Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 1, January 2017, pg.61 – 67

Machine Learning Methods of Effort Estimation and It's Performance Evaluation Criteria

Rekha Tripathi[1], Dr. P. K. Rai[2]
Study Centre for Computer Application A.P.S.University, Rewa [M.P], India
[1] Rekhatripathi1984@gmail.com, [2] pkrapsu@gmail.com

Abstract— Effort estimation is important for the control, quality and success of any software development product. Most efficient categories of effort estimation is Expert judgment, Algorithmic estimation and Machine Learning. The aim of this paper is to present the comparative analysis between traditional techniques and Machine Learning (ML) techniques. Results show that ML methods give more accurate effort estimation as compared to the traditional methods of effort estimation. The comparisons of different Machine learning techniques are done in this paper to study that which ML method is more suitable in which situation.

Keywords— Artificial intelligence (AI), Machine learning (ML), Genetic algorithm (GA), Case based reasoning (CBR), Artificial Neural Network (ANN)

I. INTRODUCTION

Effort estimation is playing an important role in successful management and quality of software development projects. It is used to measure the effort required to develop or maintain software product in person months [8]. Both under estimation and over estimation of software effort may lead to risky consequences and sometime even causes the failure of a project. It is important for judging the cost of the project and to avoid budget overrun [10]. Effort estimation in early stages of software development has been in the focus of software engineering research for a long time and it has led to the development of several approaches for effort estimation.

In effort estimation many ways are available for categorizing estimation approaches. Most efficient categories are as follows-
(i) Expert estimation: Its quantification step; on the basis of judgmental process estimation is done.

(ii) Formal estimation: Its quantification step is based on mechanical processes, e.g., the use as a formula derived from historical data.

(iii) Hybrid estimation: This estimation approach deals with a judgmental or mechanical combination of estimates from different sources.

Software effort estimation techniques fall under following three main categories:

A. Expert Judgment:
In this method an expert of software development processes estimates software development parameters. The accuracy of expert judgment estimates depends on the degree in which a new project matches within the experience and the ability of the expert. [2] [9].

B. Algorithmic Estimation:

The algorithmic estimation method is designed to provide some mathematical equations to perform software estimation. These mathematical equations are based on research and historical data and use inputs such as Source Lines of Code, number of functions to perform, and other cost drivers such as language, design methodology, etc.

C. Machine Learning:

Most techniques about software cost estimation use statistical methods, which are not able to present reason and strong results. Machine learning approaches could be appropriate at this field because they can increase the accuracy of estimation by training rules of estimation and repeating the run cycles. Commonly used ML techniques are artificial neural networks case based reasoning, Rule Induction, Genetic Algorithm, Classification and regression trees and Multiple additive regression trees (MART) etc[11]. It is difficult to determine which technique gives more accurate result on which dataset. However, a lot of research has been done in Machine learning techniques of estimation and literature suggests that ML methods are capable of providing adequate estimation models as compared to the traditional models [5][6].

II. Traditional Techniques

2.1 Function Point Analysis:

The Function Point Analysis is method of quantifying the size and complexity of a software system in terms of the functions that the systems deliver to the user. In function point analysis measurement is based on the functionality of the program. The total number of function points depends on the counts of distinct types[1]. There are two steps in counting function points:

a. Counting the user functions:- The raw function counts are arrived at by considering a linear combination of five basic software components: external inputs, external outputs, external inquiries, logic internal files, and external interfaces, each at one of three complexity levels: simple, average or complex. The sum of these numbers, weighted according to the complexity level, is the number of function counts (FC).

b. Adjusting for environmental processing complexity:- The final function point is arrived at by multiplying FC by an adjustment factor that is determined by considering 14 aspects of processing complexity[15]. This adjustment factor allows the FC to be modified by at most 35% or -35%.

Strengths:

- Function points can be estimated from requirements specifications.
- It is possible to estimate development costs in the early phases of SDLC.
- It is independent of the language, tools, or methodologies used for implementation.

Weakness:

- It requires manual work, as it can be counted manually and which require more time.
- It requires detailed knowledge of requirement for estimation of software size using function points.
- New developer cannot easily estimate the size as it requires experience with function point.

