Scientific Journal Impact Factor (SJIF): 1.711
e-ISSN: 2349-9745
p-ISSN: 2393-8161
International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Software Cost Estimation Using Clustering and Ranking Scheme

Ms. S. Buvana[1], Dr. M. Latha[2], Mr. R. Subramanian[3]
[1]*Research Scholar*
[2]*M.Sc., M.Phil., Ph.D, Associate Professor, Department of Computer Science*
[1,2] *Sri Sarada College for Women, Salem, Tamilnadu, India*
[3]*Erode Arts and Science College, Erode*

Abstract-Software cost estimation is an important task in the software design and development process. Planning and budgeting tasks are carried out with reference to the software cost values. A variety of software properties are used in the cost estimation process. Hardware, products, technology and methodology factors are used in the cost estimation process. The software cost estimation quality is measured with reference to the accuracy levels.

Software cost estimation is carried out using three types of techniques. They are regression based model, anology based model and machine learning model. Each model has a set of technique for the software cost estimation process. 11 cost estimation techniques fewer than 3 different categories are used in the system. The Attribute Relational File Format (ARFF) is used maintain the software product property values. The ARFF file is used as the main input for the system.

The proposed system is designed to perform the clustering and ranking of software cost estimation methods. Non overlapped clustering technique is enhanced with optimal centroid estimation mechanism. The system improves the clustering and ranking process accuracy. The system produces efficient ranking results on software cost estimation methods.

I. INTRODUCTION

Cost judgment or estimation is one of the most difficult tasks in project management. It is to accurately estimate needed resources and required schedules for software development projects. The software estimation process includes estimating the size of the software product to be produced, estimating the effort required, developing preliminary project schedules and finally, estimating on the whole cost of the project. In the last few years'there are many software cost estimation methods available including algorithmic methods, estimating by analogy, expert judgment method, price to win method, top-down method and bottom-up method. No one method is necessarily better or worse than the other, their strengths and weaknesses are often complimentary to each other. To understand their strengths and weaknesses is very important when the user want to estimate their projects.

For improving the accuracy of cost estimation other fields also calibrate with the software engineering fields. 2CEE is one of the cost estimation model developed by JPL for NASA [4]. This model combination of data mining and software engineering fields. This type of estimation it can calibrate machine learning algorithms with cost estimation models. The main aim of the system to provide a survey of all these type of models and methods and see which model generates the accurate sestimation.

II. RELATED WORK

During the last decades there has been evolving research concerning the identification of the best SCE method [1]. Researchers strive to introduce prediction techniques including expert judgment, algorithmic, statistical and machine learning methods. Miyazaki et al. claimed that the "de facto"

MMRE accuracy measure tends to advance models that underestimate the actual effort while Kitchenham et al. indicated the variation of accuracy measures as a primary source of inconclusive studies. Toward this direction, Foss et al. investigated the basis of this criticism through a simulation study, proposing alternative accuracy indicators and concluding that there is need for applying well established statistical procedures when conducting SCE experiments.

Myrtveit et al. extended the above-mentioned findings and pointed out that inconsistent results are not caused only by accuracy measures but also by unreliable research procedures. Through a simulation study, they studied the consequences of three main ingredients of the comparison process: the single data sample, the accuracy indicator and the cross-validation procedure. Mittas and Angelis [5] showed that the usual practice of promoting a model against a competitive one just by reporting an indicator can lead to erroneous results since these indicators are single statistics of error distributions, usually highly skewed and nonnormal. In this regard, they proposed resampling procedures for hypothesis testing, such as permutation tests and bootstrap techniques for the construction of robust confidence intervals.

Menzies et al. [6] studied the problem of "conclusion instability" through the COSEEKMO toolkit that supports 15 parametric learners with row and column preprocessors based on two different sets of tuning parameters. The Scott-Knott test presented here was used in another context, for combining classifiers applied to large databases. Specifically, the Scott-Knott test and other statistical tests were used for the selection of the best subgroup among different classification algorithms and the subsequent fusion of the models' decisions in this subgroup via simple methods, like weighted voting.

In [7], Demsar discusses the issue of statistical tests for comparisons of several machine learning classifiers on multiple datasets reviewing several statistical methodologies. The method proposed as more suitable is the nonparametric analogue of ANOVA, i.e., the Friedman test, along with the corresponding Nemenyi post hoc test. The Friedman test ranks all the classifiers separately for each dataset and then uses the average ranks of algorithms to test whether all classifiers are equivalent. In case of differences, the Nemenyi test performs all the pairwise comparisons between classifiers to identify the significant differences. This method is used by Lessmann et al. [9] for the comparison of classifiers for prediction of defected modules. The methodology described in our paper, apart from the fact that is applied to a different problem, i.e., the SCE where cost and prediction errors are continuous variables, has fundamental differences regarding the goals, the way it is used and the output.

