**Cluster Analysis & PSO for Software Cost Estimation**

Tegjyot Singh Sethi1,CH.V.M.K.Hari2,B.S.S.Kaushal3,Abhishek Sharma3

1,3 Dept.of CSE, Gitam University, Visakhapatnam, India.

1 [tjss401@gmail.com](mailto:tjss401@gmail.com), 3 [kaushalb09@gmail.com](mailto:kaushalb09@gmail.com), [3abhisheksharma9129@gmail.com](mailto:3abhisheksharma9129@gmail.com)

2 Dept.of IT, Gitam University, Visakhapatnam, India*,*[kurmahari@gmail.com](mailto:kurmahari@gmail.com)

**Abstract--** The modern day software industry has seen an increase in the number of software projects .With the increase in the size and the scale of such projects it has become necessary to perform an accurate requirement analysis early in the project development phase in order to perform a cost benefit analysis. Software cost estimation is the process of gauging the amount of effort required to build a software project. In this paper we have proposed a Particle Swarm Optimization (PSO) technique which operates on data sets which are clustered using the K-means clustering algorithm. The PSO generates the parameter values of the COCOMO model for each of the clusters of data values. As clustering encompasses similar objects under each group PSO tuning is more efficient and hence it generates better results and can be used for large data sets to give accurate results. Here we have tested the model on the COCOMO81 dataset and also compared the obtained values with standard COCOMO model. It is found that the developed model provides better estimation of the effort.

**Keywords-**Particle Swarm Optimization (PSO), K-Means, Software Cost Estimation, Constructive Cost Model (COCOMO).

**1 Introduction**

The software industry today is all about efficiency. The provident allocation of the available resources and the judicious estimation of the requisites form the basis of any planning and scheduling activity. With the increase in the expanse and impact of modern day software projects, the need for accurate requirement analysis early in the software development phase has become pivotal . For a given set of requirements, it is desirable to cognize the amount of time and money required to deliver the project prolifically. The chief aim of software cost estimation is to enable the client and the developer to perform a cost – benefit analysis. The cost / effort estimates are determined in terms of person-months(pm) which can be easily commuted to actual currency cost. The cost of the software varies depending on both complexity and lines of code and to estimate the cost we make use of Particle Swarm Optimization on clustered data. A common approach to the estimation of the software effort is by expressing it is as a function of the project size[1] and the Effort Adjustment Factor (EAF). The equation of effort in terms of size and methodology is considered as follows:

|  |  |
| --- | --- |
| Effort = a\*(size)b \* (EAF)+c | (1) |

Here a, b, c are constants. The constants are usually determined by regression analysis applied to historical data[11]. There are a number of models proposed for tuning parameters using Neural Networks, Machine learning techniques, fuzzy techniques and Genetic algorithms[2][3][5][6][8].

PSO is a robust stochastic optimization technique [4][9][10] based on the movement of intelligent swarms . PSO applies the concept of social interaction to problem solving. It uses a number of agents (particles) that constitutes a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in an N- dimensional space which adjusts its movement. According to its own flying experience (Pbest- personal best) as well as the flying experience of other particles (Gbest –global best). The basic concept of PSO lies in accelerating each particle towards its Pbest and Gbest locations with regard to a random weighted acceleration at each time. The modifications of the particle’s positions can be mathematically modeled by making use of the following equations:

|  |  |
| --- | --- |
| Vk+1 = wVik + c1 rand1 (Pbest – Sik) + c2 rand2 (Gbest – Sik) | (2) |
| Sik+1 = Sik + Vik+1 | (3) |

Where , Sik is current search point; Sik+1 is modified search point; Vik is the current velocity;Vk+1 is the modified velocity; Vpbest  is the velocity based on Pbest;Vgbest = velocity based on Gbest; w is the weighting function; cj is the weighting factors; randj are uniformly distributed random numbers between 0 and 1. In the particle swarm optimization technique , the particles searches the solutions in the solution space within the range [-s,s] . The PSO works better when operated on datasets having similar valued objects. Hence to enhance the PSO computations we have clustered the given dataset into groups of similar projects using the K Means clustering algorithm. K-Means clustering[12] is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Here an iterative refinement method is employed to find the means(centroids) of the cluster. Any new value entered is first ascertained to be of a particular cluster and then the PSO estimation is performed on it. The parameter thus estimated gives lesser error and closer results than those without using clustering.

**2 Model descriptions**

In this model we have considered “ The standard PSO with inertia weights” which works on data clustered by using the K Means clustering algorithm. The effort is given by Equation 1, in which a, b and c are parameters to be tuned by PSO.

In PSO ,swarm behavior is used for tuning the parameters of the Cost/Effort estimation. As the PSO is a random weighted probabilistic model the previous benchmark data is required to tune the parameters. Based on that data, swarms develop their intelligence and empower themselves to move towards the solution .We initially develop clusters having similar values and then apply the PSO to each cluster individually to obtain the parameter value. The following is the methodology employed to tune the parameters in the proposed models following it.

**3 Methodology**

Here we describe the methodology for our model which uses K means clustering algorithm and implements PSO on these clusters.

