ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Comparing Software Cost Estimation Using Scott-Knott Test

M.Sujitha[1], N.Sivakumar[2]

P.G. Student, Department of CSE, RVS School of Engineering and Technology, Dindigul, India[1]

University College of Engineering, Anna University (BIT Campus), Trichirappalli, India[2]

ABSTRACT*:* Software cost Estimation is the approximate judgement of the costs for a project. It is to predict the most realistic effort needed to complete a software project. Accurate effort prediction is done by comparing the trained dataset through multiple algorithms. In this paper, the proposed cost estimation model identifies the differences in accuracy of data set, and also clusters it into non-overlapping groups. 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. The Scott Knott test is to perform multiple comparisons without ambiguity. The proposed modification related to the partitioning and means grouping to obtain results without ambiguity among models. In the proposed methodology, models that did not participate in the initial group are joined for a new analysis, which allows for a better group distribution.

KEYWORDS*:* software cost estimation, software metrics, software effort estimation, statistical methods.

I.INTRODUTION

Software cost estimation is a complex activity that requires knowledge of a number of key attributes about the project for which the estimate is being constructed. Cost estimating is sometimes termed “parametric estimating” because accuracy demands understanding the relationships among scores of discrete parameters that can affect the outcomes of software projects, both individually and in concert.

Although the factors that influence the outcomes of software projects are numerous and some are complex, modern commercial software cost-estimation tools can ease the burden of project managers by providing default values for all of the key parameters, using industry values derived from the integral knowledge base supplied with the estimation tools. In addition, software cost-estimation tools allow the construction of customized estimating templates that are derived from actual projects and that can be utilized for estimating projects of similar sizes and kinds.

Software is now the driving force of modern business, government, and military operations. This means that a typical Fortune 500 corporation or a state government may produce hundreds of new applications and modify hundreds of existing applications every year. As a result of the host of software projects in the modern world, software cost estimating is now a mainstream activity for every company that builds software.

Unlike other kinds of project management tools, the commercial software cost-estimating tools do not depend upon the expertise of the user or project manager, although experienced managers can refine the estimates produced. The commercial cost-estimating tools contain the accumulated experience of many hundreds or thousands of software projects. Because of the attached knowledge bases associated with commercial cost-estimating tools, novice managers or managers faced with unfamiliar kinds of projects can describe the kind of project being dealt with, and the estimating tool will construct an estimate based on similar projects derived from information contained in its associated knowledge base.

The starting point of software estimation is the size of the project in terms of either logical source code statements, physical lines of code, function points, or, sometimes, all three metrics. The project’s size can be derived

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

from the estimating tool's own sizing logic, supplied by users as an explicit input, or derived from analogy with similar projects stored in the estimating tool's knowledge base. Even for Agile projects and those using iterative development, at least approximate size information can be created.

Using commercial estimating tools, these project attributes can either be supplied by the user or inherited from similar projects already stored in the estimating tool's knowledge base. In a sense estimating tools share some of the characteristics of the object-oriented paradigm in that they allow inheritance of shared attributes from project to project. In software-estimating terminology, these shared attributes are termed templates, and they can be built in a number of ways. For example, estimating-tool users can point to an existing completed project and select any or all of the features of that project as the basis of the template. Thus, if the project selected as the basis of a template were a systems software project, used the C programming language, and utilized formal design and code inspections, these attributes could be inherited as part of the development cycle and could become part of an estimating template for other projects.

As a general rule, software-estimating templates are concerned with four key kinds of inherited attributes: (1) personnel, (2) technologies, (3) tools, and (4) the programming environment .Three of these four factors—the experience of the personnel, the development process, and the technology (programming languages and support tools)—are fairly obvious in their impact. What is not obvious, but is equally important, is the impact of the fourth factor—environment. The environment factor covers individual office space and the communication channels among geographically dispersed development teams. Surprisingly, access to a quiet, noise-free office environment is one of the major factors that influences programming productivity. The ability to include ergonomic factors in an estimate is an excellent example of the value of commercial software cost-estimating tools. Not only do they contain the results of hundreds or thousands of completed projects, but the tools contain data about influential factors that many human project managers may not fully understand.

