International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Ranking and Clustering of Software Cost Estimation Models

Vijaya Wable[1], S. M. Shinde[2]

[1]ME (COMP) Student, JSCOE, Hadapsar, India

[2]Assistant Professor and HOD, JSCOE, Hadapsar, India

Abstract: ***In current scenario, software industries have various software cost estimation models to estimate the financial need and to develop software. The result of these models mainly requires obtaining approval to proceed and factored into business plans, budgets, financial planning and tracking mechanisms. Many of these models are providing irrelevant output thereby leading the organization in confusion. So it's necessary to choose a perfect cost estimation model which becomes the higher priority for the companies. With our research to rank the cost estimation models, proposed system uses previous performance data sets as the evidence. This System uses correlation similarity and preference model to identify the rank of the model and thereby cluster the cost estimation models. In our proposed model we have taken many parameters to perform ranking and clustering. We are targeting to demonstrate abilities of software cost estimation method and clustering them based on their features. It helps us to rank together for further usage of software.***

Keywords: cost estimation, cost estimation technique, ranking, clustering method

1. Introduction

From previous projects, this information can be used by management to improve the planning of personnel, to make more accurate tendering bids, and to evaluate risk factors. Recently, a number of studies evaluating different techniques have been published. The results of these studies revealed mainly three factors:

- There is no standard method to confirm single cost estimation model.
- Old statistical data never narrate present scenarios
- Single parameters based ranking was not enough
- These factors have been considered while preparing this proposed work & finding improvement areas. Main Objectives of proposed system are
- First to study the all possible cost estimation models
- Understanding the methods and equations of the model which are using for cost estimation
- Calculating cost Error Precision[3]
- By Using Data sets and similarity measure ranking [8] for complete models and cluster those according to their performance rate e.g. high, medium and low.

2. Literature Survey

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. The usual practice of these studies was to compare the proposed estimation method with established models on a small number of datasets

M. Jorgensen [1] and M. Shepperd provide us a basis of the improvement of software estimation research through a systematic review of previous work. The review identifies 304 software cost estimation papers in 76 journals and classifies the papers according to research topic, estimation approach, research approach, study context and data set cost estimation papers is studies, they conduct more studies on estimation methods commonly used by the software industry, a increase the awareness of how properties of the data sets impact the results when evaluating estimation methods.

Marian Petre [2] describes models whose purpose is to explain the accuracy and bias variation of an organization's estimates of software development effort through regression analysis.

B. A. Kitchenham [3] provide the software estimation research community with a better understanding of the meaning of, and relationship between, two statistics that are often used to assess the accuracy of predictive models: the mean magnitude relative error

Leonardo Lopes Bhering[4], proposed test 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 The Scott-Knott test presented here was used in another context in [5], 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 that study extensive experiments with very large datasets showed that the Scott-Knott test provided the highest accuracy in difficult classification problems. Hence, the choice of the test for the present paper was motivated by former results obtained by one of the authors

In [6] 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 pair wise comparisons between classifiers to identify the significant differences.

This method is used by Lessmann [7] 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. Specifically, the algorithm he propose ranks and clusters the cost prediction models based on the errors measured for particular dataset. Therefore, each dataset has its own set "best" models.

Nikolaos Mittas[8] 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. The proposed framework is applied in a large-scale setup of comparing 11 prediction models over six datasets. The results illustrate the benefits and the significant information obtained through the systematic comparison of alternative methods

3.Estimation Techniques

A. Algorithmic Models

These models work based on the especial algorithm. They usually need data at first and make results by using the mathematical relations. Nowadays, many software estimation methods use these models. Algorithmic Models are classified into some different models. Each algorithmic model uses an equation to do the estimation:

$$\boldsymbol{Effort = f(X1, X2, \ldots . Xn)}$$

where $(X1, X2, .. Xn)$ is the vector of the cost factors. The Differences among the existing algorithmic methods are related to choosing the cost factors and function. All cost factors using in these models are:

- Product factors: required reliability, product complexity, database size used, required reusability, documentation match to life-cycle needs.
- Computer factors: execution time constraint, main storage constraint, computer turnaround constraints, platform volatility.
- Personnel factors: analyst capability, application experience, programming capability, platform experience, language and tool experience; personnel continuity.
- Project factors: multisite development; use of software tool; required development schedule.

1) Source Line of Code

SLOC is an estimation parameter that illustrates the number of all commands and data definition but it does not include instructions such as comments, blanks, and continuation lines. This parameter is usually used as an analogy based on an approach for the estimation. After computing the SLOC for software, its amount is compared with other projects which their SLOC has been computed before, and the size of project is estimated. SLOC measures the size of project easily. After completing the project, all estimations are compared with the actual ones.

