International Journal of Computer Applications (0975 – 8887)
Volume 59– No.8, December 2012

Emerging Estimation Techniques

Amrinder Singh Grewal
M.E, CSE
University institute of Engineering and Technology, Panjab University, Chandigarh

Vishal Gupta
Assistant Professor, CSE
University Institute of Engineering and Technology, Panjab University, Chandigarh

Rohit Kumar
Assistant Professor, CSE
University Institute of Engineering and Technology, Panjab University, Chandigarh

ABSTRACT

Software cost estimation is the process of predicting the effort required to develop a software system. Accurate cost estimation helps us complete the project within time and budget. There are lots of methods are there for efforts and cost estimation, but people do not know how to use these methods. This paper provides a general overview of software cost estimation methods including the recent advances in the respective field. Machine learning techniques such as neural networks, rule induction, genetic algorithm and case-based reasoning are finding application in a wide variety of fields such as computer vision; cloud computing, econometrics and medicine. This paper highlights the cost estimation models that have been proposed and used successfully.

Keywords: surveys, software cost estimation methods, Neural networks, Machine learning.

1. INTRODUCTION

The most important task of software estimation is setting realistic expectations. Unrealistic expectations based on inaccurate estimates are the single largest cause of software failure. Accurate software cost estimates are critical to both developers and customers. Software Engineering cost models and estimation techniques are used for a number of purposes such as Budgeting, Tradeoffs and risk analysis, project planning and control, software improvement investment analysis, staff allocation, project bidding and proposal etc. The main aim of this paper to provide a survey of various emerging estimation techniques[1][2][3].There are several methods and models[4][5][6][7][8], this paper describes for the software cost estimation, but which method is suitable for cost estimation it's very difficult to decide. To solve this type problem it is very necessary to know about the software cost estimation methods and models. Several software estimation techniques have there own advantages and disadvantages.

2. BACKGROUND

The Software Estimation becomes the essential part of the software development process. Software project failures have been an important issue for software developers. Results shows that approximately between 30% and 40% of the software projects are completed and the others fail. Traditionally, researchers estimated software effort by means of off-the shelf algorithmic models such as COCOMO (Boehm, 1981) [9]where effort is expressed as a function of anticipated size; or have developed local models using statistical techniques such as stepwise regression .There is lot of work have been conducted by several authors in the field of model based estimation techniques, expertise based, learning oriented, dynamic based[10],regression based[11][12][13][14] and composite Bayesian such as cocomo 2.Common model based techniques are SLIM, cocomo, checkpoint and SEER. Delphi and rule based are come under expertise based estimation techniques. Recently, attention has turned to a variety of machine learning (ML)[15] methods to predict software development effort.. Artificial neural nets (ANNs)[16][17][18],genetic algorithms[19][20][21], case based reasoning (CBR)[22] and rule induction (RI),estimating by analogy[22],clustering techniques[23] are examples of such methods. Several researchers have applied the neural networks approach to estimate software development effort [24][25][26][27][28]. Wittig and Finnie [29][30] describe their use of back propagation learning algorithms on a multilayer perceptron in order to predict development effort and cost. They consider ANNs as promising techniques to build predictive models, because they are capable of modelling non linear relationships. There are many factors that should be considered in the selection of a estimation technique, the most common aim is to maximize the accuracy in prediction; however other issues are also important such as robustness.

This paper analysis various emerging estimation techniques as no single technique is best for all situations.

3. ESTIMATION TECHNIQUES

Generally, there are many techniques or methods for software cost estimation, which are divided into various categories, but the information industry wants a simple and accurate estimation method for their work. There are many promising estimation techniques that are capable of better estimation such as Case-Based Reasoning (CBR) Artificial Neural Networks (ANN), Decision Trees (DT), Bayesian Networks (BN), Genetic Algorithms (GA), Genetic Programming (GP), Association Rules (AR), Regression methods. Among them ANNs and Case based reasoning are most used.