**Input:** k the number if clusters, d-data set containing size of software projects, measured efforts, EAF (complexity factor).

**Output:** Set of k clusters, Optimized parameters for the clusters.

**Step 1:** Choose k objects from d as initial centroids.

**Step 2:** Assign each object to a cluster based on the minimum distance between the centroid and the value.

**Step 3:** Update the cluster mean by assigning centroid as the mean of all data values as the new centroid.

**Step 4:** Repeat the above steps until we get the stable clusters. These clusters obtained are tuned using PSO.

**Step 5:** *Initialization:* Initialize particles with random positions and velocity vectors of tuning parameters .We also need the range of velocity between [- Vmax, Vmax].

**Step 6:** *Evaluation of Fitness Function:* For each particle position with values of tuning parameters, evaluate the fitness function. The fitness function here is Mean Absolute Relative Error (MARE). The objective in this method is to minimize the MARE by selecting appropriate values from the ranges specified in step 1.

**Step 7:** *Finding the Pbest – Personal best:* If fitness (p) better than fitness (Pbest) then: Pbest = p. Here the Pbest is determined for each particle by evaluating and comparing measured and estimated effort values of the current and previous parameters values.

**Step 8:** *Finding the Gbest (global best):* Set the best of ‘Pbest’ as global best – Gbest. The particle value for which the variation between the estimated and measured effort is the least is chosen as the Gbest particle.

**Step 9:** *Update values:* Update the velocity and positions of the tuning parameters with equations (2) & (3).

**Step 10:** Repeat steps 2 to step 5 until “particles exhaust”.

**Step 11:** Give the Gbest values as the optimal solution.

**Step 12:** Stop.

**4 Model Analysis**

We have implemented the above methodology, for tuning the parameters a, b and c in “C” language. For the parameter’ a,b and c ‘the velocity and position of the particles are updated by applying the equations (2) & (3), with parameters w=0.5, c1=c2=2.0. The data is clustered by using K Means. The final allocation matrix defines sets of values in each cluster. We apply PSO individually on each cluster and output a set of parameter values for each cluster.

**5 Performance Criterion**

We consider three performance criterions which are Variance Accounted –For (VAF), Mean Absolute Relative Error (MARE) and Variance Absolute Relative Error (VARE) [7].

**6 Experimental Study**

For the study of this model we have taken 45 data values from COCOMO81 dataset, for both training and testing. We have considered 3 clusters (0, 1, 2) the clusters are indicated in Table I. By running the ‘C’ implementation of the above methodology we have obtained the following parameters for the proposed model.

**Cluster 0:**  a=0.314800; b=1.862067; c= -5.062736

**Cluster 1:**  a=0.079505; b=2.137003; c= -3.657698

**Cluster 2:** a=4.182602, b=0.963337; c= -1.449817

The Measured and Estimated Efforts Corresponding to Cluster PSO & COCOMO are given in Table I and the corresponding graph is depicted in Fig 1.The Performance Criterion is given in Table 2.

**Fig 1**. Measured Effort Vs Estimated Effort

**Table 1:** Measured (ME) and Estimated (EE) Efforts Corresponding to Cluster PSO & COCOMO

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cluster No. | Size | EAF | ME | COCOMO | EE | Cluster No. | Size | EAF | ME | COCOMO | EE |
| 0 | 16 | 0.66 | 33 | 39 | 31.22 | 1 | 28 | 0.96 | 83 | 102 | 90.80 |
| 0 | 18 | 2.38 | 321 | 214 | 157.87 | 1 | 30 | 1.14 | 87 | 130 | 126.33 |
| 0 | 20 | 2.38 | 218 | 243 | 193.19 | 1 | 32 | 0.82 | 106 | 100 | 103.67 |
| 0 | 24 | 0.85 | 79 | 108 | 94.36 | 1 | 28 | 0.45 | 50 | 47 | 40.62 |
| 0 | 13 | 2.81 | 98 | 133 | 99.89 | 2 | 4 | 2.22 | 43 | 30 | 33.85 |
| 0 | 22 | 1.76 | 230 | 201 | 170.01 | 2 | 6.9 | 0.4 | 8 | 9.8 | 9.30 |
| 0 | 13 | 2.63 | 82 | 161 | 93.16 | 2 | 3 | 5.86 | 73 | 60 | 69.18 |
| 0 | 12 | 0.68 | 55 | 33 | 16.82 | 2 | 3.9 | 3.63 | 61 | 52 | 54.88 |
| 0 | 15 | 0.35 | 12 | 20 | 12.00 | 2 | 3.7 | 2.81 | 40 | 38 | 40.00 |
| 0 | 19.5 | 0.63 | 45 | 46 | 45.00 | 2 | 1.9 | 1.78 | 9 | 10.7 | 12.37 |
| 0 | 23 | 0.38 | 36 | 33 | 36.00 | 2 | 9.4 | 2.04 | 88 | 89 | 72.43 |
| 0 | 24 | 1.52 | 176 | 193 | 172.73 | 2 | 2.14 | 1 | 7.3 | 7 | 7.25 |
| 0 | 15 | 3.32 | 237 | 239 | 156.79 | 2 | 1.98 | 0.91 | 5.9 | 5.9 | 5.90 |
| 0 | 25 | 1.09 | 130 | 145 | 132.51 | 2 | 6.2 | 0.39 | 8 | 8.4 | 8.01 |
| 0 | 21 | 0.87 | 70 | 68 | 74.30 | 2 | 2.5 | 0.96 | 8 | 8.1 | 8.26 |
| 1 | 46 | 1.17 | 240 | 212 | 328.92 | 2 | 5.3 | 0.25 | 6 | 4.7 | 3.76 |
| 1 | 30 | 2.39 | 423 | 327 | 268.86 | 2 | 10 | 3.18 | 122 | 114 | 120.79 |
| 1 | 37 | 1.12 | 201 | 238 | 196.26 | 2 | 8.2 | 1.9 | 41 | 55 | 58.88 |
| 1 | 48 | 1.16 | 387 | 239 | 357.48 | 2 | 5.3 | 1.15 | 14 | 22 | 22.53 |
| 1 | 50 | 3.14 | 1063 | 962 | 1063.00 | 2 | 4.4 | 0.93 | 20 | 14 | 14.76 |
| 1 | 40 | 2.26 | 605 | 529 | 472.89 | 2 | 6.3 | 0.34 | 18 | 7.5 | 6.92 |
| 1 | 34 | 0.34 | 47 | 44 | 47.00 | 2 | 6.7 | 2.53 | 57 | 60 | 64.67 |
|  |  |  |  |  |  | 2 | 9.1 | 1.15 | 38 | 42 | 38.92 |