2.2 Expert Judgment Method:

Expert judgment techniques involve consulting with software effort estimation expert or a group of the experts to use their experience and understanding of the proposed project to arrive at an estimate of its cost. It is the most usable methods for the software effort estimation.. Generally speaking, a group consensus technique, Delphi technique, is the best way to be used. To provide a satisfactorily broad communication bandwidth for the experts to exchange the volume of information necessary to calibrate their estimates with those of the other experts, a wideband Delphi technique is introduced over standard Delphi technique [2]. The estimating steps using this method are as follows:

a. Coordinator presents each expert with a specification and an estimation form.

b. Experts fill out forms anonymously.

c. Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.

d. Coordinator prepares and distributes a summary of the estimation on an iteration form.

e. Experts fill out forms, again anonymously, and steps 4 and 6 are iterated for as many rounds as appropriate.

The wideband Delphi Technique has subsequently been used in a number of studies and cost estimation activities. It has been highly successful in combining the free discuss advantages of the group meeting technique [15].

Strengths:

- Availability of experts makes a difference between the past project experience and the requirements of the proposed project.
- Experts can tell impacts caused by a new technique, architecture, language involved in future projects.

Weakness:

- It is hard to document the factors used by the experts or experts-group.
- The expert may be some biased, optimistic, and pessimistic, even though they have been decreased by the group consensus.

2.3 Estimating by Analogy:

In this method comparing the proposed project to previously completed similar project where the project development information id known. Actual data from the completed projects are extrapolated to estimate the proposed project. It is straightforward method act similar with expert judgment method experts often search for analogous situations to inform their opinion [12][15].

The estimating steps using this method are as follows:

a. Find out the characteristics of the proposed project.

b. Select the most similar completed projects whose characteristics have been stored in the historical data base.

c. Find out the estimate for the proposed project from the most similar completed project by analogy.

Strengths:

- Works based on actual experiences.
- Having especial expert is not important.
- Easy to find differences between the completed and the proposed project can be identified and impacts estimated.

Weakness:

- A lots of information about past projects is required
- In some situations there are no similar project.
- They are restricted to information that is available at the point that the prediction required.

III. Machine Learning Techniques

3.1 Artificial Neural Network (ANN):

An artificial neural network is a network composed of artificial neurons or nodes which imitate the biological neurons [4]. ANN have been established to be an effective tool for pattern classification and clustering. There are several algorithms available to train a neural network but this depends on the type and topology of the neural network. Each neuron is connected with the other by a connection link[14]. Each connection link is associated with weights which contain information about the input signal. This information is used by the neuron net to solve a particular problem. Each neuron has an internal state of its own. This internal state is called the activation level of neuron, which is the function of the inputs the neuron receives. . An ANN is composed of nodes organized into layers and connected through weight elements. At each node, the weighted inputs are aggregated, threshold and inputted to an activation function to generate an output of that node [1].

Strengths:

- Estimation process can give accurate results due to training of network process.
- Consistent with unlike databases, Power of reasoning.

Weakness:

- The knowledge of internal working is never known.
- Complexity increases due to learning of parameters.
- Secondly to fully implement a neural network architecture would require a lot of computational resources.

3.2 Case-Based Reasoning (CBR):

Case based reasoning is a problem solving paradigm. It is used for effort estimation of web applications. CBR based on the psychological concepts of analogical reasoning, dynamic memory and the role of previous situations in learning and problem solving [7]. It is a cyclic procedure that is composed of four stages:

1. Retrieval of similar case or cases
2. Reuse of the retrieved cases to find a solution to the problem,
3. Revision the proposed solution if necessary
4. Retention of the solution to form a new case.

When a new problem arises, we solve problems by adapting the solutions from similarly solved problems. The solution may be revised based upon experience of reusing previous cases and the outcome retained to supplement the case repository. Consequently, issues concerning case characterization, similarity and solution revision must be addressed prior to CBR system deployment [1][17].

Strengths:

- CBR has the potential to alleviate problems with calibration.
- CBR can be valuable where the domain is complex and difficult to model.
- The basis for an estimate can be easily understood.
- It can be used with partial knowledge of the target project.
- Intuitive an have a reasonable level of accuracy.
- CBR is also simple and flexible.
- May be applied to both qualitative and quantitative data, reflecting typical industrial datasets

Weakness:

- As with algorithm models, the effect of old data points is not clear.
As an organization develops and
- successively introduces new technology, the older data points may become increasingly irrelevant and potentially misleading.

3.3 Classification and Regression Tree Algorithm (CART :

CART is a tree-building technique that is used in data mining problems involving classification and regression. It is a very efficient machine learning technique and it requires very little input from the analyst. CART is a non-parametric statistical methodology developed for analyzing classification issues either from categorical or continuous dependent variables [17]. If the dependent variable is categorical, CART produces a classification tree. When the dependent variable is continuous, it produces a regression tree.

