ARPN Journal of Science and Technology

http://www.ejournalofscience.org

A Novel Imputation Technique for Software Effort Estimation

[1] Suresh Joseph. K, [2] Dr. Ravichandran.T
[1]Assistant Professor, Department of Computer Science,
School of Engineering & Technology, Pondicherry University, Puducherry, India.
[2]Principal, Hindusthan Institute of Technology, Coimbatore, India.
[1] ksjoseph.csc@gmail.com, [2] dr.travichandran@gmail.com

ABSTRACT

Statistical analysis is greatly hindered with missing information. It represents a loss of key data, but worse, it can introduce biased results in the analysis. A way to rectify the problem of missing data is to employ a sound method of imputation, a way to replace missing values with reasonable estimates. Their exists variety of estimation models like SLIM, COCOMO and other models like machine learning models which require accurate inputs for estimation of effort. It is very difficult to obtain such accurate information at the early development stage of a software project. These issues led to the introduction of variety of techniques to solve the relevant issues. The proposed approach is a hybrid approach which works based on Genetic algorithm and empirical estimation model COCOMO - II.

Keywords: *Imputation, Effort Estimation, COCOMO II, Software Engineering, Genetic Approach*

1. INTRODUCTION

Imputation is the replacement of missing values in data with estimation techniques and assumptions. The major problem that hampers useful application of imputation methods is bias, when an estimator's long-run average (expectation) differs from the quantity being estimated. The deviation becomes a danger. When this difference is systematical, the results of analyses may be biased and false conclusions are easily drawn." (Huisman 2000) Definitely, an imputation method will be plausible and consistent, reduce bias while preserving the relationship between items within the data, and can be evaluated for bias and precision (Sande 1982). Missing data are exasperation to a certain extent. Imputation is one of the key strategies that researchers employ to fill in missing data in a given dataset. Solving such issues is very intricate to put into operation and it is problem specific. Imputed data is used in order to find a suitable replacement for better accurate analyses. Imputation is a method of modifying for lost information. Missing reactions to information articles is a normal situation in any survey. This missingness regularly happens since the respondent denies or is unable to furnish information for a particular field or many and also because of typographical mistakes. Missing information may likewise effect from mis-keying or by editing process.

a. Whether imputation is a better model?

Definitely imputation model is a compatible one and is affluent enough which confirms based on the associations and relationships among the other variables in the data set. Imputation imposes models like probability model, Decision tree models, neural models on the complete data set and also other models.

This paper provides a performance comparison of COCOMO and proposed hybrid method for handling missing values in effort estimation. The paper is organized as the follows. Section II, describes the related work on effort estimation; Section III. Approach to software estimation; Section IV. Explores the Evaluation and performance analysis used in this work and Result discussions there on; Section V concludes the outcome of the proposed research.

2. RELATED WORK

a. Effort Estimation

The success in Engineering and evaluation of Software depends only when the stages of the SDLC are completed on time. One of the crucial tasks associated with Software Project Management is estimating cost estimation. Now-a-days software development has become competitive and software developers struggle because of many reasons like delivering product on time at the specified cost ensuring desirable quality. This helps us to understand the importance of estimating effort, schedule in early stages of development of the software. It cannot be concluded that exact value of effort cannot provide accurate schedules though both are closely related with each other. Schedules may slip when vague Requirements / Specifications are to be executed; i.e. when clients make unanticipated change, improper

training and personnel availability, sometimes mistakes can happen in initial stages if not corrected in the beginning.

b. Classification

Analogue estimation models – Formal
Estimation model
Expert estimation
Empirical parametric estimation models
Empirical non-parametric estimation models
Combination based estimation

c. Effort Estimation By Analogy

This method uses information from similar projects and derive estimate based on the past features of data, because it is a similar form of reckoning as human problem solving capability (Angelis and Stamelos, 2000; Leung, 2002). This method finds the similarities between projects it can be used in the early phase of the project. It is an easy way of finding effort of the project. The major problem with existing analogy-based technique is limited because of their inability to handle with non-quantitative data and missing values. Analogy based method shows best results in 60% of the cases and 30% of the cases fail because of worst predicted accuracy, hence suggesting some instability (Ruhe et al., 2003). It can be a promising in majority cases but fails in certain cases.

