Volume No: 1(2014), Issue No: 10 (October) ISSN No: 2348-4845

Effort Estimation for Object-Oriented System Using Artificial Intelligence Techniques

Mogili Umamaheswara Rao
Student (M.Tech) , CSE,
Gokul Institue of Technology and Science,
Visakhapatnam, India.

A.Achutharao
Asst. Prof, CSE,
Gokul Institue of Technology and Science,
Visakhapatnam, India.

Abstract:

Software effort estimation is a vital task in software engineering. The importance of effort estimation becomes critical during early stage of the software life cycle when the details of the software have not been revealed yet. The effort involved in developing a software product plays an important role in determining the success or failure.

With the proliferation of software projects and the heterogeneity in their genre, there is a need for efficient effort estimation techniques to enable the project managers to perform proper planning of the Software Life Cycle activates.

In the context of developing software using object-oriented methodologies, traditional methods and metrics were extended to help managers in estimation activity. There are basically some points approach, which are available for software effort estimation such as Function Point, Use Case Point, Class Point, Object Point, etc. In this thesis, the main goal is to estimate the effort of various software projects using Class Point Approach.

The parameters are optimized using various artificial intelligence (AI) techniques such as Multi-Layer Perceptron (MLP), KNearest Neighbour Regression (KNN) and Radial Basis Function Network(RBFN), fuzzy logic with various clustering algorithms such as the Fuzzy C-means (FCM) algorithm, K-means clustering algorithm and Subtractive Clustering (SC) algorithm, such as to achieve better accuracy.

Furthermore, a comparative analysis of software e_ ort estimation using these various AI techniques has been provided. By estimating the software projects accurately, we can have software with acceptable quality within budget and on planned schedules.

Keywords:

Software effort estimation, Class point approach, ANN, KNN, RBFN, Fuzzy Logic

I. INTRODUCTION:

Project Management is the process of planning and controlling the development as a system within a speci_ed time frame at a minimum cost with the right functionality. Much software fails due to faulty project management practices. Therefore, it is important to learn di_erent aspects of software project management. Key features of Project Management.

- Project Scheduling.
- Staffing.
- Monitoring and control.
- Project Estimation.
- Risk Management.
- Report Generation.

Among all these Projects, Estimation is the most challenging task. Project estimation involves size estimation, e_ort estimation, cost estimation, estimation, time estimation, sta_ng estimation. First, we determine the size of the product. From size estimation, we determine the e_ort needed. From e_ort estimation, we can determine product duration and cost.

1.1. Class Point Analysis

The class point approach was introduced by Gennaro Costagliola et al. in 1998 [1]. This was based on the function point analysis approach to represent the internal attributes of a software system in terms of counting. The idea using the Class Point Approach is the quantification of classes in a program similar to the FP measure, where the basic unit is function. It has been derived from the observations that in the procedural model, the basic programming units are functions or procedures whereas, in case of an object-oriented model, the logical building blocks are classes. The Class Point size estimation process is structured into three main phases, corresponding to similar phases in the function point approach, i.e.,

• Information processing size estimation: { Identification and classification of classes { Evaluation of complexity level of each class { Estimation of the Total Unadjusted Class Point .

• Technical complexity factor estimation.

• Final Class Point evaluation During the first step, the design specifications are analyzed in order to identify and classify the classes into four types of system components, namely Problem Domain Type (PDT), Human Interaction Type (HIT), Data-Management Type (DMT), and Task Management Type (TMT).