Strengths:

- CART is nonparametric. Therefore this method does not require specification of any functional form.
- CART does not require variables to be selected in advance.
- CART algorithm will itself identify the most significant variables and eliminate non-significant ones.
- CART results are invariant to monotone transformations of its independent variables.
- CART can easily handle outliers.
- CART has no assumptions and computationally fast.
- CART is flexible and has an ability to adjust in time.

Weakness:

- CART may have unstable decision trees.
- CART splits only by one variable.

3.4 Genetic Algorithm (GA):

The technique of GA was developed for handling general optimization problems with large search apace. Genetic algorithm based on the principles of biological evolution like natural selection. Genetic algorithms use a vocabulary borrowed from natural genetics in that they talk about genes or bits, chromosomes or bit strings, and population of individuals [17]. The basic process of GA is as follows:

1) Randomly generate a set of solutions.

2) Applying GA to the fittest chromosomes to create a new population from the previous one.

3) Repeat step 2 until the required number of generation has been produced the best solution in the final round is taken as a best approximation for that problem [3].

Strengths:

- GA can provide some significant enhancement in accuracy and has the potential to be a valid additional tool for effort estimation.
- It has been used for difficult numerical optimization problems.
- GA have been successfully used to solve system identification, signal processing and path searching problems.

Weakness:

- The main problem of this is premature convergence which does not allow it to access whole solution space constraining it to converge to a local optimum.
- The problem of choosing the various parameters such as the size of the population, mutation rate, cross over rate, the selection method and its strength.
- It have trouble finding the exact global optimum.

IV. Evaluation of Estimation Methods

The current state of software effort performance criterion is not very satisfying. In the past, many metrics and number of methods in cost estimation have been proposed. Unfortunately, most of them defined have lacked one or both of two characteristics:
(a) Sound conceptual, theoretical bases
(b) Statistically significant experimental validation
Most performance criterion metrics have been defined by an individual and then tested in a very limited environment.

In this section, we try to show the minimum distance between estimated parameters and actual ones in an experience. Measuring the performance of estimation methods is accomplished by computing several metrics including SSE,RMSE, MMRE, MAE etc. They are the popular parameters which are used for performance evaluation of methods [16].

1. Root Mean Squared Error (RMSE) The root mean squared error is defined as RMSE is recurrently used measure of variation between values predicted by a model or estimator and the values actually experimental from the thing being modeled or estimated. The root mean squared error is defined as:

$$RMSE = (\sqrt{1/n \sum_{i=1}^{n} (P_i - A_i)^2})$$

Where P_i = Predicted value for data point i;
A_i =Actual value for the data point i;
n = Total number of data points.

Advantage:
RMSE can only be compared between models whose errors are measured in the same units. There is no absolute criterion for a "good quality" value of RMSE. It depends on the units in which the variable is measured and on the degree of forecasting precision.

Drawback:
RMSE is not a good indicator of average model performance and might be a misleading indicator of average error, The RMSE is more appropriate to represent model performance than the MAE when the error distribution is expected to be Gaussian.

2. Mean Magnitude of Relative Error (MMRE) : The most widely used evaluation criterion to assess the performance of software prediction is the MMRE [16] this is computed as:

$$\text{MMRE} = 1/n \sum_{i=1}^{n} \left(\frac{(P_i - A_i)}{A_i} \right)$$

Where P_i = Predicted value for data point i;
A_i =Actual value for the data point i;
n = Total number of data points.

Advantage:

The advantage is that the comparison can be made across datasets and it is independent whether effort is measured in work/hours or work/months.

Drawback: This method is not independent of scale.

3. Mean Absolute Error (MAE): The mean absolute error measures of how far the estimates are from actual values. It could be applied to any two pairs of numbers, where one set is "actual" and the other is an estimate prediction[16].

$$\text{MAE}=1/n\sum_{i=1}^{n}(P_i - A_i)$$

Where P_i = Predicted value for data point i
A_i = Actual value for data point i
n = Total number of data points.

Advantage:
MAE is strongly suggested for error prediction while performing the intelligent data analysis.

Drawback:
MAEs is used absolute value, which is highly undesirable in many mathematical calculations.

4. Sum Squared Error (SSE): This statistic measures the total deviation of the response values from the fit to the response values. It is also called the summed square of residuals and is usually labeled as SSE.
The sum squared error is defined as:

$$SSE = \sum_{i=1}^{n}(P_i - A_i)^2$$

Where P_i = Predicted value for data point i;
A_i =Actual value for the data point i;
n = Total number of data points.