4. ARTIFICIAL NEURAL NETWORKS

ANNs posses' large number of highly interconnected processing elements called neurons, which usually operate in parallel and are configured in regular architectures. Each neuron connected with the other by a communication link and each connection link is associated with weights which contain information about the input signal. The neuron computes a weighted sum of its inputs and generates an output if the sum exceeds a certain threshold. The process continues until one or more outputs are generated. These are estimation models that can be "trained" using historical data to produce ever better results by automatically adjusting their algorithmic parameter values to reduce the delta between known actual and model predictions.[31][32]

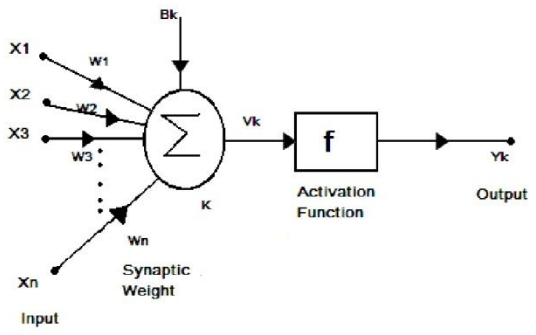

Figure 1: An Artificial Neuron

4.1 BACK PROPAGATION ALGORITHM

This is the most widely used algorithm for supervised learning with multilayered feed forward networks. Once the network has been built, the model must be trained by providing it with a set of historical project data

Inputs and the corresponding known actual values for project schedule. The model then iterates on its training algorithm, automatically adjusting the parameters of its estimation functions until the model estimate and the actual values are within some pre-specified value. Wittig has reported accuracies of within 10% for a model of this type when used to estimate software development effort, but caution must be exercised when using these models as they are often subject to the same kinds of statistical problems with the training data as are the standard regression techniques used to calibrate more traditional models[33][34].

4.2 RESILIENT BACK PROPAGATION

RPROP is used for performing supervised batch learning for multilayered feed forward Networks. The basic principle of RPROP is to eliminate the harmful influence of the size of the partial derivative on the weight step. RPROP modifies the size of weight step, directly by introducing the concept of resilient update values. As a result, the adaptation effort is not deteriorated by un-foreseeable gradient behaviour.[35]

4.3 GRADIENT DESCENT LEARNING

This algorithm tries to minimize the error E between actual and desired output by adjusting the synaptic weights between the neurons by an amount proportional to the first derivative of the mean squared error with respect to the synaptic weight. Thus if Wij is the weight update of the link connecting the ith and jth neuron of the two neighbouring layers, then Wij is defined as

$$Wij = \eta\ \partial E/\partial Wij$$

Where, η is the learning rate parameter and $\partial E/\partial Wij$ is the error gradient with reference to the weight Wij.

4.4 DELTA RULE

Delta rule is the special case of Gradient Descent Learning. Delta rule is also referred as the Widrow-Hoff Learning Rule. According to this learning rule the mechanism for weight modification during the training process acts in an appropriate way in order to minimize the difference between the desired output and the actual output produced by the processing elements. It is also called the Least Mean Square Learning Rule, because it is used to minimize the mean squared error of that difference.

4.5 CASCADE NEURAL NETWORK

Cascade NN is a feed-forward neural network [36] where the first layer will get signal from input values. Each subsequent layer will receive signal from the input and all previous layers. Cascade-correlation (CC) is an architecture and generative, feed-forward, supervised learning algorithm for artificial neural networks. Cascade- Correlation begins with a small network, then automatically iterates and adds new hidden units one by one creating a multi-layer structure. Cascade-correlation performs better then the Back-propagation learning algorithm. It has less error values, so accuracy is high in Cascade-correlation [37].

