Automatically Estimating Software Effort and Cost using Computing Intelligence Technique

Jin-Cherng Lin , Han-Yuan Tzeng, and Yueh-Ting Lin

Dept. of Computer Science & Engineering
Tatung University
Taipei 10452, Taiwan
jclin@ttu.edu.tw

Abstract

In the IT industry, precisely estimate the effort of each software project the development cost and schedule are count for much to the software company. So precisely estimation of man power seems to be getting more important. In the past time, the IT companies estimate the work effort of man power by human experts, using statistics method. However, the outcomes are always unsatisfying the management level. Recently it becomes an interesting topic if computing intelligence techniques can do better in this field. This research uses some computing intelligence techniques, such as Pearson product-moment correlation coefficient method and one-way ANOVA method to select key factors, and K-Means clustering algorithm to do project clustering, to estimate the software project effort. The experimental result show that using computing intelligence techniques to estimate the software project effort can get more precise and more effective estimation than using traditional human experts did.

Keywords

Software Effort Estimation, Project Clustering, Computing Intelligence, Particle Swarm Optimization, K-Means Clustering.

1. INTRODUCTION

The software industry are fast to grow up, the cost of software will inevitably become one of the topics of concern. Software cost estimation affects the success of software projects. However it has many problems. The software cost estimation is with high degree of uncertainty. Overestimation or underestimation might occur to happened. These problems led to the development team in budget not enough and manpower shortage, and also caused development schedule delay and the problem of poor software quality. In practice, the accuracy of software effort estimation is one of the keys to project success. For example, according to Standish Group's EXTREME CHAOS Report 2001, of 30,000 applications development projects, 23% of project failures, 49% of the project was being challenged, only 28% of the project is successful [2]. The failure of software projects for IT companies will lead to the financial losses and loss reputation

Sundarapandian et al. (Eds): CoNeCo,WiMo, NLP, CRYPSIS, ICAIT, ICDIP, ITCSE, CS & IT 07,
pp. 381–392, 2012. © CS & IT-CSCP 2012 DOI : 10.5121/csit.2012.2437

of company. Therefore, how to accurately assess the effort of the project development is an important key to the successful projects.

In the past time, the IT companies estimate the work effort of software projects by human experts, and use some statistics method and fixed parameters to make the result more precisely. Earlier study for software effort estimation, COCOMO mostly bases on the three modes as the cluster, namely Organic Mode, Semidetached Mode, and Embedded Mode [1]. However, the outcomes are always unsatisfying the management level.

In recent years, the computing intelligence techniques have been grown. It becomes an interesting topic if computing intelligence techniques can do better in this field. Currently, some researches combined with neural networks [11], genetic algorithms, differential evolution algorithm [15], and intelligent gray theory [14] apply to software effort estimation and calculation of parameters optimization. However, their approaches still based on the original COCOMO model‘s law expert classified the cluster of three modes,

Our research also focuses on applying computing intelligence to software effort estimation. However, the clustering results will be different from the COCOMO model’s law expert classified the cluster of three modes, but through the K-Means clustering of the group. After project clustered, the Particle Swarm Optimization (PSO) algorithms [8] is applied for parameter optimization. Comparing with genetic algorithms and differential evolution algorithm [15], Particle Swarm Optimization has no complex mating and mutation. It is natural selection, simple, and fast convergence. The experimental result shows that clustering algorithms combined with the optimal algorithm can be applied to software development effort assessment issues and get more precisely outcome.

2. LITERATURE REVIEW

2.1 COCOMO

COCOMO (Constructive Cost Model) is developed by Prof. Barry W. Boehm, who first published non-linear model of assessment methods in 1981. COCOMO categorizes the projects into three different levels of details, namely, Basic model, Intermediate model, and Detail model [1]. It is mainly based on past experience of software projects to assess the software effort. It is still widely used in software effort prediction and assessment of manpower.

The COCOMO model accords to the following equation to calculate the software effort/costs:

$$C = A\,S^{B} \times \prod_{i=1}^{15} x_i \qquad \text{(eq. 1) [1]}$$

Where C is estimated software effort in man-month, A and B for the estimated parameters, S for the software size in thousands of lines of executable code (KLOC), and then multiplied by the product of 15 project complex factors, x_i is adjustment factor value.