III. SOFTWARE COST ESTIMATION MODELS

The importance and the significant role of Software Cost Estimation (SCE) to the well-balanced management of a forthcoming project are clearly portrayed through the introduction and utilization of a large number of techniques during the past decades [1]. The rapidly increased need of large-scaled and complex software systems leads managers to settle SCE as one of the most vital activities that is closely related with the success or failure of the whole development process. Inaccurate estimates can be proved catastrophic to both developers and customers since they can cause the delay of the product deliverables or, even worse, the cancellation of a contract.

There is an imperative need to investigate what the state of the art in statistics is before trying to derive conclusions and unstable results concerning the superiority of a prediction model over others for a particular dataset. The answer to this problem cannot constitute a unique solution since the notion of "best" is quite subjective [2]. In fact, a practitioner can always rank the prediction models according to a predefined accuracy measure, but the critical issue is to identify how many of them are evidently the best, in the sense that their difference from all the others is statistically significant. The research question of finding the "best" prediction technique can be restated as a problem of identifying a subset or a group of best techniques. The aim of the system is therefore to propose a statistical framework for comparative

SCE experiments concerning multiple prediction models. It is worth mentioning that the setup of the current study was also inspired by an analogous attempt dealing with the problem of comparing classification models in Software Defect Prediction, a research area that is also closely related to the improvement of software quality.

The methodology is based on the analysis of a Design of Experiment (DOE) or Experimental Design, a basic statistical tool in many applied research areas such as engineering, financial and medical sciences. In the field of SCE it has not yet been used in a systematic manner. Generally, DOE refers to the process of planning, designing and analyzing an experiment in order to derive valid and objective conclusions effectively and efficiently by taking into account, in a balanced and systematic manner, the sources of variation. In the present study [8], DOE analysis is used to compare different cost prediction models by taking into account the blocking effect, i.e., the fact that they are applied repeatedly on the same training-test datasets.

The statistical methodology is also based on an algorithmic procedure which is able to produce nonoverlapping clusters of prediction models, homogeneous with respect to their predictive performance. For this purpose, the system utilizes a specific test from the generic class of multiple comparisons procedures, namely, the Scott-Knott test, which ranks the models and partitions them into clusters.

The statistical framework is applied on a relatively large-scale set of 11 methods over six public domain datasets from the PROMISE repository and the International Software Benchmarking Standards Group (ISBSG) [3]. Finally, in order to address the disagreement on the performance measures, the system applies the whole analysis on three functions of error that measure different important aspects of prediction techniques: accuracy, bias and spread of estimates.

IV. PROBLEM STATEMENT

The important role of well-established statistical comparisons in SCE is highlighted in many recent systems, especially during the last decade, where the findings are derived through formal statistical hypothesis testing. Indeed, the researchers use parametric as well as nonparametric procedures, whereas there has also been increasing interest for more robust statistical tests such as permutation tests and bootstrapping techniques for the construction of confidence intervals. The system belongs to a generic class in statistics known as “multiple hypothesis testing” and can be defined as the procedure of testing more than one hypothesis simultaneously. Briefly describing the problem, the conclusions derived from a statistical hypothesis test are always subject to uncertainty. For this reason, the system uses an acceptable maximum probability of rejecting the null hypothesis when it is true and this is referred to as a “Type I error”. In the case of multiple comparison problems, when several hypotheses are carried out, the probability that at least a Type I error occurs increases dramatically with the number of hypotheses. Multiple Comparative Algorithms (MCA) is used to evaluate the software cost estimation measure quality. The following problems are identified from the current software cost estimation models. The clustering accuracy is very low in the MCA scheme. Ranking prediction accuracy is limited in the MCA scheme. Software product property relationships are not considered.