Thousand Lines of Code (KSLOC) are used for estimation in large scale. Using this metric is common in many estimation methods. SLOC Measuring seems very difficult at the early stages of the project because of the lack of information about requirements. Since SLOC is computed based on language instructions, comparing the size of software which uses different languages is too hard. Anyway, SLOC is the base of the estimation models in many complicated software estimation methods. SLOC usually is computed by considering S_L as the lowest, S_H as the highest and S_M asthe most probable size (Roger S. Pressman, 2005).

$$S = \frac{S_L + 4S_M + S_H}{6}$$

2) Seer-Sem

SEER-SEM model has been proposed in 1980 by Galorath Inc (Galorath, 2006). Most parameters in this method are commercial and, business projects usually use SEER-SEM as their main estimation method. Size of the software is the most important feature in this method and a parameter namely Se is defined as effective size. Se is computed by determining the five indicators: newsize, existingsize, reimpl and retst as below:

Se==Newsize+ExistingSize(0.4Redesign+0.25reimp+0 .35 Retest)

After computing the Se the estimated effort is calculated as below:

$$Effort = t_d = D^{-0.2} \times \left(\frac{S_e}{C_{te}}\right)^{0.4}$$

where D is relevant to the staffing aspects; it is determined based on the complexity degree in staffs structure. C is computed according to productivity and efficiency of the project method is used widely in commercial projects.

3) Linear Models

Commonly linear models have the simple structure and trace a clear equation as below:

$$\text{Effort} = a_0 + \sum_{i=1}^{n} a_i x_i$$

Where, a1, a2.., an are selected according to the project information.

4) COCOMO

Cost models generally use some cost indicators for estimation and notice to all specifications of artifacts and activities. COCOMO 81 (Constructive Cost Model), proposed by Barry Boehm (Boehm, 1981), is the most popular method which is categorized in algorithmic methods. This method uses some equations and parameters, which have been derived from previous experiences about software projects for estimation. COCOMO-II is the latest version of COCOMO that predicts the amount of effort based on Person-Month (PM) in the software projects. It uses function point or line of code as the size metrics and composes of 17 Effort Multipliers (shown in Table II) and 5 scale factors (shown in Table III). Some rating levels are defined for scale factors including very low, low, nominal, high, very high and extra high. A quantitative value is assigned to each rating level as its weight. COCOMO II has some special features, which distinguish it from other ones. The Usage of this method is very wide and its results usually are accurate.

5) Putman's model

This model has been proposed by Putman according to manpower distribution and the examination of many software projects (Kemerer, 2008). The main equation for Putnam's model is:

$$S = E \times (\text{Effort})^{1/3} t_d^{4/3}$$

where, E is the environment indicator and demonstrates the environment ability. T is the time of delivery. Effort and S are expressed by person-year and line of code respectively. Putnam presented another formula for Effort as follows:

$$\text{Effort} = D_0 \times t_d^3$$

Table 1: Effort Multipliers

Attribute	Type	Description
RELY	Product	Required system reliability
CPLX	Product	Complexity of system modules
DOCU	Product	Extent of documentation required
DATA	Product	Size of database used
RUSE	Product	Required percentage of reusable components
TIME	Computer	Execution time constraint
PVOL	Computer	Volatility of development platform
STOR	Computer	Memory constraints
ACAP	Personnel	Capability of project analysts
PCON	Personnel	Personnel continuity
PCAP	Personnel	Programmer capability
PEXP	Personnel	Programmer experience in project domain
AEXP	Personnel	Analyst experience in project domain
LTEX	Personnel	Language and tool experience
TOOL	Project	Use of software tools

4. System Overview

Mathematical equation:

1) Pearson Correlation Model Equation

$$r = \frac{\sum xiyi - \frac{\sum xi \sum yi}{10}}{\sqrt{\left(\sum x_i^2 - \frac{(\sum x_i)^2}{10}\right)}\sqrt{\left(\sum y_i^2 - \frac{(\sum y_i)^2}{10}\right)}} \quad \text{..... (1)}$$

Let x={ Ac, Op, Er} and y=={ Ac, Op, Er}

Take Accuracy, Opinion Score, Error rate as input parameter get rank r as output

2) Preference function

$$V^{\Psi}(\rho) = \sum_{i,j:\rho(i)>\rho(j)} \Psi(i,j) \quad \text{..... (2)}$$

Block diagram:

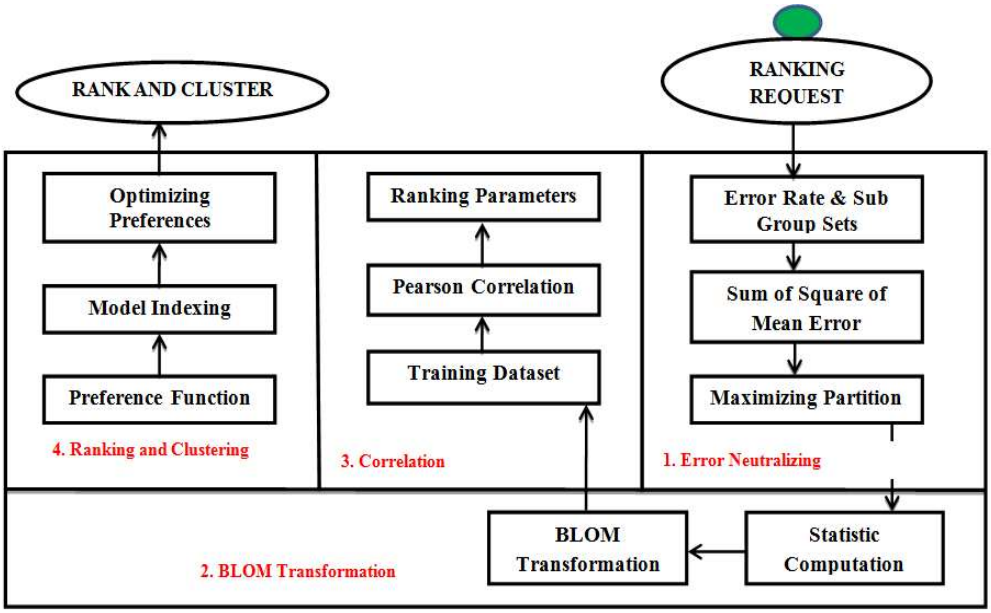
Following are the steps for system flow from user give ranking request to getting rank and cluster of software cost estimation model.

1) Error Neutralization

Step 1) In this first give ranking request then perform following step 1) Accepting Error Rate from Dataset.
Step 2) Divide mean errors in sub group sets
Step 3) then calculate the group sum of squares of the mean errors
Step 4) Finding the partition that maximizes the value of the sum of square

2) Bloom transformation

Step 5) Compute the statistics using the following equation



$$\lambda = \frac{\pi G_{j*}}{2(\pi - 2)S^2}$$

step 6) Applying Bloom transformation

$$\Phi^{-1}\left(\frac{r_i - 3/8}{n + 1/4}\right)$$

It is important to note that the Bloom transformation is monotonous and therefore the order of the values is kept intact. The output of the algorithm is a ranking of the models according to their transformed error measures and, moreover, a clustering scheme where each cluster consists of the sorted models that do not have significant difference in their error measures.

3) Correlation

Step 7) Accept Accuracy, Opinion Score, Error rate as input
Step 8) detecting similar models with similar attributes using Pearson Rank Correlation using equation (1)
Step 9) Getting Two model similarity
Step10) Getting training Dataset
Step11) Detecting the similar model in Training dataset
Step12) Vector of similar model set
Step 13) Getting All model names

4) Ranking and clustering

Step15) Detecting users preference over two models using preference function
Step 16) Detecting Model corresponding order
Step 17) Model indexing
Step18) Optimizing Models
Step 19) Ranking models

5. Algorithm design and platform

A.Algorithm for Error Neutralization and Bloom Transformation

Input: Dataset D ={e_1,e_2,e_3,….e_n}
Output: Rs as rank
Step 0)start
Step1) Get Set D
Step2)divide D into subgroup
Step3)

$$\sum_e 2 = e_1^2 + e_2^2 + \cdots \ldots \ldots \ldots e_n^2$$

Step4)calculate maximum partition for error rate generate set M_e
Step5) Compare M_e with other subset
Step6) Using Blom transformation get

distribution error rates

$$\Phi^{-1}\left(\frac{r_i - 3/8}{n + 1/4}\right)$$

Step 7) Merge all sub grops
Step8)compute in descending order
Step9)index X rank
Step10)stop

B. Algorithm for Correlational

Input: Dataset Ds and training set Ts, Accuracy Ac, Opinion set Op, Error rate Er
Output: Rs as rank
step:
step 1)Accept User Parameter acceptance x,y
step 2) Accept Accuracy, Opinion Score, Error rate
let x={ Ac, Op, Er} and y=={ Ac, Op, Er}
step 3)detecting similar models with similar attributes

using Pearson Rank Correlation these are following step for that use equation 2

step i)calculate x*y

$$\sum x{:}\, y = x_1y_1 + x_2y_2 + x_3y_3 + \cdots \ldots \ldots \ldots \ldots x_ny_n$$

so $\sum x{:}\, y$=A

step ii)

$$\sum x = x_1 + x_2 + x_3 + \cdots \ldots \ldots \ldots \ldots x_n = B$$

Step iii)

$$\sum y = y_1 + y_2 + y_3 + \cdots \ldots \ldots \ldots \ldots y_n = C$$

step iv) $N_{r=A-(B*C)/n}$

step v) $\sum_x 2 = x_1^2 + x_2^2 + \cdots \ldots \ldots \ldots x_n^2 = M)$)