4.6. SINGLE LAYER FEED FORWARD NETWORK ARCHITECTURE

Ch. Satyananda Reddy and KVSVN Raju proposed a single layer feed forward network architecture that is constructed to accommodate the COCOMO II model [38]. This model maps COCOMO model to a neural network with minimal number of layers and nodes to increase the performance of the network. The neural network that is used to predict the software development effort is the single layer feed forward neural network with the identity function at both the input and output units. Two different learning algorithms back propagation and RPROP are used to train the network to find the best learning algorithm. It was observed that the neural network model with RPROP provided significantly better cost estimations than the estimation done using COCOMO model. The use of the proposed neural network to estimate in(PM) in

the “(2)” requires 23 input nodes in the input layer which are in(EMi) for I = 1 to 17 , SFi * 0.01 * in(size) where i= 1 to 5 and 1.01* in (size) and a bias = in(A).
“Fig.2”

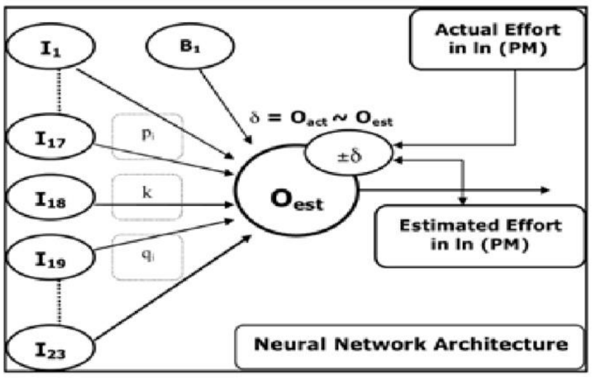

Figure 2 Neural Network Architecture [2]

4.7 RBF Neural Network with Incremental Learning.

A Radial basis function (RBF) neural network with a new incremental learning method based on the regularized orthogonal least square (ROLS) algorithm is proposed for face recognition [39]. It is designed to accommodate new information without retraining the

initial network. For avoiding the expensive reselecting process, the selection of repressors for the new data is done locally. In addition, it accumulates previous experience and learns updated new knowledge of the existing groups to increase the robustness of the system. The conventional ROLS involves retraining the whole neural network when new training data are added. The proposed algorithm achieves comparable recognition accuracy and requires lesser training time and hidden neurons compared to the conventional ROLS-based RBF neural network. The experimental results have shown that the proposed method achieves higher recognition accuracy as compared to the IPCA, ILDA, I-ELM, and EI-ELM with much lower computational complexity. It also achieves a comparable recognition rate to the online boosting in the AR database.

4.8 MULTILAYER PERCEPTRON (MLP) MODEL

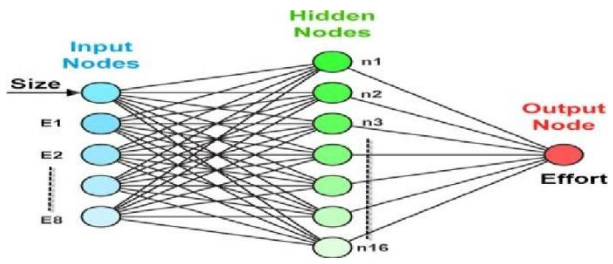

Figure 3 Multilayer Perceptron Model [3]

This section presents the MLP neural network model (fig: 3). each layer is composed of nodes and in the fully connected network considered here each node connects to every node in subsequent layers. Each MLP is composed of a minimum of three layers consisting of an input layer, one or more hidden layers and an output layer. The main inputs to the proposed MLP model are software size and team productivity represented by the eight environmental factors (E1–E8) The output of the model is software effort. The structure of the proposed neural network is depicted in Fig.3 the network will stop training when the number of epochs reaches 250 or when the Mean Squared Error (MSE) becomes zero.

The time was set to “infinity” which indicates that the training time does not have a control on when the training should stop. Ali Bou Nassifa, Danny Hob used algorithm to train the MLP model was Levenberg– Marquardt back propagation. They find that When the small dataset was used, it is clear that the MLP model outperforms the log-linear regression model as well as the other models.MLP with a lot of hidden layers performs more better than with one or two hidden layer specially regarding the output performance parameters. Also, when you compare between the o/p of MLP and the mathematical formula, you will found thatThe o/p performance parameter are best suited with the experimental [31]