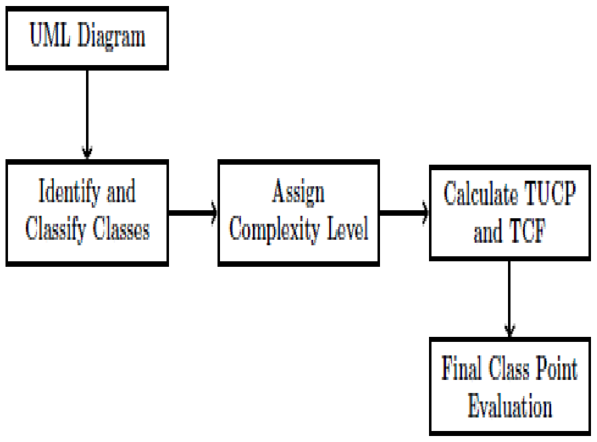

Figure 1.1: Steps to Calculate Class Point

II. ADAPTIVE REGRESSION TECHNIQUES:

2.1 Introduction:

The e_ort involved in developing a software product plays an important role in determining the success or failure. In the context of developing software using object-oriented methodologies, traditional methods and metrics were extended to help managers in e_ort estimation activity. Software project managers require a reliable approach for e_ort estimation. It is especially important during the early stage of the software-development life cycle. In this chapter, the main goal is to estimate the cost of various software projects using class point approach and optimize the parameters using various types of adaptive regression techniques such as Multi-Layer Perceptron (ANN), K Nearest Neighbor Regression (KNN) and Radial Basis Function Network(RBFN) to achieve better accuracy. Furthermore, a comparative analysis of software e_ ort estimation using these various adaptive regression techniques has been provided. By estimating the software projects accurately, we can have software with acceptable quality within budget and on planned schedules.

2.1.1 Multi-Layer Perceptron (MLP):

MLP is a feed-forward neural network with one or more layers between input and output layer. Feed-forward means that data ows in one direction from input to output layer (forward). The back propagation learning (BPA) algorithm is basically used to train this type of model.

2.1.2 K Nearest Neighbor Regression (KNN):

K Nearest Neighbor Regression (KNN) is presented by LUC P. DEVROYE [35] in the year 1978. In pattern recognition, the KNN is a method for classifying objects based on closest training examples in the feature space. It is non-parametric and lazy algorithm. In this case, an object is classi_ed by a majority vote of its neighbors, with the object being assigned for the class most common for its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned for the class of its nearest neighbour.

2.1.3 Radial Basis Function Network (RBFN):

Radial basis function emerged as a variation of multi layer perceptron technique [19]. The theory of function approximation helps in deriving the idea of RBFN. The architecture of the RBFN is quite simple. An input layer which consists of a source's nodes; a hidden layer in which each neuron computes its output using a radial basis function, which is in general a Gaussian function, and an output layer which builds a linear weighted sum of hidden neuron outputs and supplies the response from the network (effort).

2.2 Proposed Approach:

The proposed work is based on data derived from forty student projects [1] developed using Java's language and intends to evaluate software-development e_ort. The use of such data on the validation process has provided initial experimental evidence on the e_ectiveness of the CPA. These data are used in the implementation of various adaptive methods for regression such as MLP, KNN and RBFN system model. The calculated result is then compared to measure the accuracy of the models.

2.3 Model Design Using Multi-Layer Perceptron:

This technique uses one parameter. This parameter sets the number of hidden neurons to be used in a three-layer neural network. The number of neurons used is directly proportional to the training time.

The values are typically ranges between 2 to 40, but it can be increase up to 1000. While implementing the normalized data set using Multi-Layer Perceptron technique for a di_erent number of hidden neurons, the following results have been obtained. The Table-2.3.1 provides minimum NRMSE value obtained from Training set and Test set using the Multi-layer Perceptron technique for each fold for a speci_c number of hidden neurons. Hence the average over the NRMSE values for training set and test set is treated as the _nal result. The proposed model generated using the Multi-Layer Perceptron technique is plotted based upon the training and testing sample as shown in Figure-2.1. From Figure-2.1, it has been observed that the predicted value is highly correlated with actual data.

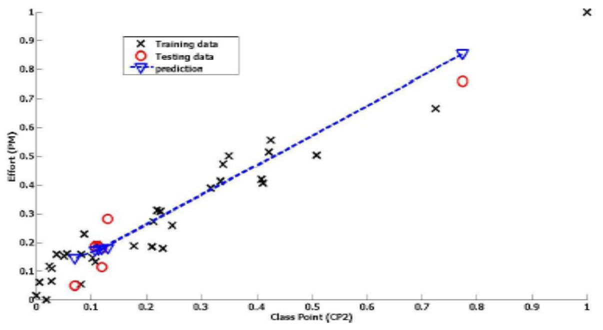

Figure 2.1: Multi-Layer Perceptron based Effort Estimation Model

III. TSK-FUZZY LOGIC SYSTEM:

3.1 Introduction:

The success of software development depends very much on proper estimation of e_ort required to develop the software. There are basically some points approach, which are available for software e_ort estimation such as Function Point, Use Case Point, Class Point, Object Point, etc. In this chapter, to estimate the e_ort of various software projects using Class Point Approach. The parameters are optimized using various AI techniques such as fuzzy logic to achieve better accuracy.