The four key sets of attributes must be considered whether estimating manually or using an automated estimating tool. However, one of the key features of commercial software-estimating tools is the fact that they are repositories containing the results of hundreds or thousands of software projects, and so the effect of these four attribute areas can be examined, and their impacts can be analysed.There is a standard sequence for software cost estimation, which the author has used for more than 35 years. This sequence can be used with manual software cost estimates, and also mirrors the estimation stages in the software-estimation tools that the author has designed. There are ten steps in this sequence, although the sequence starts with 0 because the first stage is a pre-estimate analysis of the requirements of the application

.

II.LITERATURE SURVEY

Lessmannet.al. [1] presented a framework for organizing comparative classification experiments in software defect prediction and conduct a large-scale benchmark of 22 different classification models over 10 public-domain data sets from the NASA Metrics Data (MDP) repository and the PROMISE repository. Overall, an appealing degree of predictive accuracy is observed, which supports the view that metric-based classification is useful. Comparisons are based on the area under the receiver operating characteristics curve (AUC). The benchmarking study assesses the competitive performance of several established and novel classification models so as to appraise the overall degree of accuracy that can be achieved with (automated) software defect prediction today, investigate whether certain types of classifiers excel, and thereby support the (pre)selection of candidate models in practical applications. The issue of these methods had not the potential to achieve general accuracy improvements across all types of classifiers. The advantages of the method are degree of nonlinearity within the MDP data sets is limited and compared to cross validation or bootstrapping; the split sample setup saves a considerable amount of computation time, which, in turn, can be invested into model selection to ensure that the classifiers are well tuned to each data set. Tsoumakaset. al. [2] reported the best subgroup among different classification algorithms and the subsequent fusion of the decision of the models in this subgroup with simple methods like Weighted Voting. The main motivation for combining multiple predictive models is the improvement over the accuracy of a single classification or regression model. Another reason is the scaling of inductive algorithms to very large databases. Most inductive algorithms are too computationally complex and suffer from memory problems when applied to very large databases. A solution to this problem is to horizontally partition the

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

database into smaller parts, train a predictive model in each of the smaller manageable part and combine the predictive models. Finally, the problem of learning from multiple physically distributed data sets that can't be collected to a single site due to privacy or size reasons, can also be tackled with the combination of multiple predictive models, each trained on a different distributed data set. The issue are stacking will have a severe problem to generalize well from the meta-level training data for a large ensemble size. The problem with Stacking is the high dimensionality of the meta-data for many classifiers. The advantages are improves over simple selection and fusion methods, leading to performance comparable with the state-of-the-art heterogeneous classifier combination method of Stacking without the additional computational cost and learning problems of meta-training.Menzieset. al. [3] has reported the study of 158 effort estimation methods on data sets based on COCOMO features. Four "best" methods were detected that were consistently better than the "rest" of the other 154 methods. These rankings of "best" and "rest" methods were stable across (a) three different evaluation criteria applied to (b) multiple data sets from two different sources that were (c) divided into hundreds of randomly selected subsets using four different random seeds. The complication is that any future effort estimation analysis should be preceded by a "selection study" that finds the best local estimator. However, the simplification is that such a study need not be labor intensive, at least for COCOMO style data sets. The Issues are implications of our work on other estimation frameworks is an open and pressing issue. These biases can make us miss or misunderstand important effects. Hence, it is important to have some audit tool to check human intuitions. The advantagesof data pruning is to reduce the prediction variance. Dem_sar [4] has presented the current practice and then theoretically and empirically examines several suitable tests. Based on that, we recommend a set of simple, yet safe and robust non-parametric tests for statistical comparisons of classifiers: the Wilcoxon signed ranks test for comparison of two classifiers and the Friedman test with the corresponding post-hoc tests for comparison of more classifiers over multiple data sets. A number of test data sets is selected for testing, the algorithms are run and the quality of the resulting models is evaluated using an appropriate measure, most commonly classification accuracy. The issue of multiple hypothesis testing is a well-known statistical problem and the t-test is, just as averaging over data sets, affected by outliers which skew the test statistics and decrease the test's power by increasing the estimated standard error. The advantages are the empirical analysis also shows that replicability of the tests might be a problem, thus the actual experiments should be conducted on as many data sets as possible and empirical results suggest that they are also stronger than the other tests studied.