V. ENHANCED MULTIPLE COMPARATIVE ALGORITHMS (EMCA)

The software cost estimation measures are used to estimate the software product cost for the development process. The Multiple Comparative Algorithms are used to analyze the quality of the software cost estimation measures. The algorithm ranks and clusters the cost prediction models based on the errors measured for a particular dataset. Therefore, each dataset has its own set of “best” models. This is more realistic in SCE practice since each software development organization has its own dataset and wants to find the models that best fit its data rather than trying to find a globally best model which is

unfeasible [10]. Furthermore, the clustering as an output is different from the output of pairwise comparisons tests, like the Nemenyi test. For larger numbers of models the overlapping homogeneous groups resulting from pairwise tests are ambiguous and problematic in interpretation. On the other hand, a ranking and clustering algorithm provides clear groupings of models, designating the group of best models for a particular dataset.

The goal of the system is to further extend the research concerning the comparison and ranking of multiple alternative SCE models. The system uses a framework for conducting comparative experiments and present an evaluation of this analysis over different datasets and prediction models. The Multiple Comparative Algorithm (MCA) based software cost estimation measure quality analysis scheme is improved with optimal centroid based clustering scheme. The system also uses the software product property relationships for cost estimation process. The Enhanced Multiple Comparative Algorithm (EMCA) is proposed to measure and rank the cost estimation measures with high accuracy levels. The Multple Comparative Algorithms (MCA) scheme is used to measure the software cost prediction model quality levels. The clustering and ranking model is used to evaluate the measures. The Non overlapped clustering mechanism is used in the MCA analysis scheme. The EMCA scheme is enhanced with optimal centroid based clustering mechanism to iimprove the cluster quality measures. The rank values are fetched from the EMCA cluster results. The software cost estimation schemes are categorized with reference o the cluster results. Random centroid based clustering scheme is replaced with optimal centroid based model. The system improves the clustering accuracy levels and prediction levels.

VI. SOFTWARE COST ESTIMATION USING CLUSTER ANALYSIS

The software cost estimation scheme analysis operations are evaluated with different statistical models. The Multiple Comparative Algorithms (MCA) scheme is used to rank and cluster the software cost estimation methods. The Enhanced Multiple Comparative Algorithms (EMCA) scheme is used to improve the clustering accuracy and prediction measure analysis. The system is divided into four major modules. They are dataset analysis, cost estimation, MCA scheme and EMCA scheme. The dataset analysis scheme is used fetch data from the datasets. The cost estimation mechanism is used to measure the software cost based on the software product information. The MCA scheme is used to rank and cluster the software cost estimation models. The EMCA scheme is also used to measure the quality of the software cost estimation measures.

6.1. Dataset Analysis

The dataset analysis module is designed to load the dataset values from benchmark data collections. The data values are collected from the standard benchmark datasets. The datasets are collected in ARFF model. The Attribute Relational File Format (ARFF) model is used to provide the software produce details. The attribute for the ARFF file is provided in attribute text files. The ARFF and text files are used as the input for the system. The ARFF file details are converted in to Comma Separated Values (CSV) format using the WEKA tool. The CSV file is used as the input for the software product cost estimation mechanism.

6.2. Cost Estimation

The cost estimation operations are carried out under cost estimation module. The system uses the three different cost estimation measures. They are regression model, analogy model and machine learning model. 11 different cost estimation measures are used in the system. The cost values are estimated using the software product properties collected from the CSV input file. The cost values are

estimated and updated in to the database. The cost estimation process is carried out for the selected software product.

6.3. Ranking and Clustering with MCA

The ranking process is carried out with clustering functions for software cost estimation model analysis. Multiple Comparative Algorithms are used for the ranking and clustering process. Non overlapping clustering scheme is used in the system. Random centroids are used in the clustering process. The software cost estimation measures are ranked with their property values. The clustering and ranking results are produced for the selected software product for all cost estimation measures.

6.4. Ranking and Clustering with EMCA

The Enhanced Multiple Comparative Algorithms (EMCA) scheme is used to rank and cluster the software cost estimation measures. The optimal centorids are used for the clustering process. The system improves the clustering quality with optimal centroids. The software cost estimation measure is analyzed with cost and other property values. The system produces better ranking and clustering solutions for the software cost estimation process.

VII. PERFORMANCE ANALYSIS

The software cost estimation models are analyzed with Multiple Comparative Algorithms (MCA) and Enhanced Multiple Comparative Algorithms (EMCA) schemes. The clustering scheme is improved in EMCA model using the optimal centroid scheme. The system is tested with 11 estimation measures with standard benchmark datasets. The Fitness measures, purity and entropy measures are used for the performance analysis process.