**Table 2:** Performance Criterion

|  |  |  |  |
| --- | --- | --- | --- |
| Model | VAF (%) | MARE (%) | VARE (%) |
| Cluster-PSO Model | 94.6618 | 17.3564 | 3.7287 |
| COCOMO Model | 95.4722 | 20.7606 | 4.1224 |

**7 Conclusion**

Software cost estimation is based on a probabilistic model and hence it does not generate exact values. However if good historical data is provided and a systematic technique is employed we can generate better results. In this study we have proposed a new model to estimate the software effort. In order to cluster the values into similar groups we have used K-means clustering algorithm and in order to tune the parameters we use particle swarm optimization methodology algorithm. It is observed that the clustered-PSO gives more accurate results when juxtaposed with the standard COCOMO. On testing the performance of the model in terms of the error rate the results were found to be useful. This method can be applied to large datasets to generate efficient values. These techniques can be applied to other software effort models.

**References**

1. John w. Bailey and victor R.Basili:”A meta model for software development resource expenditures”, Fifth International conference on software Engineering, CH-1627-9/81/0000/0107500.75@ 1981 IEEE, PP 107-129(1981).**CH**
2. Lionel C. Briand, Khaled El Emam, Frank Bomarius,COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment, International Software Engineering Research Network Technical Report ISERN-97-24 (Revision 2).PP 1-24, (1997)
3. Tanja Gruschke,Empirical Studies of Software Cost Estimation: Training of Effort Estimation Uncertainty Assessment Skills, 11th IEEE International Software Metrics Symposium (METRICS 2005), 1530-1435/05 © IEEE(2005)
4. Magne Jørgensen, Evidence-Based Guidelines for Assessment of Software Development Cost Uncertainty, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 31, NO. 11,PP: 942-954, NOVEMBER (2005)
5. Alaa F. Sheta, ,Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects, Journal of Computer Science 2 (2): PP:118-123, (2006)
6. Martin Auer, Adam Trendowicz, Bernhard Graser, Ernst Haunschmid, and Stefan Biffl, ,Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 2,PP: 83-92 FEBRUARY (2006)
7. CH.V.M.K.Hari ,Prasad Reddy.P.V.G.D, M.Jagadeesh “Interval Type 2 Fuzzy Logic for Software Cost Estimation Using Takagi-Sugeno Fuzzy Controller ”,Proceedings of 2010 International Conference on Advances in Communication, Network, and Computing, DOI 10.1109/CNC.2010.14, 978-0-7695-4209-6/10 © IEEE (2010).
8. Magne Jørgensen and Martin Shepperd,A Systematic Review of Software Development Cost Estimation Studies, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 33, NO. 1, PP: 33-53, JANUARY (2007)
9. Riccardo Poli , James Kennedy and Tim Blackwell, Particle swarm optimization An overview, Springer, DOI 10.1007/s11721-007-0002-0, Swarm Intell PP: 33–57, (2007)
10. Felix, T. S. Chan , Manoj Kumar Tiwari, ,Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, I-TECH Education and Publishing, ISBN 978-3-902613-09-7,PP: 1- 548, (2007)
11. Jacky Wai Keung, Barbara A. Kitchenham, and David Ross Jeffery , Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 4, PP: 471-484, JULY/AUGUST (2008)
12. Wu Bin, Zheng Yi, Liu Shaohui,and Shi Zhonghi ,“CSIM: A Document Clustering Algorithm Based on Swarm Intelligence”, 0-7803-7282-4/02@2002 IEEE PP:477-482,(2002).