IJCSEC- International Journal of Computer Science and Engineering Communications.
Vol.3, Issue 2, 2015, Page.636-640, ISSN: 2347–8586

Software Effort Estimation Using Scott Knott Test

M.Sujitha[1],N.Sivakumar[2]
PG Scholar[1, 2],
Dept of Computer Science and Engg[1,2],
RVS School of Engineering[1], University College of Engineering[2],
Dindigul, India[1], Trichy, India[2]
sujineha1@gmail.com[1],nsivamecse@gmail.com[2]

Abstract- Software Cost Estimation is used for large-scaled and complex software systems leads managers to settle SCE as one of the most vital activities that is closely related to predicate the success or failure of the whole development process. Propose a statistical framework based on a multiple comparisons algorithm in order to rank several cost estimation models, identifying those which have significant differences in accuracy, and clustering them in non-overlapping groups. In the existing work Scott-Knott test was used to rank and cluster the software estimation models. The test proposed by Scott Knott, a procedure of means grouping, is an effective alternative to perform procedures of multiple comparisons without ambiguity. This study aimed to propose a modification related to the partitioning and means grouping in the said procedure, to obtain results without ambiguity among treatments, organized in more homogeneous groups. In the proposed methodology, treatments that did not participate in the initial group are joined for a new analysis, which allows for a better group distribution. The proposed methodology is considered effective, aiming at the identification of elite cultivar groups for recommendation.

Index Terms— software cost estimation; software metrics; software effort estimation; statistical methods.

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 overhead incase if team increases along with the tools are necessary for estimation.

LITERATURE SURVEY

Jorgensen and Sheppard [2] survived to classify the estimation topics, estimation approach, research approach and result analysis to valid the threads.The Web-based library cost estimation provided to identify the relevant estimation research. They reported to increase the breadth search for relevant studies and to search manually for relevant papers with set of journals was essential. They were conducted study on estimation method to use by the software industry and to increase the awareness the property of the data sets to impact the result for evaluated estimation method. Mittas and Angelis [3] compared the cost predication model through resampling technique. They compared prediction model based on selection method with various accuracy measure. The most accurate prediction model has to provide the accuracy measure. The accuracy measures used the probability distributions to testing the any hypothesis and the traditional parametric methods. They used statistical simulation tools to test the significance of the difference between the accuracy of two prediction regression and estimation method by analogy. The statistical simulation used permutation tests and bootstrap technique used to build the confidence intervals for the difference measures.

They used as model based, expertise based, learning oriented, dynamic based, regression based and cocomo. They estimated the effort hours, staff size and deployment, portfolio impact, risk, maintenance, schedule and hardware resource requirement.
Kitchenham et. al. [4] provided the software effort estimation of accuracy predictive models such as mean magnitude relative error (MMRE), and number of prediction was within 25% of the actual, pred(25). They used MMRE and pred(25) were measure the spread and the kurtosis. They measured the accuracy of significant and statistics such as MAMRE and pred(25) to measure the property of the distribution. They suggested that measures of the central location and skewness as well as measures of spread and kurtosis. They used as the basis for comparing prediction systems.
The accuracy measures
(I)Magnitude of Relative Error(MRE)

$$y_A = actual \quad y_E = estimated$$

MRE=

(ii)Magnitude of Relative Error to Estimate (MER).
MER=
Known data set validated the result by simulation method, traditional parametric and non-parametric procedures. The datasets were tested by parametric and non-parametric paired sample tests, bootstrap confidence intervals, and 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

MMRE=

y=actual, 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.

PROPOSED SYSTEM

i. User Interface: A graphical user interface is developed which provide user with some predefined options as well as some options are provides where user can input in plain English. Predefined options are provided in cases where a numeric value is needed, otherwise natural language is 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 is used to translate user response and query to specific rules and vice versa. It is simply acts as an interface between User Interface and Inference Engine.

iii. Inference Engine: The goal 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 is further strengthened by adding a database of static information. Inference Engine is intermediate between user interface and knowledge base. 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:

1. Line of Code base software,
2. Component base software.

a. Define the Public-domain data set.

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. The repositories contain data from a wide range of projects are in the public domain.

