to Regression Graphics, Dennis Cook and Sanford Weisberg, Wiley Series, 1994.

Helmer66 - Social Technology, Basic Books, New York, O. Helmer, 1966.

IFPUG94 - International Function Point Users Group (IFPUG), Function Point Counting practices Manual, Release 4.0, 1994.

Judge93 - Learning and Practicing Econometrics, George G. Judge, William Griffiths, R. Carter Hill, Wiley, 1993.

Leamer78 - Specification Searches, Ad hoc Inference with Nonexperimental Data. Edward E. Leamer, Wiley Series, 1978.

Mullet76 - "Why regression coefficients have the wrong sign", G. M. Mullet, Journal of Quality Technology, 1976.

Park92- "Software Size Measurement: A Framework for Counting Source Statements", R.M. Park et al, CMU-SEI-92-TR-20, Software Engineering Institute, Pittsburg, PA, 1992.

Poulin97 - Measuring Software Reuse, Principles, Practices and Economic Models, Jeffrey S. Poulin, Addison Wesley, 1997.

USC-CSE94 - Center for Software Engineering , "COCOMO II Cost Estimation Questionnaire," Computer Science Department, USC Center for Software Engineering, http://sunset.usc.edu/Cocomo.html, 1997.

USC-CSE97 - Center for Software Engineering , "COCOMO II Model Definition Manual," Computer Science Department, USC Center for Software Engineering, http://sunset.usc.edu/Cocomo.html, 1997.

Weisberg85 - Applied Linear Regression, Weisberg, S., 2nd Ed., John Wiley and Sons, New York, N.Y., 1985.

5 Acknowledgements

This work is being supported both financially and technically under the AFRL Contract F30602-96-C-0274, "KBSA Life Cycle Evaluation" and by the COCOMO II Program Affiliates: Aerospace, Air Force Cost Analysis Agency, Allied Signal, AT&T, Bellcore, EDS, Raytheon E-Systems, GDE Systems, Hughes, IDA, JPL, Litton, Lockheed Martin, Loral, MCC, MDAC, Motorola, Northrop Grumman, Rational, Rockwell, SAIC, SEI, SPC, Sun, TI, TRW, USAF Rome Lab, US Army Research Labs, Xerox.