5. CBR

Case based reasoning is a Machine learning model. It has the capability to model the past experience of experts in many fields of problem solving, i.e., by adapting past cases which appear similar to the present problem. CBR, originating in analogical reasoning, and dynamic memory and the role of previous situations in learning and problem solving, has received much attention recently. Cases are abstractions of events (solved or unsolved problems), limited in time and space. Aarmodt and Plaza [40] describe CBR as being cyclic and composed of four stages:

i) Retrieve the most similar case or cases, i.e., previously developed projects.

ii) Reuse of the retrieved cases to find a solution to the problem

iii) Revise the proposed solution

iv) Retention of the solution to form a new case

The clear-cut advantage that CBR has over use of algorithmic models is that the use of CBR evades the need to model the domain and also possess the capability to explain its reasoning. In CBR it is possible to view such cases which are retrieved as similar to the target case and to view the adaptation strategies that operate on the retrieved cases which results in the particular prediction. CBR also allows manual adaptation so that an expert working in this can extrapolate from the similar retrieved cases and thus adjust the recommended solution if felt necessary. In recent years some tangible research in the application of CBR to effort estimation suggests that CBR can provide a practical support to software development managers [41] Examples of successful CBR tools for software project estimation include: Estor, a CBR system dedicated to the selection of similar software projects for the purpose of estimating effort, and more recently, FACE and ANGEL.

6. GENETIC PROGRAMMING

The idea of evolutionary computing was introduced in 1960 by I Rechenberg in his work Evolutionary studies.Genetic programming is one of the emerging techniques of estimation. The basic ideas used are based on the Darwinian theory of evolution, which says that genetic operations between chromosomes eventually leads to fitter individuals which are more likely to survive.GP is an extension of GA, which removes the restriction that the chromosome representing individual has to be a fixed length binary string. In general in

GP, the chromosome is some type of program, which is then executed to obtain reqd. results. According to Collin j. burgess and martin lefly GP can significantly produce better estimate of efforts that any other techniques. Jeroen Eggermonta [42] used Tree based GP, Ramped Half-and-Half Method and genetic operators for their research. In an experiment, GGGP is used because of its flexibility and incorporating background knowledge, also shows great potentials in being applied in other software engineering modelling program.[43] Genetic Programming can find a more advanced mathematical function between KLOC and effort[44].

7. APPLICATIONS OF SOFTWARE ESTIMATION TECHNIQUEs

Software effort estimation is very important task for software industry for successful completion of the project. Some important characteristics of NNs are that they exhibit mapping capabilities, their capability to generalize, processing in parallel and fault tolerance.[45] Neural networks have been successfully applied to a variety of real world tasks in industry, business and science. Applications include Accounting and finance, health and medicine, engineering[46] and manufacturing, marketing, bankruptcy prediction ,image processing, handwriting recognition ,speech recognition , product inspection and fault detection[47].CBR offers enormous advantages over the other effort estimations techniques[48]. Attempts to quantify the casual dependencies within the domain have led to the development of the various algorithmic models. The clear-cut advantage that CBR has over use of algorithmic models is that the use of CBR evades the need to model the domain and also possess the capability to explain its reasoning [49]] Genetic programming have been used in many areas including Neural Network Optimization, Image Analysis, Generation of a knowledge base for expert systems, Fuzzy Logic Control, hardware control etc.

8. DISCUSSION

This paper has presented an overview of a variety of software estimation techniques, providing an overview of several popular estimation models currently available. Today, almost no model can estimate the cost of software with a high degree of accuracy. No one method or model should be preferred over all others. For a specific project to be estimated, which estimation methods should be used depend on the nature of the project. According to the weaknesses and strengths of the methods, you can choose some methods to be used. This paper elaborates various emerging software effort estimation, ANNs based techniques, Genetic programming and Case based reasoning, among them CBR and NNs are used most frequently. Till date it is proved that these models provide better results than other estimation models like algorithmic methods, estimating by analogy, expert judgment method, top-down method, and bottom-up method. Findings [50]
Show that ANNs seems to be the more accurate than CBR and GP. A number of performance comparisons between neural and traditional estimation techniques have been made by many studies. In addition, several computer experimental evaluations of neural networks for classification problems have been conducted under a variety of conditions

The future work is to study new software cost estimation methods and models that can be help us to easily understand the software cost estimation process.