2.2 Pearson product-moment correlation coefficient [3]

Pearson analysis can calculate the Pearson correlation index, a linear correlation value between the two indicator variables. Its equation as:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}}$$ (eq. 2) [3]

r is the correlation coefficient for the XY co-variance divided by the standard deviation of X and Y standard deviation of the product, and its value range between 1 and -1, the average for the X, for the Y's average.

2.3 One-Way ANOVA

One-Way ANOVA has also been applied in many fields, for example, Chen et al used it in genetic engineering [4], Tang applied it to hotel staff job satisfaction analysis [5], Ropponen applied it to software development risks [6]. In short, One-Way ANOVA compares two or more samples of the population if they are the same statistical population. ANOVA values generally determine $P<0.05$ and $F>4$, where F value compares the number of differences in average volume between groups, P value of confidence level, $P<0.05$ represents by 95 percent confidence level.

2.4 K-Means clustering algorithm [7]

K-Means clustering algorithm is a simple and efficient data clustering method. Compared to other clustering algorithms, it is relatively simple and faster. K-Means clustering of the basic concept of randomly generated in the initialization N group centers, and then calculate distance of each group and the cluster center. If distance smaller than other cluster center that classified in to the group, .

2.5 Particle Swarm Optimization algorithm

Particle swarm optimization is simulated biological behavior of the flock and genetic algorithms [8, 9, 17, 18]. Comparing with other method of search for optimal solutions, such as differential evolution algorithm [15], ant algorithms [16], Particle Swarm Optimization method is more simple and rapid. It is less than genetic algorithm and differential evolution algorithm for mating, mutation, natural selection and other complex simulation of biological evolution. It can more efficiently find the optimal solution.

3. MULTIPLE FACTORS CLUSTERING MODEL

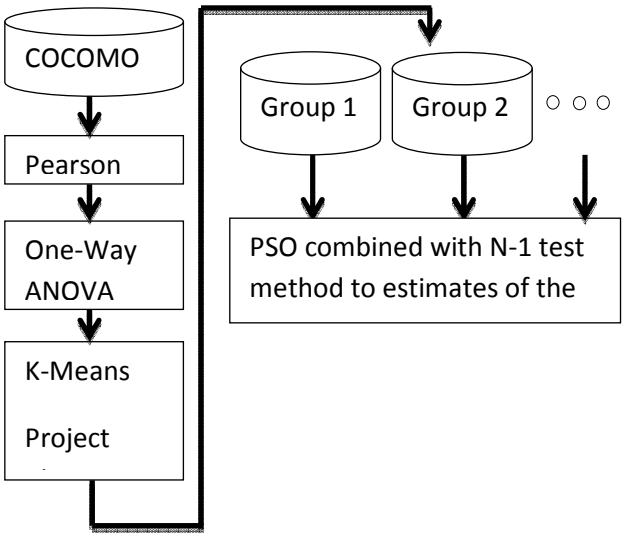

Figure 1. Multiple factors clustering model

We use the COCOMO database of 63 historical project data as evaluation data set. By Pearson correlation analysis and One-Way ANOVA to the 15 factors in the data set for correlation analysis, we choose N key factors. According to these N-factors, we apply the K-Means clustering algorithm to clustering all software projects into some groups. For each group, selected a software project data to be the prediction project, the others in the same group as training projects, using PSO algorithm to find the optimization parameters. Do it loop to make each project to be the prediction project. For example, suppose a group have five projects. In the first round, selected the first project as prediction project, using other four projects as training projects; in the second round, selected the second project as prediction project, using other four projects as training projects, and so on. While each time the optimization parameters is found, let each project to calculate the MMRE as evaluation criteria. This method is namely N-1 test method. The system architecture is shown in Figure 1.

3.1 Effort adjustment factor and correlation analysis

Instead of using human experts to do the clustering, our approach uses the clustering algorithm to do it automatically. However, there are 15 effort adjustment factors in the COCOMO data set. It is too many to calculate for the automatic clustering algorithm. We should reduce it.