The similarities between functions are identified by Euclidean similarity (ES) and Manhattan similarity (MS) (Sheppard and Schofield, 1997). The similarity computation is done as follows

$$\mathrm{Sim}(p,p\prime) = 1/\left[\sqrt{\sum_{i=1}^{n} w_i \mathrm{Dis}(f_i,f\prime_i)} + \delta\right] \quad \delta = 0.0001$$

$$\mathrm{Dis}(f_i,f_i') = \begin{cases} (f_i - f_i')^2, & \text{if } f_i \text{ and } f_i' \text{ are numeric or ordinal} \\ 1 & \text{if } f_i \text{ and } f_i' \text{ are nominal and } f_i = f_i' \\ 0 & \text{if } f_i \text{ and } f_i' \text{ are nominal and } f_i \neq f_i' \end{cases}$$

Where ‘p’ and ‘p’’ are the projects, ‘WI’ is the weight assigned to each feature and it can vary between 0 and 1. ‘Fi’ and ‘fi’’ displays the ‘ith’ feature of a project and ‘n’ demonstrates the number of features. ‘δ’ is used to obtain non zero results. The MS formula is similar to the ES but it does computation by finding the absolute difference between the features. The similarity function is as follows.

$$\mathrm{Sim}(p,p\prime) = 1/\left[\sum_{i=1}^{n} w_i \mathrm{Dis}(f_i,f_i') + \delta\right] \quad \delta = 0.0001$$

$$\mathrm{Dis}(f_i,f_i') = \begin{cases} |f_i - f_i'| & \text{if } f_i \text{ and } f_i' \text{ are numeric or ordinal} \\ 1 & \text{if } f_i \text{ and } f_i' \text{ are nominal and } f_i = f_i' \\ 0 & \text{if } f_i \text{ and } f_i' \text{ are nominal and } f_i \neq f_i' \end{cases}$$

After selecting the K most similar projects, it is made possible to estimate the effort and cost of the new project as per the feature selected. The most common solution functions are: the closest analogy as most similar project (Walkerden and Jeffery, 1999), the average of most similar projects (Sheppard and Schofield, 1997), the median of most similar projects (Angelis and Stimulus, 2000) and the inverse distance weighted mean (Kadoda et al., 2000). The mean describes the average of the effort of K most similar projects, where K > 1. The median describes the median of the effort of K most similar projects, where K > 2. The inverse distance weighted mean adjusts the portion of each project in estimation by using the Equation.

$$\widehat{C}_p = \sum_{k=1}^{K} \frac{\mathrm{Sim}(p,p_k)}{\sum_{i=1}^{n} \mathrm{Sim}(p,p_k)} C_{p_k}$$

Where p shows the new project, pk illustrates the k^{th} most similar project, Cpk is the effort value of the k^{th} most similar project pk, Sim (p, pk) is the similarity between projects pk and p and K is the total number of most similar projects.

d. Expert Estimation Techniques

This estimation schema was developed by Barry Boehm in 70s (Software Engineering Economics, Boehm), each group member in the estimation team provides an anonymous estimate, the coordinator assesses the estimates given by the team and combines the estimate, If the estimates are crazy reassessment is done finally effort is computed.

e. SEER

SEER (System Evaluation and Estimation of Resources) is a proprietary model owned by Galorath Associates, Inc. In 1988, Galorath Incorporated this work on the original version of SEER-SEM which resulted in an initial solution of 22,000 LOC. SEER is an algorithmic software project management application planned specifically for effort estimation. This model is relied on the initial effort of Dr. Randall Jensen. The mathematical equations that are used in SEER were not available to the public, but Dr. Jensen in his writings made the basic equations available for review. The basic equation of Dr. Jensen is also referred as "software equation".

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

$$S_e = C_{te}(Kt_d)^{0.5}$$

Where, 'S' is the effective LOC, 'ct' is the effective developer technology constant, 'k' is the total life cycle cost in man-years, and 'td' is the development time in years.

f. COCOMO

COCOMO (Constructive Cost Model) is an empirical estimation scheme proposed in 1981. It is a model for estimating effort, cost, and schedule for development of software projects. This parametric model certainly provides better results. Barry Boehm definition towards his model "Basic COCOMO is good for rough order of magnitude estimates of software costs, but its accuracy is necessarily limited because of its lack of factors to account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and other project attributes known to have a significant influence on costs."