9. ACKNOWLEDGEMENT

Many thanks to Mr. Vishal Gupta Assistant Professor and Rohit kumar Assistant Professor in UIET, Pan jab University Chandigarh, for doing this literature Review.

Reference

[1] Kjetil Molokken and Magne Jorgensen,” A Reviewof Surveys on Software Effort Estimation.”

[2] Narendra Sharma, Aman Bajpai, Mr. Ratnesh Litoriya,” A comparison of software cost estimation methods: A Survey,”2012 The International Journal of Computer Science & Applications.

[3] Hareton Leung, Zhang Fan,” Software Cost Estimation.”

[4] Chetan Nagar, Anurag Dixit,” Software Efforts andCost Estimation with a Systematic Approach,” JULY 2011 Journal of Emerging Trends in Computing and Information Sciences

[5] Jovan Zivadinovic, Zorica Medic,” methods ofeffort estimation in software Engineering,” international symposium engineering management and Competitivenes, 2011, Serbia

[6] Bernhard Peischl, Mihai Nica, Markus Zanker,Wolfgang Schmid,” Recommending effort estimation methods for software project management, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,2009.

[7] Vahid Khatibi, Dayang N. A. Jawawi,” Software Cost Estimation Methods: A Review,” Journal of Emerging Trends in Computing and Information Sciences 2010-11

[8] Barry Boehm, Chris Abts and Sunita Chulani,” Software development cost estimation approaches – A survey,” Annals of Software Engineering 2000

[9] Boehm, B. W, & Valerdi, R. “Achievements and challenges in cocomo-based software resource estimation”, IEEE Software, 2008

[10] Madachy, “A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment,” Ph.D. Dissertation, University of california

[11] Donald F. Specht,” A General Regression Neural Network,” IEEE transactions on neural networks,1991

[12] Roheet Bhatnagar, Vandana Bhattacharje,”Software Development Effort Estimation – Neural Network vs. Regression Modeling Approach,” International Journal of Engineering Science and Technology 2010

[13] Ali Bou Nassifa, Danny Hob,” Towards an early software estimation using log-linear regression and a multilayer perceptron model,” The Journal of Systems and Software (2012)

[14] Cuauhtemoc Lopez-Martin, Claudia Isaza &Arturo Chavoya,” Software development effort prediction of industrial projects applying a general regression neural network,” Empir Software Eng (2012)

[15] Jianfeng Wen,Shixian Li,Changqin Huang,”Systematic literature review of machine learning based software development effort estimation models, “Information and Software Technology (2012)

[16] Jaswinder Kaur, Satwinder Singh, Dr. KaranjeetSingh Kahlon, Pourush Bassi,” Neural Network-ANovel Technique for Software Effort Estimation,” International Journal of Computer Theory and Engineering 2010

[17] Stephen G. MacDonell, Martin J. Shepperd,” Combining techniques to optimize effort predictions in software project management The Journal of Systems and Software (2003)

[18] Heejun Park, Seung Baek," An empirical validation of a neural network model for software effort estimation," Expert Systems with Applications(2008)

[19] Kavita Choudhary," GA Based Optimization of Software Development Effort Estimation," IJCST, September 2010

[20] Adriano L.I. Oliveira , Petronio Braga, Márcio L," GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," Information and Software Technology(2010)

[21] Collin j. burgess and martin lefly," can genetic programmin improve software estimation. A review," information and software tech. 2001

[22] Manpreet Kaur, Sushil Garg," Analysis of Neural Network based Approaches for Software effort tEstimation and Comparison with Intermediate COCOMO," International Journal of Engineering and Innovative Technology June 2012

[23] Anita Lee, Chun Hung Cheng, Jaydeep," Software development cost estimation: Integrating neural network with cluster analysis." Information & Management 1998

[24] Ali idri, taghi, M. khoshgoftaar, Alain abran,"can neural network be easily interpreted in software cost estimation?