step vi)Q=B^2/n

step vii)Z=$\sqrt{M-Q}$

step viii)

$$\sum_y 2 = y_1^2 + y_2^2 + y_3^2 + \cdots \ldots \ldots \ldots y_n^2 = V$$

step ix)u=C^2/n

step x)Z=$\sqrt{V-u}$

step xi)dr=T-Z

step xii) $P_r = nr/dr$

C. Algorithm of Ranking and clustering

1) Getting Two model similarity
2) Getting training Dataset
3) Detecting the similar model in Training dataset
4) Vector of similar model set
5) Getting All model names
6) Detecting users preference over two models using preference function

$$V^{\Psi}(\rho) = \sum_{i,j:\rho(i)>\rho(j)} \Psi(i,j)$$

7) Detecting Model corresponding order
8) Model indexing
9) Optimizing Models
10) Ranking models

User of the system should have operating systems like Windows XP, Vista and Windows7. The system is implemented using JAVA. We required Minimum of Dual Core of 2.2 GHZ, 2GB RAM.

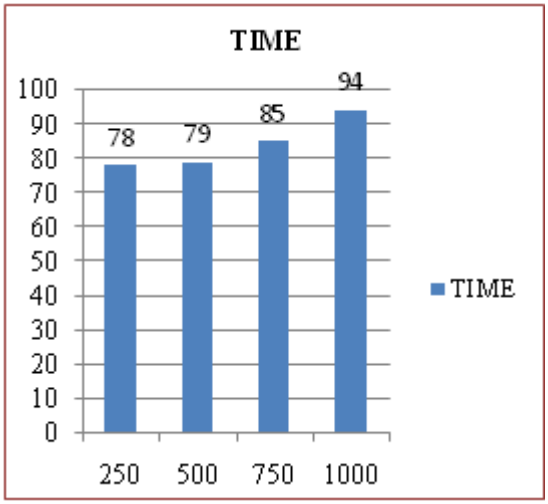
6. Conclusion and Future Scope

In our proposed approach we have successfully created 7 to 10 cost estimation software and ask the user to perform the operation of different software as input so that many outcomes of these type of operation can be save in database and considered further as dataset. This dataset gets feed as input of our estimation model where using similarity search, different function, we rank the model and finally cluster them based on rank. Our model can be enhance as web application where different countries cost estimation parameters can be given as input to rank cost estimation model.

7. Result

Finally system show following result. first we take 250 data of all five models it's processing time is 78 millisecond as Data get increases there processing time is also get increased.

Data	Time
250	78
500	79
750	85
1000	94



8. Acknowledgment

I express true sense of gratitude towards my project guide Prof. S.M. Shinde, head of computer department for his invaluable co-operation and guidance that she gave me throughout my project. I like to specially thank our P.G coordinator Prof. M.D.Ingle for inspiring me and providing me all the lab facilities, which made project work very convenient and easy. I would also like to express my appreciation and thanks to JSCOE principal Dr. M.G. Jadhav and all my friends who knowingly or unknowingly have assisted me throughout my hard work.

References

[1] M. Jorgensen and M. Shepperd "A Systematic Review of Software Development Cost Estimation Studies", vol. 33, no. 1, pp. 33-53, Jan. 2007.

[2] Marian Petre, David Budgen and Jean Scholtz says in "Regression Models of Software Development Effort Estimation Accuracy and Bias". Empirical Software Engineering, 9, 297–314, 2004

[3] B.A.Kitchenham ,L.M.Pickard, S.G.MacDonell band, M.J.Shepperd " What accuracy statistics really measure" , vol. 148, pp. 81-85, June 2001.

[4] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings," vol. 34, no. 4, pp. 485-496, July/Aug. 2008.

[5] G. Tsoumakas, L. Angelis, and I. Vlahavas, "Selective Fusion of Heterogeneous Classifiers," Intelligent Data Analysis, vol. 9, no. 6, pp. 511-525, Dec. 2005.

[6] J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets" , J. Machine Learning Research, vol. 7, pp. 1-30, 2006.
[7] 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.
[8] Nikolaos Mittas and Lefteris Angelis "Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm", VOL. 39, NO. 4, APRIL 2013

Author Profile

Ms Vijaya Wable received the B.E (IT) from SVPMCOE, Malegoan (BK) and M.E. degrees in Computer Engineering from JSMP College of Engineering in Pune. She works as Senior Lecturer in Jayawantrao Sawant Polytechnic. She is interested doing work on research of software cost estimation Model.

Prof. Sharmila M. Shinde is serving as Assistant Professor, Head of Computer Engineering Department, Jayawantrao Sawant College of Engineering, Hadapsar, Pune University, India