7.1. Datasets

The datasets for the experimentation are derived from two sources, namely, the PROMISE repository and the International Software Benchmarking Standards Group (ISBSG, release 10). The main reason for this selection was that these datasets have been extensively used to empirically validate or justify a large amount of research results, whereas they are also publically available. Each dataset contains a different number of projects and a set of independent variables with mixed-type characteristics, whereas the dependent variable that has to be predicted is the actual effort. Another criterion for the selection of the datasets was the ability to apply all the competitive prediction methods on them. Therefore, the system did not consider datasets with too many categorical variables which cause problems to certain methods like regression and Neural Networks.

The ISBSG repository contains 4,106 software projects from more than 25 countries, but most of the variables have a large amount of missing values. Keeping in mind the guidelines of ISBSG suggesting filtering of the data projects, the system decided to discard the projects with missing values. Moreover, an important issue in SCE is the utilization of datasets with high quality in the process of evaluation and comparison of prediction models. Due to this fact and the instructions of the ISBSG organization that point out not taking into account projects with low quality on projects marked with “A” in Data Quality Rating and UFP Rating. Finally, the independent characteristics utilized in the construction of the alternative models are the order to retain the compatibility with other studies.

7.2. Fitness Measure

The F-measure is a harmonic combination of the precision and recall values used in information retrieval. Each cluster obtained can be considered as the result of a query, whereas each preclassified set of documents can be considered as the desired set of documents for that query. Thus, the system can

calculate the precision P(i, j) and recall R(I, j) of each cluster j for each class i. If n_i is the number of members of the class i, n_j is the number of members of the cluster j and n_{ij} is the number of members of the class i in the cluster j, then P(i, j) and R(i, j) can be defined as

$$P(i, j) = \frac{n_{ij}}{n_j}, \qquad (1)$$

$$R(i, j) = \frac{n_{ij}}{n_i} \qquad (2)$$

The corresponding F-measure F(i, j) is defined as

$$F(i, j) = \frac{2 * P(i, j) * R(i, j)}{P(i, j) + R(i, j)} \qquad (3)$$

Then, the F-measure of the whole clustering result is defined as

$$F = \sum_i \frac{n_i}{n} \max_j (F(i, j)), \qquad (4)$$

where n is the total number of documents in the data set. In general, the larger the F-measure is, the better the clustering result is (2).

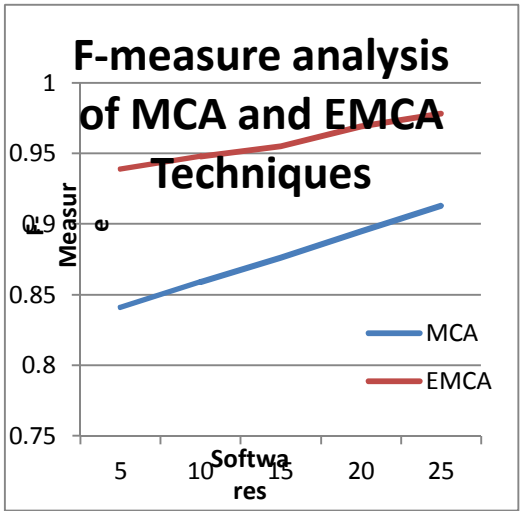

Figure: 7.1. F-measure Analysis of MCA and EMCA Techniques

The Fitness measure analysis between the Multiple Comparative Algorithms (MCA) scheme and the Enhanced Multiple Comparative Algorithm (EMCA) schemes are shown in figure 7.1. The results show that the Enhanced Multiple Comparative Algorithm (EMCA) scheme increases the Fitness measure 10% than the Multiple Comparative Algorithm (MCA) scheme.

7.3. Purity

The purity analysis between the Multiple Comparative Algorithms (MCA) scheme and the Enhanced Multiple Comparative Algorithm (EMCA) schemes are shown in figure 7.2. The results show that the Enhanced Multiple Comparative Algorithm (EMCA) scheme increases the purity analysis 20% than the Multiple Comparative Algorithm (MCA) scheme.