Pearson correlation analysis is used to analyze the close degree of the COCOMO 15 effort adjustment factors to the software effort. We use Pearson correlation analysis by the eq. 2 to calculate the correlation coefficient of 15 factors, we select according to the slope of the correlation coefficient is positive ($r > 0$) and the high degree of relationship factors as K- means clustering based. One-Way ANOVA analysis methods is also used in more than three population

differences. We set P<0.05 and F>4 as selection standards, and select high F value of factors as base of project clustering.

3.2 K-Means Software Project Clustering

Selected key adjustment factors, using Pearson analysis and One-way ANOVA methods, is used as the coordinates of multiple dimensions, In short, the two dimensions of the data used for clustering, coordinates can be used as two-dimensional X, Y coordinates to do clustering.

3.3 Parameter Optimization using PSO

Using PSO to optimize the parameters of COCOMO, each particle contains two dimensions X and Y coordinates, X and Y are A and B parameters in the COCOMO model equation in eq. 2, respectively. Using PSO for optimal characteristics finds the best value of A and B parameters as the prediction project parameters. First, X and Y coordinates are randomly generated 40 particles of range between 0 and 1 in a two-dimensional space, and give each particles random initial speed. Then the X and Y coordinates of particles as a predictor parameters, and use MMRE as fitness Value. Each particle must be recorded optimal solution that on their through path, the solution called the local optimal solution (Pbest). Each particle must also have social behavior that each particle to find the optimal solution in the current search, called a global optimal solution (Gbest).

$$V_{id} = w \times V_{id} + c_1 \times Rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{gd} - x_{id}) \quad \text{(eq. 3) [18]}$$

$$x_{id} = x_{id} + V_{id} \quad \text{(eq. 4) [18]}$$

When the particles are getting Pbest and Gbest, using the equation (4) update each particle's flight speed. The next particle coordinates for the current location coordinates add updated flight speed as the new coordinates (eq. 4). In the eq. 3, which P_{id} is Pbest, P_{gd} is Gbest, C_1, C_2 is learning constant, in general, learning constant can be set to 2, and w is inertia weight, usually set to decimal that between 0 and 1, Rand() is between 0 and 1 random numbers [8]. This study tested the weight of inertia to 0.05 as the more appropriate inertia weight. The algorithm steps are as follows:

Step 1. Randomly generated 40 particles and give initial velocity in

two-dimensional space.

Step 2. Calculate the fitness of each particle.

Step 3. Update Pbest of each particles.

Step 4. Update Gbest of each particles .

Step 5. Update velocity of each particles using eq. 3.

Step 6. Update position of each particles using eq. 4.

Step 7. Repeat Step 2 to Step 6 until the stop condition.

3.4 Performance indicators

It's usually using Mean of MRE (MMRE) and Prediction level (Pred) as accuracy reference value In the research of software effort estimation. In this study, using the Pred and MMRE as accuracy reference value.

- MMRE
  Software effort estimation in the assessment of evaluation criteria commonly used Mean
- Magnitude of Relative Error (MMRE), the formula as equation (6) below.

$$MMRE = \frac{1}{N}\sum_{i=1}^{i=N}\frac{|actual\_effort - predicated\_effort|}{actual\_effort} \quad \text{(eq. 5) [11]}$$

In this study, MMRE for the PSO algorithm as the effort estimates of fitness value and evaluation criteria. MMRE value is the smaller that the prediction effort closer the actual effort. Which actual_effort is actual effort , predicated_effort is prediction effort ， N is number of project，i is NO. i project [10, 11].

- Pred(x)
  Prediction level is used in software project effort assessment. The accuracy of the assessed value of less than set percentage of the total assessed value accounting. The Pred(x) equation is as follows:

$$PRED(x) = \frac{k}{N} \quad , \quad k = \#\ (\forall i, MRE_i \le x) \quad \text{(eq. 6) [11]}$$

In this study, we will use MMRE as the main evaluation criteria. Pred (x) value is the higher the better. x is a percentage value, k is the representative of the assessed value of MRE is less than or equal to x. N is the number of all the projects [11, 12].

