



Scott Knott Test Based Effective Software Effort Estimation through a Multiple Comparison Algorithms

N.Padma priya¹, D.Vidyabharathi²

PG scholar, Department of CSE, SONA College of Technology, Salem, TamilNadu, India¹

Assistant professor (SG), Department of CSE, SONA College of Technology, Salem, TamilNadu, India²

ABSTRACT: The rapidly increased need for managers is to settle Software Cost Estimation (SCE) for large scaled complex systems. The SCE is one of the most vital activities that is closely related with the success or failure of the whole development process. The Proposed system is a statistical framework based on a multiple comparison algorithms. The cost estimation model identifies the differences in accuracy of data set, and also clusters it into non-overlapping groups. Different software effort estimation models are used and these models have been developed for specific development environments and lead to support specific software development methodologies. Specific technologies are not bound in modern software development methodologies. To overcome this problem a system is proposed to improve cost effort estimation methods and then compared using appropriate statistical procedures for ensuring the appropriate results. An intelligent Expert System is developed which supports all type of software development regardless of their type either using a conventional computer languages or a component based visual languages.

KEYWORDS: software cost estimation; software metrics; software effort estimation; statistical methods.

I. INTRODUCTION

Prediction of the effort is used to complete the software project by comparing the prediction models over past historical data set. This framework is based on a multiple comparisons algorithm, to rank several cost estimation models.

Software Engineering cost model and estimation techniques are used for budgeting, trade-off, risk analysis, and project planning with control to provide software improvement investment analysis.

The estimation increases the breadth of the search for relevant studies which conduct more studies on estimation methods commonly used by the software industry and also increases the awareness of how properties of the dataset impacts the results when evaluating the estimation methods.

Accuracy is measured by the Magnitude of Relative Error (MRE) and MRE to the Estimate (MER). This can be achieved by accurate cost estimation. This needs the knowledge of size specifications, source code, manuals and the rate at which the requirements are likely to change during development and also the probable number of bugs that are likely to be encounter. The capability of development team and the salary over head incase if team increases along with the tools are necessary for estimation.

II. LITERATURE SURVEY

The software cost estimation researchers [2] supports with a list of journal papers with relevant historical papers. The First task is to estimate the purpose to include the particular journal paper for estimating the effort and cost. The second task is to identify the relevant papers. The third task is to classify the papers on their properties with respect to their estimation topics, estimation approach, research approach and result analysis to valid the threads. This review increases the breath of search. Complete search gives study on estimation methods by software industry and predicts the awareness about result impacting. This directs and supports the future estimation research. The other review principally aims to introduce software estimation researchers to the variety of formal estimation models.

The software cost estimation[3] includes the following approaches such as model based, expertise based, learning oriented, dynamic based, regression based and composite cocomo. These approaches capture the knowledge and



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

experience by domain of interest. It estimates the effort hours, staff size and deployment, portfolio impact, risk, maintenance, schedule and hardware resource requirement.

The resembling techniques provide best accuracy measure without its superiority. Simulation tools [4] include regression and estimation by analogy. The accuracy measures

(i) Magnitude of Relative Error (MRE) $y_A = \text{actual}$ $y_E = \text{estimated}$

$$\text{MRE} = \frac{|y_A - y_E|}{y_A}$$

(ii) Magnitude of Relative Error to Estimate (MER).

$$\text{MER} = \frac{|y_A - y_E|}{y_E}$$

Known data set validates the result by simulation method, traditional parametric and non parametric procedures. The datasets are tested by parametric and non parametric paired sample tests, bootstrap confidence intervals, permutation tests and finish test.

Mean Magnitude of Relative Error (MMRE) accurately select the best model [5] and predict the effort from size in 3 ways. The ways are

OL: Ordinary Least squares on the raw data,

MR: Median Regression technique on raw data,

LNOLS: Ordinary Least squares Regression.

The problem of relying on within company [6] involves time to accumulate data, technologies will get change and the data will be collected in consistent manner. This review gives the complete analysis which project used to construct each model: accuracy measured, cross validation methods, fully defined methods, good comparison method. Mean Magnitude of Relative Error [7] predicts the software performance

$$\text{MMRE} = \frac{|y - y^{\wedge}|}{y}$$

$y = \text{actual}$, $y^{\wedge} = \text{prediction}$

It selects the model that is closest to the true model most of the time. MMRE is preferred which can be applied with ease to compare a linear regression model with a non linear arbitrary function estimator.