III.PROPOSED WORK

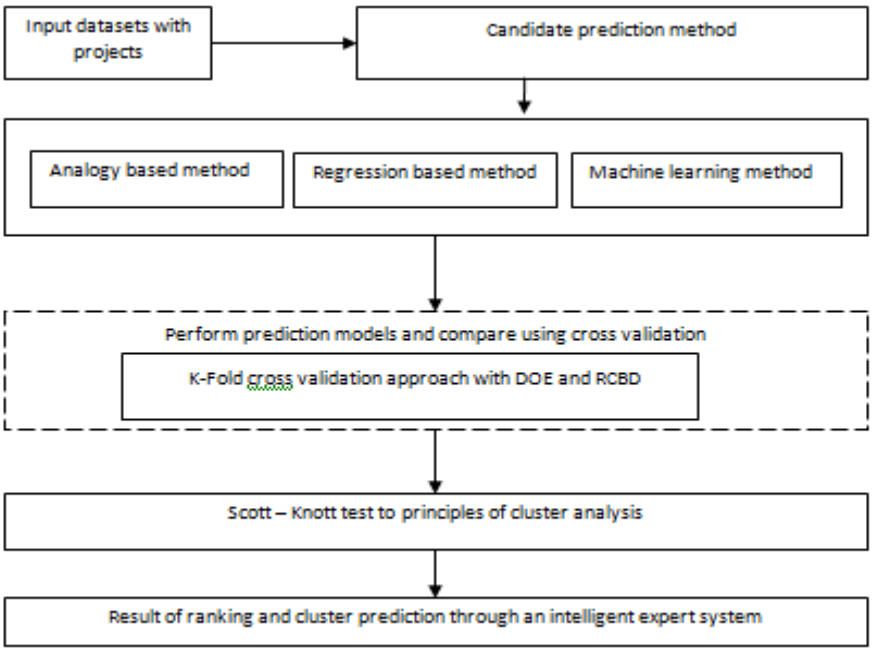

Figure no. 1. System Architecture

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Public-domain datasets with different characteristics are used in order to address the inherent problem of prediction systems, i.e., 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 and, more importantly, they are in the public domain. The candidate methods can be grouped into three main categories that are regression-based models, analogy-based techniques, and machine learning methods. All these models are well-established methods, there is a vast literature on them, and they have been already applied in SCE.

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

3.1 Define the Public-domain Datasets

Public-domain datasets with different characteristics are used in order to address the inherent problem of prediction systems, i.e., 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 and, more importantly, they are in the public domain.

3.2 Candidate Prediction Methods

The candidate methods can be grouped into three main categories that are regression-based models, analogy-based techniques, and machine learning methods. All they have been applied in SCE.

- **Regression-based Model**

Ordinary least squares (OLS)

Thismethod is used for estimating the unknown parameters in a linear regression model. This method minimizes the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation. The resulting estimator can be expressed by a simple formula, especially in the case of a single regressor on the right-hand side.

Suppose the data consists of nobservations $\{ y_i, x_i \}n$ i=1. Each observation includes a scalar response y_i and a vector of p predictors (or regressor) x_i. In a linear regression model the response variable is a linear function of the regressor:

$$y_i = x_i'\beta + \varepsilon_i$$

where β is a p×1 vector of unknown parameters; ε_i's are unobserved scalar random variables (errors) which account for the discrepancy between the actually observed responses y_i and the "predicted outcomes" $x'_i\beta$; and ′ denotes matrix transpose, so that x′ β is the dot product between the vectors x and β.

Least trimmed squares (LTS)

It, is a robust statistical method that fits a function to a set of data whilst not being unduly affected by the presence of outliers. It is one of a number of methods for robust regression.

Instead of the standard least squares method, which minimises the sum of squared residuals over n points, the LTS method attempts to minimise the sum of squared residuals over a subset, k, of those points. The n-k points which are not used do not influence the fit.

In a standard least squares problem, the estimated parameter values, β, are defined to be those values that minimise the objective function, S(β), of squared residuals

$$S = \sum_{i=1}^{n} r_i(\beta)^2,$$

where the residuals are defined as the differences between the values of the dependent variables (observations) and the model values

$$r_i(\beta) = y_i - f(x_i, \beta),$$

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Least Median of Squares Regression

The n observations $(x^i, y^i) = (x^{il}, \ldots, x^i,, y^i)$ belong to the linear space of row vectors of dimension p assumed that all observations k_i^{th} $x_i = 0$ have been deleted, because they give no information on 0. (This condition is automatically satisfied if the model has an intercept because then the last coordinate of each xi equals 1.) Moreover, it is assumed that in the (p + 1)- dimensional space of the (x_i, y_i), there is no vertical hyperplane containing more than [n/2] observations. (Here, a vertical hyperplane is a p-dimensional subspace that contains (0, . . . ,0) and (0, . . . , 0, 1). The notation [r] stands for the largest integer less than or equal to r.)