3.1.1 Fuzzy Logic System:

Fuzzy sets were introduced by L. A. Zadeh (1965) [23]. This technique is used to represent and manipulate non-precise data, but rather fuzzy. This technique provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. Fuzzy system consists of three main components: fuzzi_cation process, inference from fuzzy rules and defuzzication process.\

3.2 Methodology Used:

To implement the Fuzzy system and to _nd the number of rules di_erent types of the clustering algorithm are used that has been described in the following section. 3.2.1 Subtractive Clustering (SC) The subtractive clustering technique is proposed by Stephen L. Chiu [30] in 1994. Clustering has been often exercised as a preprocessing input phase used for the design of the RBF neural networks. To use subtractive clustering, four parameters should be pre-initialized [29] These parameters are Hypersphere cluster radius in data space, Squash Factor (_), Reject Ratio (__), Accept Ratio (_). Hypersphere cluster radius in data space de_nes neighborhood data points outside this radius has little inuence upon the potential.

3.2.1 Fuzzy C-Means Clustering (FCM):

Fuzzy C-Means clustering (FCM), also known as ISODATA, is a data clustering algorithm in which each data point belongs to a cluster, to a degree, speci_ed by a membership grade. It is _rst developed by Dunn [31] and improved by Bezdek [32]. FCM employs fuzzy partitioning such that a given data point can belong to several groups in the degree of belongings speci_ed by membership grades between 0 and 1. However, FCM still uses a cost function which is to be minimized while trying to partition the data set.

3.2.2 K-Means Clustering:

The K-means clustering (Hard C-means clustering), is a crisp clustering algorithm based on _nding data clusters in a data set such that a cost function of dissimilarity measure is minimized:

IV. EXPERIMENTAL APPARATUS AND PROCEDURE:

The following Table-4.1 shows the values of the constants p0 and p1 for each generated fuzzy rule in TSK based fuzzy model using the subtractive clustering algorithm for CP2 in one fold.

Table 4.1: Type-1 TSK Fuzzy Model Developed Using Subtractive Clustering

Algorithm for CP2.

Fuzzy Rules	if x, then $z = p_1 \times x + p_0$
Rule - 1	if $x = exp(-\frac{1}{2}(\frac{x-0.1024}{0.3780})^2)$, then $z = 0.0075 \times x + 0.1460$
Rule - 2	if $x = exp(-\frac{1}{2}(\frac{x-0.3357}{0.3780})^2)$, then $z = 0.00765 \times x + 0.4109$
Rule - 3	if $x = exp(-\frac{1}{2}(\frac{x-0}{0.3780})^2)$, then $z = 0.00765 \times x + 0.0154$
Rule - 4	if $x = exp(-\frac{1}{2}(\frac{x-0.2140}{0.3780})^2)$, then $z = 0.1038 \times x + 0.2554$
Rule - 5	if $x = exp(-\frac{1}{2}(\frac{x-0.7243}{0.3780})^2)$, then $z = 0.3178 \times x + 0.5909$
Rule - 6	if $x = exp(-\frac{1}{2}(\frac{x-0.5079}{0.3780})^2)$, then $z = 0.4048 \times x + 0.3426$
Rule - 7	if $x = exp(-\frac{1}{2}(\frac{x-1.0000}{0.3780})^2)$, then $z = 0.7024 \times x + 0.5952$

Table 3.2: RMSE Value using FIS (SC) for different Radius:

Fold	Diff. radius	Training Set Validation Error (RMSE)	Test Set Prediction Error (RMSE)
1	0.4	0.0621	0.0762
2	0.3	0.0536	0.0959
3	0.5	0.0589	0.0909
4	0.4	0.0655	0.0589
5	0.4	0.0546	0.0884
Average		0.0590	0.0823

Table 4.3: Type-1 TSK Fuzzy Model Developed Using Fuzzy C-Means Clustering Algorithm for CP2