III. PROPOSED SYSTEM

I. User Interface: A graphical user interface has been developed which provide user with some predefined options as well as some options are provided where user can input in plain English. Predefined options are provided in cases where a numeric value is needed, otherwise natural language has been used for both questions as well as answers the next question displays on the basis of previous response from the user. Thus an intelligent interaction occurs between user and computer.

ii. Natural Language Processor: NLP has been used to translate user response and query to specific rules and vice versa. It simply acts as an interface between User Interface and Inference Engine.

iii. Inference Engine: The basic objective of Inference Engine is to access knowledge Base on the basis of input parameters, supplied by the user. The developed Inference Engine is level 2-Type engine which not only provides basic reasoning but explanation facility has also been added that reproduces the logic to reach its conclusion. In order to reach a conclusion and offer an expert advice to the user, reasoning of the engine has been further strengthened by adding a database of static information. This database contains static information needed for calculation like effort adjustment factors in COCOMO.

iv. Knowledge Base: As the objective of the system is to effort estimation for different types of software development including variation of technology used as well as methodology followed, therefore four sets of rules have been incorporated in the knowledge base to support software effort determination for:

- I. Line of Code base software,
- II. Component base software

a. Define the Public-domain data set



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Public-domain data set with different characteristics are used in order to address the inherent problem of prediction systems, which means their high dependency on the types of data. Alternative error functions measuring different important aspects of error are studied. The repositories contain data from a wide range of projects are in the public domain.

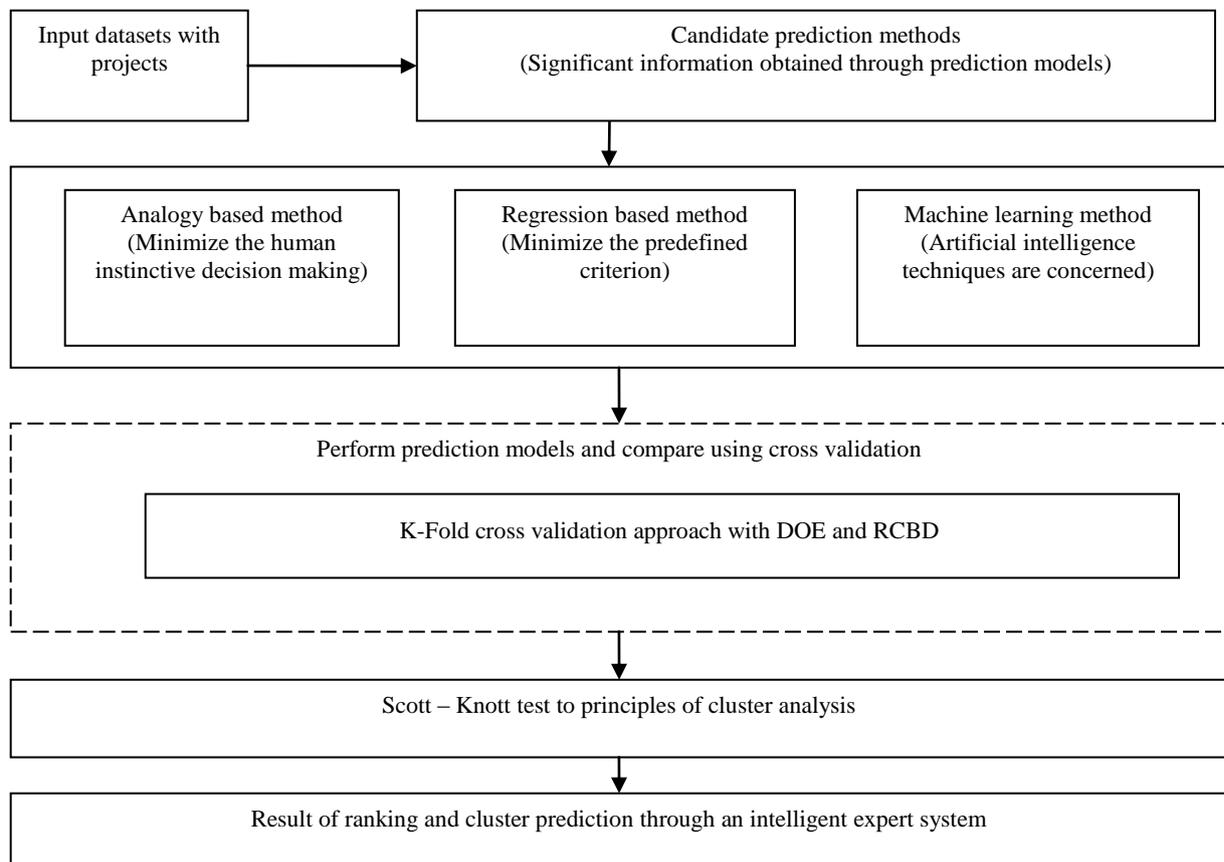
b. Candidate prediction methods

The candidate methods can be grouped into three main categories that are [1] regression-based models, analogy-based models, and machine learning methods. All these models are well-established methods, they are applied in SCE.