This is the maximal value for all regression-equivariant estimators. (By regression equivariance, I mean the property $T(\{(x_i,y_i + x_iv); i = 1 , \ldots ,n)) = T(\{(x_i, y_i); i = 1, \ldots , n)) + v$ for any vector v.) For this variant of the LMS, Corollary 1 holds whenever strictly more than +(n + p - 1) of the observations are in an exact fit situation, which also corresponds to the repeated median.

- **Naïve bayes classification**

It is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model" .Abstractly, the probability model for a classifier is a conditional model.

$p(C|F_1, \ldots , F_n)$

Class variable $C$with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. The problem is that if the number of features $n$is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, this can be written,

$$p(C|F_1, \ldots , F_n) = \frac{p(C)\ p(F_1, \ldots , F_n|C)}{p(F_1, \ldots , F_n)}.$$

In plain English the above equation can be written as,

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}.$$

- **Neural network (NN)**

A network structure is defined with a fixed number of inputs, hidden nodes and outputs. The first approach incorporates the neural network module into the existing or modified expert system. This approach uses the neural network to preprocess the incoming data for suspicious activities and forwards them to the expert system. This enhances the efficiency of the detection system .Second; an algorithm neurotree is chosen to realize the learning process to detect intrusion. In this method, the neural network process data from the network audit logs and analyzes it for intrusion detection. The trained NN ensemble is employed to generate a new training set through replacing the desired class labels of the original training examples, with those output from the trained NN ensemble. Some extra training examples are also generated from the trained NN ensemble and added to the new training set.

3.3Method Comparison Results

- **K-fold cross-validation approach with DOE**

DOE 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 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. In the case of our experimental setup, 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.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

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

The RCBD setup incorporates an additional factor, the so-called block, which 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, 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.

3.4 Principles of Cluster Analysis

Scott-Knott procedures are also presented in a graphical manner for two cases. The diagram 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 for each one of the competitive models and, more precisely, the asterisk depicts the mean value of the transformed error function. 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. 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, we utilize 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 clustering refers to the treatments (methods or in our case models) 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.

3.5 Performance Evaluation

In order to address the disagreement on the performance measures, 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 DISCUSSION

SCOTT-KNOTT ALGORITHM

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

$$e_{(1)}^- \le e_{(2)}^- \le \cdots \le e_{(d)}^-$$

Step 2: For each$e_{(j)}^-$,j=1,…….,d-1,separate the group of all ordered means E into two subgroups

$E_1 = \{e_{(1)}^-, \dots, e_{(j)}^-\}andE_2 = \{e_{(j+1)}^-, \dots, e_{(d)}^-\}$

and compute the between groups sum of squares:

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

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.

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

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

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

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

$$G_j = \max\{G_j, j = 1, \dots, d\}$$

Step 4: compute

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

distribution is computed by

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

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

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

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

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.

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 data set in concurrent project. The project estimation can also be done using this method and is the future work to be continued. 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. The proposed methodology makes a differentiated partitioning of the available models possible while ensuring the principle of absence of ambiguity or superposition of model groups. The proposed methodology is a more effective option when the objective is to identify one or few elite groups and discard inferior and intermediate groups. There is a loss of the global partitioning structure, while the identification of a specific subgroup with better performance is facilitated**.**

REFERENCES

[1] 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.
[2] G. Tsoumakas, L. Angelis, and I. Vlahavas, "Selective Fusion of Heterogeneous Classifiers," Intelligent Data Analysis, vol. 9, no. 6, pp. 511-525, Dec. 2005.
[3] T. Menzies, O. Jalali, J. Hihn, D. Baker, and K. Lum, "Stable Rankings for Different Effort Models," Automated Software Eng., vol. 17, no. 4, pp. 409-437, Dec. 2010.
[4] J. Dem_sar, "Statistical Comparisons of Classifiers over Multiple Data Sets," J. Machine Learning Research, vol. 7, pp. 1-30, 2006.
[5] 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.
[6] 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.
[7] M. Shepperd and G. Kadoda, "Comparing Software Prediction Techniques Using Simulation," IEEE Trans. Software Eng., vol. 27, no. 11, pp. 1014-1022, Nov. 2001.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.