4. EXPERIMENTAL RESULTS

4.1 Pearson correlation analyze COCOMO adjustment factor

First, we combined all three of the original COCOMO model project type, including simple model, Semidetached Mode, embedded mode, total of 63 historical project data. 15 factors and software effort for Pearson correlation analysis(equation(3)), and generate the correlation coefficient r. The result of each factors and effort by Pearson correlation analysis, we take the adjustment factors that correlation coefficient is larger than 0.2 as clustering factors, The results (Figure 2) shows the database size (DATA) and the effort is positive correlation, followed by Modern Programming Practices (MODP) next to Required Software Reliability (RELY) and Computer Turnaround Time (TURN). Pearson analysis phase, we selected four adjustment

factors, followed by use of One-Way ANOVA selected N factors as a basis for K-means clustering.

4.2 One-Way ANOVA Analyzing COCOMO Parameters

In the One-Way ANOVA analysis phase, we using each adjustment factors and software effort to analysis, we decided to P value <0.05, F value > 4 as the selection criteria. After analysis, all

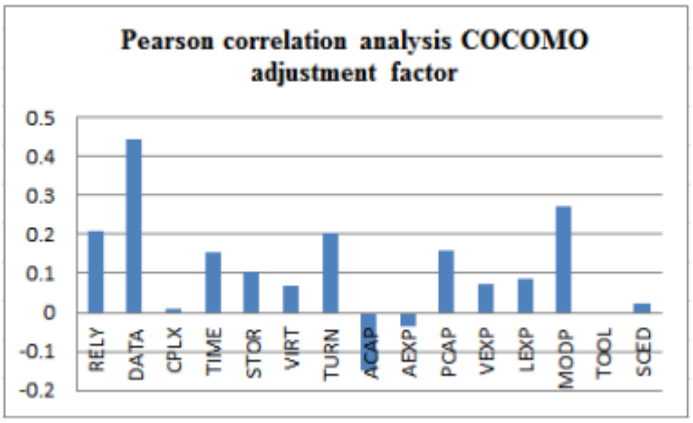

Figure 2. Pearson correlation analysis COCOMO factors

of P values are greater than 0.05. By the One-Way ANOVA analysis results (Figure 3), we will select Analyst capability (ACAP), Applications Experience (AEXP), Programmer Capability (PCAP), Programming Language Experience (LEXP).

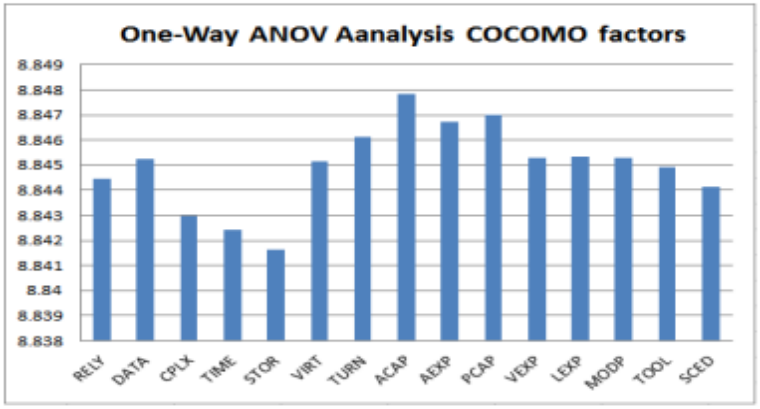

Figure 3. One-Way ANOVA analysis COCOMO factors

The Pearson correlation analysis and One-Way ANOVA analysis of the eight selected adjustment factor as clustering. The eight of factors that are Database size(DATA) Required Software Reliability(RELY), Analyst capability (ACAP) , Programmer Capability (PCAP), Applications Experience (AEXP),Computer Turnaround Time (TURN), Programming Language Experience (LEXP), Programming Language Experience (MODP) .