3. APPROACH TO ESTIMATION

To derive effort estimates and time schedule required to develop a software system; if the size of the system is 'S' and 'E' is effort then

$$F(S) = E \text{ and } E= (a+bS^c)\, m(x)$$

m(x) is the multipliers (The factors which influences the effort), a, b and c are constants. Based on this relationship COCOMO model has been built.

a. COCOMO II

This unique COCOMO Model has now been super-ceded by COCOMO II. In COCOMO II effort is expressed as Person-Month or the amount of time a person spends working on the software project for one month.

$$PM = a * Size^b * \prod_{i=1}^{15} EM_i$$

Where "a" and "b" are the domain constants in the COCOMO model. It has 15 effort multipliers. This estimation scheme accounts the experience and data of the past projects, which is extremely complex to understand and apply the same.

$$E = B + 0.01 * \sum_{j=1}^{5} SF_j$$

Hence, in principle, the effort is calculated by multiplying the estimation variable with the constant 'a' in the first stage, and some effort can be added with or deducted from the calculated effort at the second stage. In addition to that, each user should calibrate the model and the attribute values in accordance to their own historical projects data, which will reflect local circumstances that greatly influences accuracy of the model. From these viewpoint, whenever using the algorithmic effort estimation models, it is preferred that the impacts of cost drives have to be quantified and assessed in a proper way. Since the significance of the vagueness and uncertainty features that are inhabited in the effort drives due to the cognitive judgments are less, this Imputation approach can be preferred and applied to change the estimation scheme of the COCOMO II, which can substitute values in the place of vague information.

Recently many machine learning algorithms were proposed and used as the imputation techniques for estimating the effort. In majority of the cases we find that the imputation models posses' significance compared to other models in terms of performance. In this series, this research proposes a hybrid model of imputation technique, which holds better significance than other recent models. This is a comprehensive model which uses Graphical and Genetic Approaches. Genetic Algorithms (GA) are stochastic especially in domains where a direct search method cannot reveal better outputs and of course in the large search spaces. They have an advantage to operate with incomplete data to extract significant rules. This proposed hybrid effort estimation model produces accurate outputs.

It is observed that the accuracy of the COCOMO II is relied on three attributes; the value of estimation variable, the overall value of the effort multipliers and their impact in the estimation scheme. Here the estimation variable is the primary attribute and standard methods available for estimating the estimation variables (e.g. LOC, FP count, etc…)[7]. It is simple to estimate the overall value of the effort multipliers after assigning the appropriate values as per the requirements. But the complicated issue is to estimate impact of the effort multipliers, which plays a major role in the estimation scheme and causes for overestimation or underestimation of the software development effort. In this view, this work is aimed at refining the cost drivers and scale factors handling mechanisms in COCOMO II estimation scheme. There are 17 effort multipliers as cost drivers and they are devised into two groups; Optimistic Group and Pessimistic Group.

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

Definition-1: Pessimistic Group: This group can be defined as a set of effort multipliers of whose range of values are directly proportional to the overall effort to be predicted and it can be described as follows:

RELY
DATA
CPLX
RUSE
DOCU
TIME
STOR
PVOL

Pessimistic Group

PG= {$EM_{10,}$ $EM_{11,}$ ….. EM_{17}}

Definition-2: Optimistic Group (OG): This group can be defined as a set of effort multipliers of whose range of values are inversely proportional to the overall effort to be predicted and it can be described as follows:

ACAP
PCAP
PCON
AEXP
PEXP
LTEX
TOOL
SITE
SCED

Optimistic Group

OG= {$EM_{1,}$ $EM_{2,}$ ….. EM_{9}}

This classification makes a sense in the estimation scheme and plays a significant role in improving the accuracy of the COCOMO II estimation model. The three attributes model can be visualized as a three edged object in a graphical form.
The effort is E

$$E = \prod_{i=1}^{\alpha} \left(\frac{0.5 * LOC^{2*PG} EM_i}{LOC^{0.88}} \right) * \prod_{i=1}^{\beta} \left(\frac{0.5 * LOC^{2/OG} EM_i}{LOC^{0.88}} \right)$$

Where, α and β are number of non Nominal effort multipliers in the PG and OG.
Further, the above equation is given as input to the Genetic Part. This Genetic approach is used for optimization problems and it curtails the residual values with the Hybrid Imputational approach.

Genetic algorithm is used in this research to enhance the performance of effort estimation. This approach is useful to select the most relevant features of effort estimates and cost drivers which have certainly significant influence on the COCOMO effort estimation model even if the input level is incomplete. If uncertainty exists in the input parameters, errors in the estimates are more or accuracy may not be good.