An alternative prediction technique was also based on the conclusions of a systematic review on SCE studies. Jorgensen and Sheppard[4] pointed out that the regression-based models dominate since half of all studies deal with the problem of fitting, improvement of a regression model. Furthermore, the researchers' interest for the analogy-based techniques [9] is steadily increased during the end of the decade. At last, the distribution of estimation methods also reveals that the proportion of machine learning techniques (Classification and Regression Trees and Neural Networks) presents an increasing trend.

It is obvious that the prediction techniques used in our experimentation is to tuning of certain parameters in order to build meaningful correct models. Consider example, the ratio-scaled variables of regression-based models[1] are checked in order to investigate whether the normality assumption is satisfied, and also the nominal and ordinal variables are replaced with new dummy variables and then a stepwise procedure is adopted to extract the most significant independent variables. In analogy based methods [8], the dissimilarity measure taking into account various types of variables, for the selection of the best number of the "neighbor" projects is determined through the leave-one-out cross-validation procedure. Regarding neural network models specifies the number of nodes for hidden layers. In RMiner, the NN hyper parameter H [2] is optimized using a grid search with a backward selection algorithm, to avoid over fitting, there is an internal k-fold process is used. Thus the best parameter is selected with; the model is retrained with all training data.

CART model is concerned; utilize the Recursive Partitioning algorithm [1] as implemented in S-PLUS in which the model is fitted using binary recursive partitioning whereby the data are successively split along coordinate axes of the predictor variables so that the split which maximally distinguishes the response variable at any node in the left and the right branches which are to be selected. This splitting continues until nodes are pure or data are too sparse, to the recommendations of S-PLUS manual. Finally, for the case of the Naive Bayes classifier methodology [8] computes the conditional a-posterior probabilities of the dependent variable given the independent predictors using the Bayes rule,



c. Method comparison results

K-fold cross-validation with Design of Experiment

DOE [1] constitutes an entire branch area in statistics involving fundamental concepts that have to be specified and controlled in advance. The basic element of a DOE [1] is the experimental unit, which is the “object” on which the researcher wishes to measure a response variable. The purpose is to study the effect of one or more factors (categorical variables) on the response variable. The different categories of a factor are known as levels or treatments [1]. In the experimental setup [6], the predictive performance of each competitive model is evaluated through a k-fold cross-validation approach in which the original dataset is randomly partitioned into k subsamples of equal size. During a repeated procedure, each one of the subsamples is considered as the validation sample (test set) and the remaining k -1 subsamples as the training sets used for fitting the models.

Repeated Measures Design similarly to the Randomized Complete Block Design (RCBD)

The RCBD [1] incorporates an additional factor takes into account the grouping of similar experimental units. The incorporation of this extra factor is considered advantageous in order to identify true differences between treatments or, equivalently[1], the true treatment effect. Indeed, when different treatments are applied to similar (or the same) experimental units which form, in any sense, a block, there is a source of variation between blocks which cannot be explained by the difference between treatments [1]. This source of variation is represented by the block factor that is considered in the analysis. In our context, the splitting of data [4] into different training-test pair’s represents the blocking factor, i.e., each block is a specific pair of training-test subsets, where all models are applied and validated.

d. Principles of cluster analysis

Scott-Knott procedures are also presented in a graphical manner for two cases. The diagram [1] plots comparative models (x-axis) against the transformed mean errors (y-axis), whereby all methods are sorted according to their ranks. The vertical dashed lines indicate which models give statistically different results and thus are clustered into homogeneous group. The Scott- Knott algorithm resulted in four homogeneous groups of models with similar performances. Each small vertical solid line represents the prediction performance of the competitive models depicts the mean value of the transformed error function [1]. It is clearly inferred from the results of Scott-Knott tests, where the analogy-based techniques are clustered together in the same group of methods for all experiments [1]. Statistical methodology is also based on an algorithmic procedure which is able to produce non-overlapping clusters of prediction models and homogeneous with their predictive performance. It utilizes a specific test of Scott-Knott test which ranks the models and partitions them into clusters. The clustering refers to the treatments being compared and not to the individual cases, while the criterion for clustering together treatments is the statistical significance of differences between their mean values.

e. Performance evaluation

In order to address the disagreement on the performance measures [1], we apply the whole analysis on three functions of error that measure different important aspects of prediction techniques: accuracy, bias, and spread of estimates.