4.3 Software Project Clustering

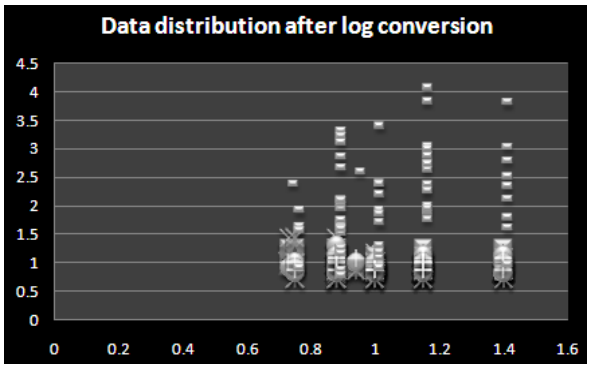

Figure 4. Data distribution after log conversion

The eight of selected key adjustment factor and effort are total of nine variables. We use K-Means clustering algorithm to do project clustering. However, the value of project effort is much higher than other eight key adjustment factor value in COCOMO data set. If we use the values of project effort directly, the outcomes of clustering is easily impacted by this outstanding value. It results that clustering is very poor. In order to make clustering more desirable, so we take the effort value to do logarithm conversion. After conversion, indeed, the clustering results are improved. It is shown in Figure 4.

Finally, each group could be assigned to the data points about 10 projects, this study tested three, four, five as fixed group of numbers. Divided into five groups led to some group's data less than five projects, so, we divided into three groups and four groups as fixed number of group for software project clustering. Clustering results of detail in Table1 and Table 2 .

Table 1. Clustering of results in three groups

Group	Number of Projects
Group1	34
Group2	19
Group3	10

Table 2. Clustering of results in four groups

Group	Number of Projects
Group1	24
Group2	8
Group3	13
Group4	18

4.4 Parameter Optimization and Performance indicators

According to the results of clustering software project, each group has been identified by particle swarm optimization to optimize COCOMO parameters and use N-1 test method to estimate effort. In this study, particle swarm optimization to 5000 generations as the stop condition, inertia weight through experimental tests to 0.05 as the most appropriate inertia weight, C_1 and C_2 are set to 2. The test results presented in Table 3 and Table 4.

Table 3. Estimation software effort in three groups

Group	MMRE	Pred(0.25)
Group1	0.2267	0.5588
Group2	0.2358	0.5789
Group3	0.2284	0.5000
Mean	0.2303	0.5459

Table 4. Estimation software effort in four groups

Group	MMRE	Pred(0.25)
Group1	0.202123804	0.6666
Group2	0.213253173	0.625
Group3	0.289618454	0.461
Group4	0.165956676	0.7222
Mean	0.217738027	0.6187

4.5 Comparative result

Prediction of effort has been much research in COCOMO, Huang and Chiu [11] using fuzzy neural network on software effort estimation, Koch and Mitlöhner using voting rules on software effort estimate [12], Ahmed et al using fuzzy logic- based framework prediction software effort[13]. Azzeh et al using Fuzzy grey relational analysis for effort prediction in multiple databases [14]. However, most research estimation software effort base on three project type of COCOMO. It is little related research that using project database of the effort adjustment factor and software effort to clustering project and estimation software effort. So, this study will be compare with based on the COCOMO three project type to estimation software effort and using statistical and intelligent computing on project clustering and estimation software effort. Related research to compare the data shown in Table 5.

Table 5. Related research data comparison table

Compare Category	Mean of MMRE	Mean of Pred(0.25)
This research with three group	0.2303	0.5459
This research with four group	0.2177	0.6187
COCOMO	0.26	0.54
ANN	0.37	0.40
FNN	0.22	0.75
FGRA	0.232	0.667

5. CONCLUSIONS

This study aimed to apply computing intelligence techniques to estimate software effort. This approach is different from the past method in that human experts categorize the software projects into various groups. We first do the factors correlation analysis, and then cluster the projects using automatic clustering algorithm. The last is parameter optimization and estimate the software effort. The experimental data shows that our approach can accurately estimate the software effort than by human experts.

ACKNOWLEDGEMENTS

This research was supported by the National Science Council, Taiwan, R.O.C. (grant NSC-100-2221-E-036-026-) and Tatung University, Taiwan, R.O.C. (grant B100-I02-031).

REFERENCES

[1] B.W. Boehm, "Software Engineering Economics" ,Englewood Cliffs ,NJ : Prentice-Hall,1981.