[25] Ch.Satyananda Reddy and KVSVN Raju," An Optimal Neural Network Model for Software Effort Estimation," DENSE Research Group

[26] Jagannath Singh and Bibhudatta Sahoo," Software Effort Estimation with Different Artificial Neural network." "2nd National Conference- Computing, Communication and Sensor Network,"2011

[27] Jitendra, Vikas, Kuldeep, Samiksha," Cost prediction using Neural Network Learning Techniques," IJCSMS International Journal of Computer Science and Management Studies, 2011

[28] Diplomarbeit," Effort Estimation of Software Development Projects with Neural Networks," Hannover, den 2007

[29] G. R. Finnie, G. E. Wittig and J-M. Desharnais," A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models," Queensland 4229, Australia and Quebec, Canada

[30] G. Witting and G. Finnie, "Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development effort", J. Information Systems

[31] Bogdan m wilamowsky," NN architectures and learning, "fellow member IEEE, USA

[32] I.F Barcelos Tronto, J.D. Simoes da Silva, N. Sant'Anna," The artificial neural networks model forsoftware effort estimation,"2006

[33] Sebastian Seung," Multilayer perceptrons and back propagation Learning," 2002

[34] Mikael Boden," A guide to recurrent neural networks and back propagation," Halsted University, 2001

[35] Martin Riedmiller, "RPROP- Description and Implementation Details, Technical Report", University of Karlsruhe,1994

[36] Vachik S.Dave, Kamlesh Dutta," Application of Feed-Forward Neural Network in Estimation of Software Effort," Intelligent Systems & Communication (ISDMISC) 2011

[37] Anjana Bawa, Mrs.Rama Chawla," Experimental of Effort Estimation Using Artificial Neural Network."

[38] Ch. Satyananda Reddy, KVSVN Raju, V.Valli Kumari, "Single Layer Artificial Neural Network Model for Software Effort Estimation", IEEE International Advance Computing Conference 2009

[39] Yee Wan Wong, Kah Phooi Seng and Li-Minn Ang," Radial Basis Function Neural Network with Incremental Learning for Face Recognition," IEEE transactions on systems, man, and cybernetics 2011

[40] Aarmodt and Plaza (1994)," Case-Based Reasoning: Foundational issues, Methodical Variations and System Approaches. AI Communications

[41] Ekbal Rashid, Vandana & Srikanta Patnaik," The Application of Case-Based Reasoning to Estimation of Software Development Effort."

[42] Jeroen Eggermonta," Genetic Programming," Leiden University Medical Center.

[43] y. shan, IR mckey, Y chang,"software project cost estimation using genetic programming."

[44] Sumeet Kaur Sehra, Yadwinder Singh Brar, and Navdeep Kaur," soft computing techniques for software project effort estimation," International Journal of Advanced Computer and Mathematical Science2011

[45] Girish kumar jha,"Artificial NNs and its applications."

[46] Rossana M. S. Cruz, Helton M. Peixoto and Rafael M. Magalhães," Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering."

[47] F. Farnood Ahmadia, M. J. Valadan Zoeja, H. Ebadia, M. Mokhtarzadea," the application of neural networks, image processing and cad-based environments facilities in automatic road extraction and vectorization from high resolution satellite images."

[48] Mukhopadhyay T. S. Vicinanza & M. Prietula"Examining the feasibility of a case-based reasoning model for software effort estimation."

[49] Sarah Jane Delany, Padraig Cunningham," The Application of Case-Based Reasoning to Early Software Project Cost Estimation and Risk Assessment."2000

[50] Carolyn Mair, Gada Kadoda, Martin Lefley," An Investigation of Machine Learning Based Prediction Systems."

[51] Guoqiang Peter Zhang," Neural Networks for Classification: A Survey," IEEE transactions on systems, man, and cybernetics2000."

[52] Mukta Paliwal, Usha A. Kumar," Neural networks and statistical techniques: A review of applications," Expert Systems with applications (2009)

[53] Jamal, Nazzal, Ibrahim," Multilayer Perceptron Neural Network (MLPs) For Analyzing the Properties of Jordan Oil Shale," World Applied Sciences Journal 2008.