Advantage:

It is symmetrical, smooth and differentiable. There is a unique point optimizing the loss function, which is not the case using mean absolute error.

Disadvantage:

It can be used specific type models for example measure of variation within a cluster. If all cases within a cluster are identical the SSE would then be equal to 0.

V. Conclusions

In this paper we have presented a overview of machine learning estimation techniques and their strengths and weakness and also described performance evaluation criteria of estimation techniques. Some traditional techniques about effort estimation use statistical methods, which are not able to present reason and strong results. Machine learning approaches could be appropriate estimation technique because they can increase the accuracy of estimation by training rules of estimation and repeating the run cycles.
There are many factors that impact software effort estimates such as team size, concurrency, intensity, fragmentation, software complexity, computer platform and different site characteristics in case of software development. Researchers have developed different methods for estimation but there is no estimation method, which can present the best estimates in all various situations, and each technique can be suitable in the special project. For instance GA is useful in the area of scientific research involving biological evolution whereas CART analysis may be useful in many financial applications. Similarly CBR is being developed for use in help- desk Systems, a relatively new application and ANN may be employed for risk management or sales forecasting.

REFERENCES

[1] G. R. Finnie and G. E. Wittig, "A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks " Case-Based Reasoning and Regression Models. J.SYSTEMS SOFTWARE, pp.281-289, 1997.
[2] Jorgensen, M. "Practical guidelines for expert-judgment-based software effort estimation", IEEE Software, 22(3), 57-63. Doi: 10.1109/MS. 73, 2005.
[3] J. B. Colin and L. MartiCan, "Genetic programming improves software effort estimation? A comparative evaluation. Information and software technology, pp. 863-873, 2001.
[4] J. Kaur, S. Singh, K. S, Kahlon, and P. Bassi, "Neural Network-A Novel Technique for Software Effort estimation," International Journal of Computer Theory and Engineering, vol. 2, no. 1. pp. 17-19, 2010.
[5] K. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort," IEEE Transactions on Software Engineering, vol.21, Feb.1995.

[6] L. Radlinki and W. Hoffmann, "On Predicting Software Development Effort Using Machine Learning Techniques and Local Data," International Journal of Software Engineering and Computing, vol. 2, pp.123-136, 2010.
[7] Letchmunam,Roper,wood,"Investigating effort prediction of web based application using CBR",2004.
[8] Lalit V. Patil ,Sagar K Badjate and S.D.Joshi," Develop Efficient Technique of Cost Estimation Model for Software Applications "International Journal of Computer Applications (0975 – 8887) Volume 87 – No.16, February 2014 .
[9] Mamoona Humayun and Cui Gang" Estimating Effort in Global Software Development Projects Using Machine Learning Techniques" International Journal of Information and Education Technology, Vol. 2, No. 3, June 2012.
[10] M. Fabriek, M. van de Brand, S. Brinkkemper, F. Harmsen, and R.W. Helms, "Reasons for success and failure in offshore software development projects," European Conference on Information System, pp. 446-457, 2008.
[11] M. Ruchika and J. Ankita, "Software Effort Prediction using Statistical Machine Learning Methods, " (IJACSA) InternationalJournal of Advanced Computer Science and Applications, vol. 2, no.1, 2011.
[12] Moayed, Ghani,Mojtaba,"Comparing between Web Application Effort Estimation Methods", Fifth International Conference on Computational Science and applications, 2007,IEEE.
[13] M. O. Elish, "Improved estimation of software project effort using multiple additive regression trees," Expert Systems with Applications pp. 10774–10778, 2009.
[14] Prabhakar and Maitreyee Dutta, " Prediction of Software Effort Using Artificial Neural Network and Support Vector Machine " International Journal of Advanced Research in Computer Science and Software Engineering, March - 2013, pp.40-46.
[15] Rekha Tripathi, Dr. P. K. Rai" Comparative Study of Software Cost Estimation Techniques" International Journal of Advanced Research in Computer Science and Software Engineering Volume 6, Issue 1, January 2016.
[16] S.Malathi andDr.S.Sridhar " Analysis of size matrics and effort performance criterion in software cost estimation" Indian Journal of Computer Science and Engineering (IJCSE) Vol. 3 No. 1 Feb-Mar 2012.
[17] Yogesh Singh, Pradeep Kumar Bhatia and Omprakash Sangwan" A Review of Studies on Machine Learning Techinques" International Journal of Computer Science and Security, Volume (1) : Issue (1).