[2] The Standish Group Internaction "EXTREME CHAOS Report 2001" ,2001.

[3] Joseph Lee Rodgers and W. Alan Nicewander. "Thirteen Ways to Look at the Correlation Coefficient" The American Statistician, Vol. 42, No. 1 (Feb., 1988), pp. 59-66,1988.

[4] Dechang Chen, Dong Hua, Jaques Reifman, Xiuzhen Cheng "Gene Selection for Multi-Class Prediction of Microarray Data" IEEE Proceedings of the Computational Systems Bioinformatics (CSB'03), 2003.

[5] Xiuli Tang, Yingkang Gu " Research on Hotel Staff Job Satisfaction: the Case of Shanghai" IEEE Information Management and Engineering (ICIME), pp.118-120, April 2010.

[6] Janne Ropponen and Kalle Lyytinen "Components of Software Development Risk : How to Address Them? A Project Manager Survey" IEEE Transactions On Software Engineering,Vol.26, NO.2 ,2000.

[7] J. B. MacQueen "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-29,1967.

[8] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proc. IEEE Int'l. Conf. on Neural Networks, IV, 1942–1948. Piscataway, NJ: IEEE Service Center ,1995.

[9] Shi, Y. Eberhart, R. "A modified particle swarm optimizer" IEEE Evolutionary Computation Proceedings 1998 ,1998.

[10] Shan, Y. ,McKay, R.I. ,Lokan, C.J. ,Essam, D.L. "Software Project Effort Estimation Using Genetic Programming" IEEE Communications, Circuits and Systems and West Sino Expositions vol.2, pp.1108-1112, 2002.

[11] Sun-Jen Huang , Nan-Hsing Chiu "Applying fuzzy neural network to estimate software development effort" Springer Science+Business Media, LLC,pp.73-83, 2007.

[12] Stefan Koch , Johann Mitlöhner "Software project effort estimation with voting rules" Decision Support Systems Volume 46, Issue 4, March 2009, pp. 895-901, 2009.

[13] Moataz A. Ahmed, Moshood Omolade Saliu , Jarallah AlGhamdi "Adaptive fuzzy logic-based framework for software development effort prediction" Information and Software Technology Vol.47, Issue 1, 1 January 2005, pp. 31-48, 2005.

[14] Mohammad Azzeh , Daniel Neagu , Peter I. Cowling "Fuzzy grey relational analysis for software effort estimation" Springer Science + Business Media, LLC 2009, 2009.

[15] Yucheng Kao, Jin-Cherng Lin , Jian-Kuan Wu "A Differential Evolution Approach for Machine Cell Formation" IEEE Industrial Engineering and Engineering Management, 2008.,pp. 772-775,2008.

[16] Marco Dorigo, Mauro Birattari, Thomas St¨utzle "Ant colony optimization" Computational Intelligence Magazine, IEEE,pp.28-39,2006.

[17] Yahya Rahmat-Samii "Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in Engineering Eelectromagnetics" IEEE Applied Electromagnetics and Communications, 2003. ICECom 2003, 2003.

[18] Shu Wang ,Ling Chen "A PSO Algorithm Based on Orthogonal Test Design" Natural Computation, 2009. ICNC '09. Fifth International Conference,pp.190-194, 2009.

Authors

Jin-Cherng Lin received the Ph.D. degree in Information Engineering and Computer Science from National Chiao-Tung University, Taiwan, in 1989. He is currently an associate professor in the Department of Computer Science and Engineering at Tatung University, Taiwan. His research interests include software testing and validation, software quality assurance, computer network management, and computer network security. His email is jclin@ttu.edu.tw.

Han-Yuan Tzeng received his M.S. degree in Information Engineering and Computer Science from Tatung University, Taiwan, in 2010. He is currently an R&D engineer. His research interests include computing intelligence techniques, software engineering, and system design.

Yueh-Ting Lin received his M.S. degree in Mechanical and Electro-Mechanical Engineering from The National Sun Yat-Sen University, Taiwan, in 2011. He is currently an R&D engineer. His research interests include computing intelligence techniques, intelligent transformation system, and digital signal process.