The general procedure of a genetic algorithm is as follows [Goldberg, 1989; Michalewicz, 1996]

1. Randomly generate a population of solutions.
2. Calculate the fitness value for each of the population.
3. Creation of offspring
   a. Reproduction
   b. Cross over
   c. Mutation
4. Computation of new solution and fitness of each solution.
5. Repeat step 3 to 4 until optimum solution is attained.

The Genetic Approach follows a cyclic process of Reproduction, Crossover and Mutation for N populations. Estimation process starts with an initial population of N project (chromosomes) with 17 EM's (genes) as input which follows genetic operations of Reproduction, Crossover and Mutation. In reproduction initial population chromosomes are grouped based on project mode and each chromosomes in each group is ranked based on fitness function.

For the opted pair of parental chromosomes crossover non-nominal effort multipliers weight age through adjustment. Weight age of operation is performed for repairing OG each non-nominal multiplier are adjusted by adding or deducting weight age based on, tracking made on all the chromosomes for specific effort multiplier.

b. Wilcox on Signed-Rank Test

The performance measure of a model predicting numeric values is the correlation between predicted and

ARPN Journal of Science and Technology

http://www.ejournalofscience.org

actual values. Correlation ranges from +1 to -1 and a correlation of +1 means that there is a perfect positive linear relationship between variables. And can be calculates as follows

The correlation coefficient for COCOMO II is 0.6952 and the correlation coefficient for proposed model is 0.9985

4. EVALUATION AND PERFORMANCE ANALYSIS

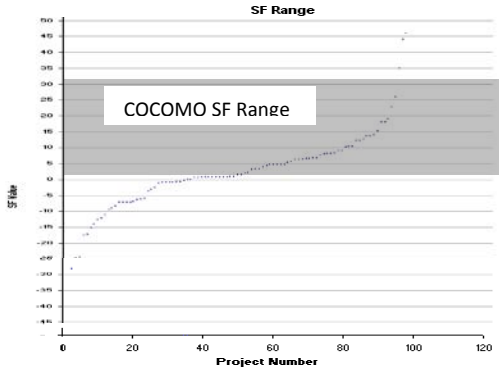
The performance of this model is validated by means of Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE), Root Mean Square (RMS) and Relative Root Mean Square (RMS & RRMS).

A. PERFORMANCE EVALUATION CRITERIA

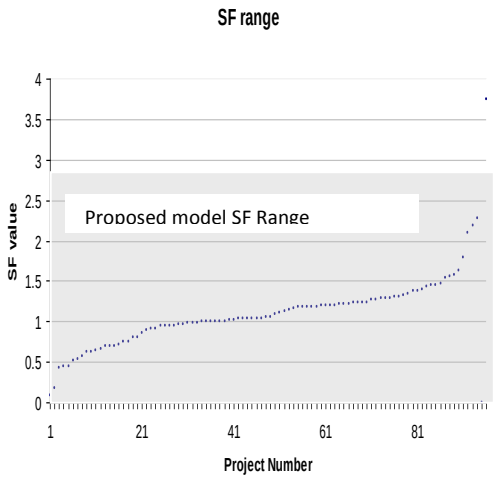
Criteria	Formula	Description
MAE	$MAE = \frac{\sum \lvert E_{ACT} - E_{PRED} \rvert}{n}$	MAE measures of how far the estimates are from actual values. It could be applied between any two pairs like actual and predicted or estimate
MMRE	$MMRE = \frac{100}{N}\sum_{1}^{N}\frac{\lvert predicted_i - actual_i \rvert}{actual_i}$	MMRE is the average percentage of the absolute values of the relative errors over an entire data set.
RMS	$RMS = \left[\frac{1}{N}\sum_{1}^{N}\frac{(predicted_i - actual_i)^2}{actual_i}\right]^{0.5}$	RMS is the square root of the arithmetic mean of the squares of the original values
RRMS	$RRMS = \frac{RMS * N}{\sum_{i}^{N} actual}$	Relative Root Mean square is the ratio between RMS and the average of the actual effort