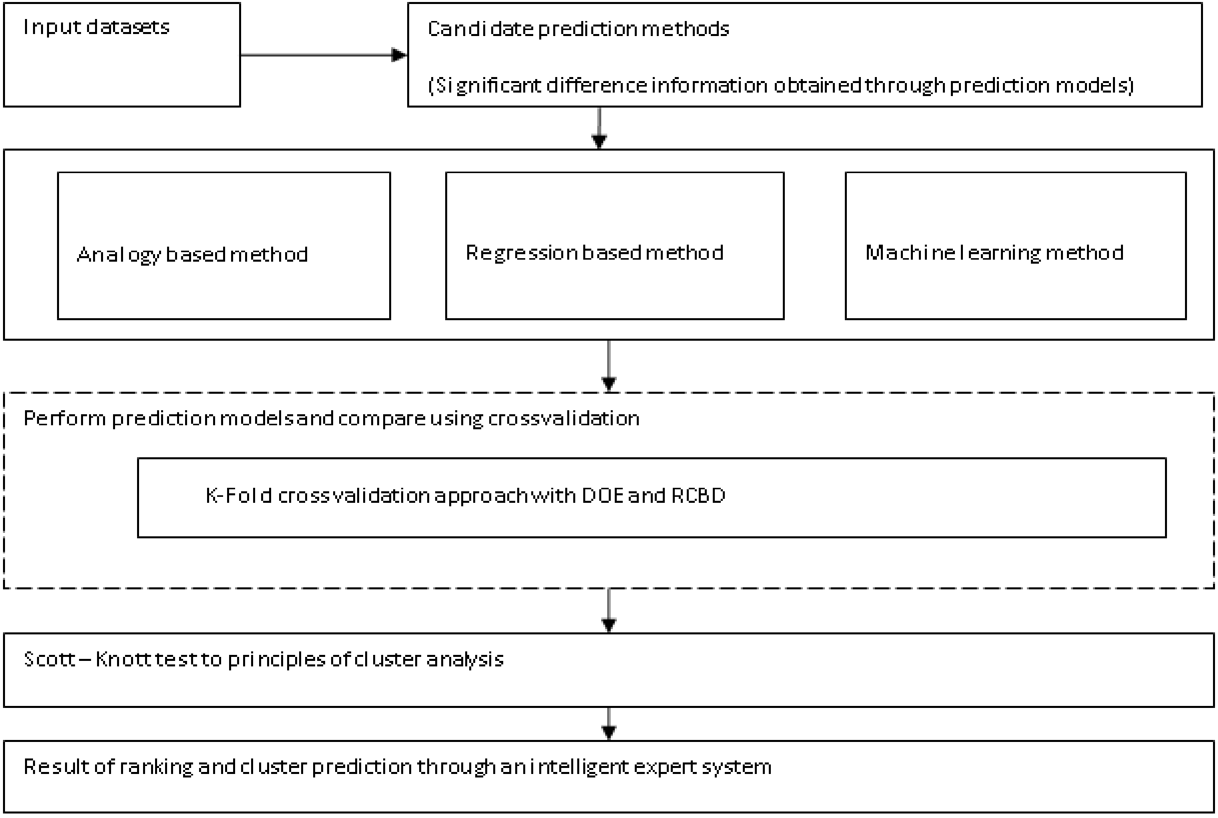
b. Candidate prediction methods

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[2] 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,


Input datasets
Candidate prediction methods
(Significant difference information obtained through prediction models)
Analogy based method
Regression based method
Machine learning method
Perform prediction models and compare using crossvalidation
K-Fold crossvalidation approach with DOE and RCBD
Scott – Knott test to principles of cluster analysis
Result of ranking and cluster prediction through an intelligent expert system

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.

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 ,j=1, d for each model in ascending order.

(1)

Step 2: For each ,j=1,…….,d-1,separate the group of all ordered means E into two subgroups and compute the between groups sum of squares:

G) (2)

Where are the cardinalities of the two subgroups and , , are the means of groups.

(3)

(4)

(5)

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

(6)

Step 4: compute

λ = (7)

Distribution is computed by

V= (8)

Step 5: If then the same test is applied to each group separately.

If then all means belongs to the same homogeneous group.

This methodology proposes an alteration in the way of partitioning groups. The process begins with the formation of groups that maximize the sum of squares, based on the same concept as the Scott-Knott test. Two groups were formed first (one with models 9 and 10 and the second with 1, 2, 3, 4, 5, 6, 7 and 8).

Upon the formation of these groups, the second group was discarded and the possible partitions in the first group performed, resulting in two new subgroups (one with model 10 and the other with 9). This second subgroup was also discarded. Consequently, model 10 represented a group, the first formed group.

A new grouping analysis was performed with all previously discarded models (1, 2, 3, 4, 5, 6, 7, 8, and 9), which divided the models in two groups (group one 6, 7, 8 and 9 and group two 1, 2, 3, 4 and 5). Once again, the models of the second group were discarded and new possible partitions sought in the first group. No possibility of forming new subgroups was verified. Consequently, the models 6, 7, 8 and 9 represent the second group. As the procedure continues, new analyses are carried out with the previously discarded models (1, 2, 3, 4 and 5), until all models are grouped. Summing up, the new procedure consists in the removal of the models that form a new group and in the performance of new analyses with the remaining models, so that at each step a new group is formed while the number of remaining models.

CONCLUSION

The intelligent software cost estimation uses User Interface, Natural Language Processor, Inference Engine and Knowledge Base. These systems establish the significant effort estimation and provide the error accuracy in software cost estimation through the whole project. The candidate prediction models are used in software cost estimation for projects. The Scott test is used to compare multiple prediction models based on error accuracy in each phase of prediction model. Both the schedule 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, pp. 537-551,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.A.Kitchenham, L.M.Pickard,S. G. MacDonell, and M. J.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 Technique”, 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.