Fuzzy Rules	if x, then $z = p_1 \times x + p_0$
Rule - 1	if $x = exp(-\frac{1}{2}(\frac{x-0.0300}{0.3664})^2)$, then $z = 0.0010 \times x + 0.0659$
Rule - 2	if $x = exp(-\frac{1}{2}(\frac{x-0.2291}{0.3664})^2)$, then $z = 0.0346 \times x + 0.2932$
Rule - 3	if $x = exp(-\frac{1}{2}(\frac{x-0.7429}{0.3664})^2)$, then $z = 0.2878 \times x + 0.6817$
Rule - 4	if $x = exp(-\frac{1}{2}(\frac{x-0.0990}{0.3664})^2)$, then $z = 0.2949 \times x + 0.1421$
Rule - 5	if $x = exp(-\frac{1}{2}(\frac{x-0.9993}{0.3664})^2)$, then $z = 0.6470 \times x + 0.7047$
Rule - 6	if $x = exp(-\frac{1}{2}(\frac{x-0.3880}{0.3664})^2)$, then $z = 0.6890 \times x + 0.2167$
Rule - 7	if $x = exp(-\frac{1}{2}(\frac{x-0.2064}{0.3664})^2)$, then $z = 0.6940 \times x + 0.0482$

Table 4.4: Type-1 TSK Fuzzy Model Developed Using K-Means Clustering Algorithm for CP2

Fuzzy Rules	if x, then $z = p_1 \times x + p_0$
Rule - 1	if $x = exp(-\frac{1}{2}(\frac{x-0.2182}{0.3020})^2)$, then $z = 0.0159 \times x + 0.1460$
Rule - 2	if $x = exp(-\frac{1}{2}(\frac{x-0.0462}{0.3020})^2)$, then $z = 0.0255 \times x + 0.4127$
Rule - 3	if $x = exp(-\frac{1}{2}(\frac{x-0.1099}{0.3020})^2)$, then $z = 0.0262 \times x + 0.0126$
Rule - 4	if $x = exp(-\frac{1}{2}(\frac{x-0.0338}{0.3020})^2)$, then $z = 0.0307 \times x + 0.2708$
Rule - 5	if $x = exp(-\frac{1}{2}(\frac{x-0.8328}{0.3020})^2)$, then $z = 0.2974 \times x + 0.6404$
Rule - 6	if $x = exp(-\frac{1}{2}(\frac{x-0.1113}{0.3020})^2)$, then $z = 0.3236 \times x + 0.4709$
Rule - 7	if $x = exp(-\frac{1}{2}(\frac{x-0.3906}{0.3020})^2)$, then $z = 0.4942 \times x + 0.8736$

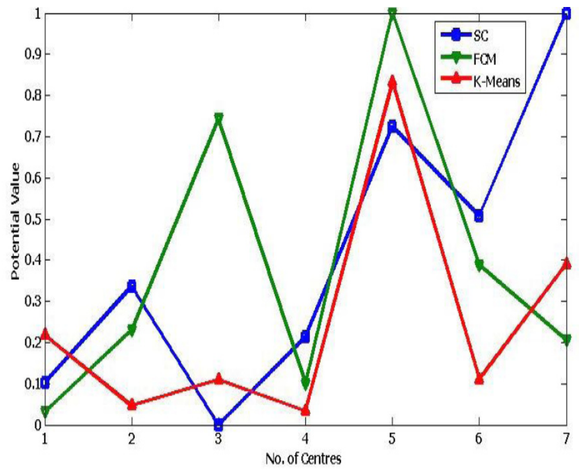

Figure 4.2: Center Points Generated Using SC, FCM and K-Means.

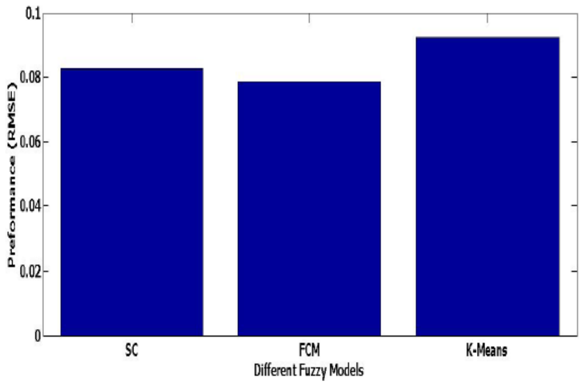

Figure 3.3: Comparison of RMSE values for SC, FCM and K-Means clustering.

Table 4.5: Comparison of RMSE Value between SC, FCM and K-Means.