ARPN Journal of Science and Technology

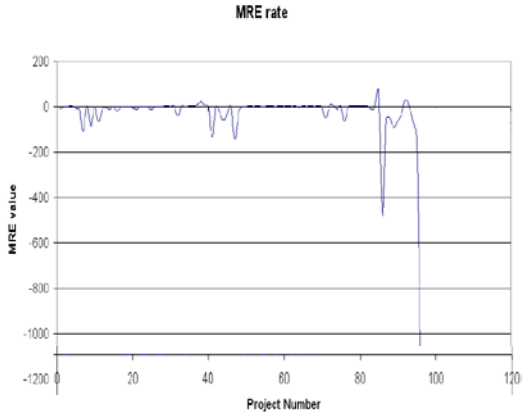
http://www.ejournalofscience.org



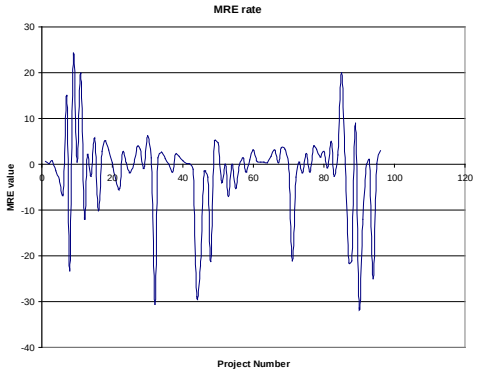
COCOMO Scale Factor for



Proposed model Scale Factor for



COCOMO II MRE Rate



MRE rate of Proposed Model

A. MRE ANALYSIS

Sl. No.	MRE range	COCOMO II		Proposed model	
		No. of Project	Acceptance Status	No. of Project	Acceptance Status
1	>50	1	Under estimation	0	-
2	30 to 50	1	Under estimation	0	-
3	10 to 29.9	1	Under estimation	5	Under estimation
4	5 to 9.9	3	Under estimation	5	Under estimation
5	-5.0 to 4.9	59	accepted	69	accepted
6	-5.1 to -10	5	Over estimated	4	Over estimated
7	-10.1 to -40	10	Over estimated	13	Over estimated
8	-40.1 to - 100	10	Over estimated	0	-
9	<-100	6	Over estimated	0	-

B. PERFORMANCE COMPARISON CHART

Parameters	COCOMO Effort Estimation	Proposed Effort Estimation
No. of Cost drives	17	17
No. of Scale Factors	5	5
Scale Factors range	1.01 to 31.62	0.087 to 2.899
No. of Project lie within the SF range	45	95
No. of project does not lie within the SF range	51	1
Maximum MRE (%)	1,073.51	31.46
MMRE (%)	33.65	5.79
RMS	2254.99	51.1603
RRMS	4.43237	0.100559
Correlation Coefficient	0.6952	0.9985
No. of Project over estimated	47	36
No. of project under estimated	49	60
Standard deviation for over estimated projects	168.8	10.17
Standard deviation for under estimated projects	11.057	5.479
Variance for over estimated projects	28,494.42	103.4668
Variance for under estimated projects	122.1234	30.03

5. CONCLUSION

The purpose of this research was to provide insight into how sophisticated imputation techniques are and this facilitates the understanding. It also integrates between statisticians and software engineers to make succession effort estimation. All decision making process requires quality data which is of greater importance for any validation process. In this paper the validation of COCOMO effort estimation model along with the proposed hybrid model was investigated. It has been found that the hybrid estimation model enhances the performance of effort estimation in early stages of software development life cycle.

REFERENCES

[1] Satyananda, “An improved fuzzy approach for COCOMO’s effort estimation using Gaussian Membership Function” Journal of Software, vol 4, 2009

[2] Little RJA, Rubin DB (1987) “Statistical analysis with missing data”, Wiley series in probability and statistics, 1st edn. Wiley, New York.

[3] B.W. Boehm, “Software Engineering Economics”, Prentice-Hall, 1981.

[4] Ali Idri, Abdelali Zakrani and Azeddine Zahi “Design of Radial Basis Function Neural Network for software effort estimation”, IJCSI, July 2010.

[5] Musflek p, Pedrycz W, Succi G, Reformat M, “Software Cost Estimation with Fuzzy Models” Applied Computing Review, Vol. 8, pp 24-29, 2000

[6] Kpaus Dohmen, “An Improvement of the Inclusion – Exclusion Principle”, Journal of Archiv Der Mathematik, Springer, Vol. 72, pp 298-303, 1999

[7] Jovan Živadinović, Ph.D*, Zorica Medić, Dragan Maksimović, Aleksandar Damnjanović, M.Sc, Slañana Vujčić , “Methods of effort estimation in software engineering", International Symposium Engineering Management And Competitiveness 2011 (EMC2011)June 24-25, 2011, Zrenjanin, Serbia