IV. RESULTS AND DISCUSSIONS

SCOTT-KNOTT ALGORITHM:

The Scott-knott algorithm is utilized in cluster analysis to segregate the group of data into separate clusters. The procedure for the Scott-knott algorithm as follows:

Step 1: sort the means of the error measures $e_j^-, j=1, d$ for each model in ascending order.

$$e_{(1)}^- \leq e_{(2)}^- \leq \dots \leq e_{(d)}^- \quad (1)$$

Step 2: For each $e_{(j)}^-, j=1, \dots, d-1$, separate the group of all ordered means E into two subgroups $E_1 = \{e_{(1)}^-, \dots, e_{(j)}^-\}$ and $E_2 = \{e_{(j+1)}^-, \dots, e_{(d)}^-\}$ and compute the between groups sum of squares:

$$G_j = k(|E_1|)(e_{E_1}^- - e_E^-)^2 + |E_2|(e_{E_2}^- - e_E^-)^2 \quad (2)$$

Where $|E_1|, |E_2|$ are the cardinalities of the two subgroups and $e_{E_1}^-, e_E^-, e_{E_2}^-$ are the means of groups.

$$e_E^- = \frac{1}{d} \sum_{j=1}^d e_{(j)}^- \quad (3)$$

$$e_{E_1}^- = \frac{1}{|E_1|} \sum_{j \in E_1} e_{(j)}^- \quad (4)$$

$$e_{E_2}^- = \frac{1}{|E_2|} \sum_{j \in E_2} e_{(j)}^- \quad (5)$$

Step 3: Find the partition that maximizes the value of the sum of squares:

$$G_j = \max_{j \in \{1, \dots, d\}} G_j \quad (6)$$

Step 4: compute

$$\lambda = \frac{\pi}{2(\pi-2)} \frac{G_j}{s^2} \quad (7)$$

χ_v^2 Distribution is computed by

$$V = \frac{k}{(\pi-2)} \quad (8)$$

Step 5: If $\lambda > \chi_v^2$ then the same test is applied to each group separately.

If $\lambda < \chi_v^2$ then all means belongs to the same homogeneous group.

V. CONCLUSION

SCE depends on several issues, even on personal criteria like experience, preference of statistical software. The intelligent expert effort estimation uses User Interface, Natural Language Processor, Inference Engine and Knowledge Base. This expert system improves the software cost effort estimation results and also improves the accuracy in cost estimation. Based on this, the best prediction model for SCE is estimated. The Scott-knott algorithm is thus used to measure the error in the prediction model and also gives the probability of the usage of dataset in concurrent project. The project estimation can also be done using this method and is the future work to be continued. Both the schedule



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

and cost estimation is mandatory requirement for success of the project. The step-by-step procedure followed in this method is considered as one of the major drawback. To overcome this, an efficient algorithm can be designed for estimation process.

REFERENCES

- [1] NikolaosMittas and Lefteris Angelis, "Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm," IEEE Trans. Software Eng., vol. 39, no.4, April. 2013.
- [2] M. Jorgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," IEEE Trans. Software Eng., vol. 33, no. 1, pp. 33-53, Jan. 2007.
- [3] M. Shepperd and G. Kadoda, "Comparing Software Prediction Techniques Using Simulation," IEEE Trans. Software Eng., vol. 27, no. 11, pp. 1014-1022, Nov. 2001.
- [4] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd, "What Accuracy Statistics Really Measure," IEE Proc. Software Eng., vol. 148, pp. 81-85, June 2001.
- [5] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, "A Simulation Study of the Model Evaluation Criterion MMRE," IEEE Trans. Software Eng., vol. 29, no. 11, pp. 985-995, Nov. 2003.
- [6] N. Mittas and L. Angelis, "Comparing Cost Prediction Models by Resampling Techniques," J. Systems and Software, vol. 81, no. 5, pp. 616-632, May 2008.
- [7] E. Stensrud and I. Myrtveit, "Human Performance Estimating with Analogy and Regression Models: An Empirical Validation," Proc. IEEE Fifth Int'l Software Metrics Symp., pp. 205-213, Nov. 1998.
- [8] B. Kitchenham and E. Mendes, "Why Comparative Effort Prediction Studies May Be Invalid," Proc. ACM Fifth Int'l Conf. Predictor Models in Software Eng., pp. 1-5, May 2009.
- [9] I. Myrtveit, E. Stensrud, and M. Shepperd, "Reliability and Validity in Comparative Studies of Software Prediction Models," IEEE Trans. Software Eng., vol. 31, no. 5, pp. 380-391, May 2005.
- [10] S.Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings," IEEE Trans. Software Eng., vol. 34, no. 4, pp. 485-496, July/Aug. 2008.