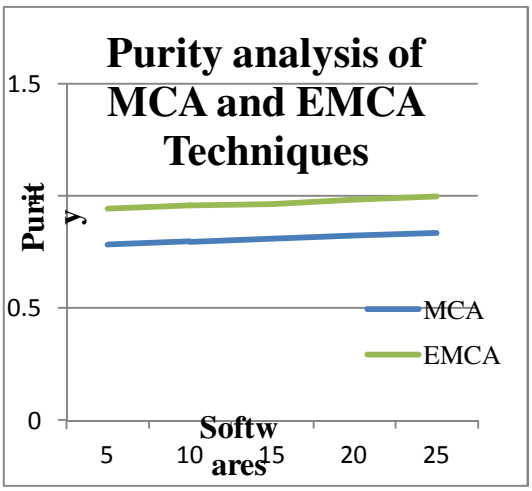

Figure No: 7.2. Purity Analysis of MVSC, SMVSC and HSMVSC Techniques

7.4. Entropy

Entropy reflects the homogeneity of a set of objects and thus can be used to indicate the homogeneity of a cluster. This is referred to cluster entropy. Lower cluster entropy indicates more homogeneous clusters. On the other hand, the system can also measure the entropy of a prelabeled class of objects, which indicates the homogeneity of a class with respect to the generated clusters. The less fragmented a class across clusters, the higher its entropy and vice versa. This is referred to as class entropy.

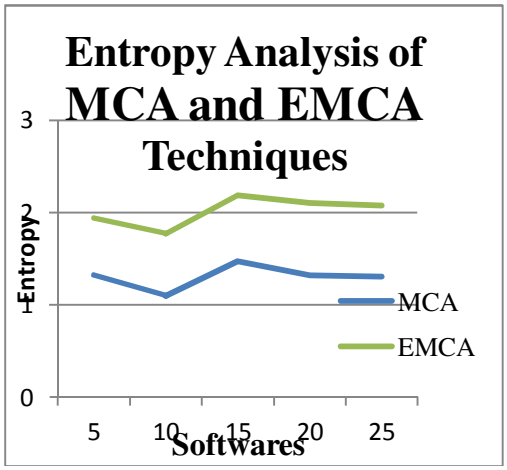

Figure: 7.3. Entropy Analysis of MCA and EMCA Techniques

The Entropy analysis between the Multiple Comparative Algorithms (MCA) scheme and the Enhanced Multiple Comparative Algorithm (EMCA) schemes are shown in figure 7.3. The results show that the Enhanced Multiple Comparative Algorithm (EMCA) scheme increases the Entropy analysis 25% than the Multiple Comparative Algorithm (MCA) scheme.

VIII. CONCLUSION AND FUTURE WORK

The software cost estimation measures are analyzed using the statistical methods. Ranking and clustering operations are carried out under the analysis process. The system deals with a critical research issue in software cost estimation concerning the simultaneous comparison of alternative prediction models. They are ranking and clustering in groups of similar performance. The system examined the

predictive power of 11 models over six public domain datasets. The system uses the Multiple Comparative Algorithms (MCA) for the clustering and ranking process. The MCA scheme is enhanced with optimal centroid models to improve the accuracy level. The standard benchmark data values are used to test the quality of the ranking and clustering system.

The software cost estimation is performed with different types of estimation methods. The system analyzes the software cost estimation methods and produces the ranked result based on the quality and accuracy factors. Regression, Analogy and Machine learning approaches are used in the cost estimation process. The system can be enhanced with the following features.

- The system can be enhanced to measure the maintenance cost for the software products.
- The system can be adapted to measure the efforts and risk levels for the software's.
- The system can be improved to analyze the design quality issues based ranking and clustering scheme.

REFERENCES

[1] M. Jorgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," IEEE Trans. Software Eng., Jan. 2007.
[2] Nikolaos Mittas and Lefteris Angelis, "Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm", IEEE Transactions On Software Engineering, Vol. 39, No. 4, April 2013.
[3] ISBSG Data Set 10, http://www.isbsg.org. 2007.
[4] Qiang He, Jun Han, Jean-Guy Schneider and Steve Versteeg, "Formulating Cost-Effective Monitoring Strategies for Service-Based Systems", IEEE Transactions On Software Engineering, May 2014.
[5] 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.
[6] T. Menzies, D. Baker and K. Lum, "Stable Rankings for Different Effort Models," Automated Software Eng., vol. 17, no. 4, pp. 409-437, Dec. 2010.
[7] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," J. Machine Learning Research, 2006.
[8] Ayse Tosun Misirli and Ayse Basar Bener, "Bayesian Networks For Evidence-Based Decision-Making in Software Engineering", IEEE Transactions On Software Engineering, June 2014
[9] S. Lessmann, C. Mues and S. Pietsch, "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings," IEEE Trans. Software Eng., July/Aug. 2008.
[10] Juan Manuel Vara, Alvaro Jimenez and Esperanza Marcos, "Dealing with Traceability in the MDD of Model Transformations", IEEE Transactions On Software Engineering, June 2014.