	Subtractive Clustering	Fuzzy C-Means	K-Means
RMSE	0.0823	0.0785	0.0925

CONCLUSION:

Several approaches have already been de_ned in literature for software e_ort estimation. However, the CPA is one of the di_erent cost estimation models that has been widely used because it is simple, fast, accurate to a certain degree. Fuzzylogic technique is further used to _nd out the complexity level of the class and to calculate optimized class point. Then the calculated class point values are being normalized and used to optimize the e_ort estimation result. The optimization is achieved by implementing di_erent arti_cial (AI) techniques such as ANN, KNN, RBFN, and fuzzy logic system with di_erent clustering algorithm using normalized class point value. Finally, the generated minimum results of di_erent have been compared for estimating the performance of di_erent models.

The result shows that RBFN based e_ort estimation model gives less value of NRMSE. Hence it can be concluded that the e_ort estimation using the RBFN model will provide more accurate results than other AI techniques.

REFERENCES:

[1] G. Costagliola, F. Ferrucci, G. Tortora, and G. Vitiello, \Class point: an approach for the size estimation of object-oriented systems," Software Engi- neering, IEEE Transactions on, vol. 31, no. 1, pp. 52{74, 2005.

[2] J. Matson, B. Barrett, and J. Mellichamp, \Software development cost estimation using function points," Software Engineering, IEEE Transactions on, vol. 20, no. 4, pp. 275{287, 1994.

[3] F. Heemstra and R. Kusters, \Function point analysis: Evaluation of a software cost estimation model," European Journal of Information Systems, vol. 1, no. 4, pp. 229{237, 1991.

[4] W. Zhou and Q. Liu, \Extended class point approach of size estimation for oo product," in Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, vol. 4, pp. 117{122, IEEE, 2010.

[5] S. Kanmani, J. Kathiravan, S. S. Kumar, and M. Shanmugam, \Neural network based e_ort estimation using class points for oo systems," in Proceedings of the International Conference on Computing: Theory and Applications, ICCTA '07, (Washington, DC, USA), pp. 261{266, IEEE Computer Society, 2007.

[6] S. Kim, W. Lively, and D. Simmons, \An e_ort estimation by uml points in early stage of software development," Proceedings of the International Con- ference on Software Engineering Research and Practice, pp. 415{421, 2006.

[7] Z. Zia, S. Tipu, K. Khan, and S. Zia, \Software cost estimation using soft computing techniques," Advances in Information Technology and Manage- ment, vol. 2, no. 1, pp. 233{238, 2012.

[8] S. Kanmani, J. Kathiravan, S. S. Kumar, and M. Shanmugam, \Class point, based e_ort estimation of oo systems using fuzzy subtractive clustering and arti_cial neural networks," in Proceedings of the 1st India software engineering conference, ISEC '08, (New York, NY, USA), pp. 141{142, ACM, 2008.

[9] V. S Moertini, \Introduction to _ve data clustering algorithms," INTEGRAL Majalah Ilmiah Matematika dan Ilmu Pengetahuan Alam, vol. 7, no. 2, 2008.

[10] K. Bataineh, M. Naji, and M. Saqer, \A comparison study between various fuzzy clustering algorithms," EDITORIAL BOARD, vol. 5, no. 4, p. 335, 2011.

[11] K. Hammouda and F. Karray, \A comparative study of data clustering techniques," Tools of intelligent systems design. In Course Project, SYDE, vol. 625, 2000.

[12] A. Sheta, \Software e_ort estimation and stock market prediction using takagi-sugeno fuzzy models," in Fuzzy Systems, 2006 IEEE International Conference on, pp. 171{178, IEEE, 2006.

[13] A. Kaur and A. Kaur, \Comparison of mamdani-type and sugeno-type fuzzy inference systems for air conditioning system," International Journal of Soft Computing and Engineering (IJSCE), ISSN, pp. 2231{2307, 2012.

[14] A. L. Oliveira, \Estimation of software project e_ ort with support vector regression," Neurocomputing, vol. 69, no. 13, pp. 1749{1753, 2006.

[15] K. Vinay Kumar, V. Ravi, M. Carr, and N. Raj Kiran, \ Software development cost estimation using wavelet neural networks," Journal of Systems and Software, vol. 81, no. 11, pp. 1853{1867, 2008.