Research Journal of Applied Sciences, Engineering and Technology 10(3): 293-297, 2015
DOI: 10.19026/rjaset.10.2490
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 20, 2014 Accepted: February 8, 2015 Published: May 30, 2015

Research Article

Improvised Analogy based Software Cost Estimation with Ant Colony Optimization

[1]D. Manikavelan and [2]R. Ponnusamy
[1]Department of Computer Science and Engineering, Sathyabama University,
[2]Rajiv Gandhi College of Engineering, Chennai, India

Abstract: The aim of this study is to provide an efficient methodology in estimating project development cost using analogy. Cost estimation is one of the greatest challenges in software industry to be successful enough in delivering a project within the schedule and with quality. In most cases, the delivered product either loses out on the quality or the expected timeline, owing to improper and imprecise estimation of the project cost. Deriving near accurate project cost could be done using analogy, wherein previous project data set is manipulated to arrive at the accurate cost for the current project. Ant Colony Optimization (ACO) technique implemented over analogy provides a better solution to overcome the challenges faced during cost estimation. In our study, we follow a three step methodology, based on ACO to arrive at the project development cost from promised datasets. Firstly, we extract the matching projects from data set based on the nearest values of a parameter. Secondly, we identify and group projects based on sizing. Thirdly, we improve similarity measures to match our project against those in data set.

Keywords: Analogy, ant colony optimization, cost estimation, line of code, measurement error

INTRODUCTION

The practice of effort estimation is critical during software development. Without accurate estimates, project managers cannot set realistic goals for software delivery dates nor can they staff their projects in an efficient and optimal way. Software effort estimation can be done through two major approaches: estimates based on formal models such as the constructive cost model and function points or estimates based on analogy where the process of generating an estimate is based largely on past experience, expert judgments, Parkinson's law. Constructive Cost Model is an algorithmic software cost estimation model which was developed by Barry (1981). It uses a basic regression formula with parameters that are derived from historical project data. Limitations of COCOMO are that it ignores hardware issues and personnel turnover levels. Functional Point Analysis is an ISO recognized method to measure the functional size of an information system. It was developed by Albrecht (1979). The limitation of functional point is that it is performed after the creation of design specifications. The oldest metric for software projects is that of "Lines of Code" (LOC). This metric was first introduced in 1960 and was used for economic, productivity and quality studies. The limitation is lack of accountability and developer experience, Putnam's 78, SLIM is the first algorithmic cost model. It is based on the Norden/Rayleigh function and generally known as a macro estimation model (for large projects). Limitation is that estimates are extremely sensitive to the technology factor and not suitable for small projects. Estimation by analogy means creating accurate estimation for the proposed project by comparing the proposed project to similar projects. Lots of new research is being done on analogy, exploiting the essential assumption of Analog-Based effort Estimation (Kocaguneli *et al*., 2012) in this study proposed an approach to design TEAK using an easy path principle in order to avoid the computational cost of tools. Visual comparison of Software Cost Estimation model by regression error characteristic Analysis (Mittas and Angelis, 2010) in this study introduced (REC) Regression Error Characteristic which is a powerful visualization tool having interesting geometrical properties in order to easily compare and validate the different prediction models. The adjusted Analogy-Based Software effort Estimation based on Similarity (Chiu and Huang, 2007) in their article proposed an estimation model by adopting GA method used to adjust an effort based on similarity distance. Providing Statistical Inference to Analogy-Based Software cost estimation (Keung *et al*., 2008). In this study used the strength of correlation between the distance matrix of project features and the distance matrix of known effort values of data. Active Learning and Effort Estimation: Finding the Essential Content of Software Effort Estimation Data (Kocaguneli *et al*., 2013), In this study Software Effort Estimation data can be reduced to small essential

Corresponding Author: D. Manikavelan, Department of Computer Science and Engineering, Sathyabama University, Chennai, India

Table 1: Promised dataset comparison of COCOMO and functional point

S. No.	KLOC	People	Effort	Time	COCOMO (organic)			Functional point		
1	218	25	4110.20	228.34	1.05	684.840	650.60	0.012	941.76	33.73
2	101	15	6219.95	345.55	1.05	305.310	290.05	0.012	436.32	25.18
3	116	11	9200.06	511.13	1.05	353.090	335.44	0.012	501.12	26.54
4	116	8	8477.57	470.97	1.05	353.090	335.44	0.012	501.12	26.54
5	108	13	7139.35	396.63	1.05	327.570	311.19	0.012	466.56	25.83
6	95	6	13382.31	743.47	1.05	286.290	271.98	0.012	410.40	24.60
7	52	5	9073.86	504.10	1.05	152.050	144.45	0.012	224.64	19.56
8	190	19	14123.79	784.65	1.05	592.790	563.15	0.012	820.80	32.01
9	112	12	10534.59	585.25	1.05	340.320	323.30	0.012	483.84	26.18
10	27	3	3855.46	214.19	1.05	76.408	72.58	0.012	116.64	15.25
11	384	23	19315.65	1073.12	1.05	1240.960	1178.90	0.012	1658.80	41.82
12	55	17	8170.75	453.98	1.05	161.280	153.22	0.012	237.60	19.98
13	127	15	6476.65	359.79	1.05	388.330	368.91	0.012	548.64	27.47
14	159	9	19046.30	1058.13	1.05	591.670	467.09	0.012	686.88	29.92
15	256	28	15391.82	855.11	1.05	810.700	770.17	0.012	1105.90	35.85

content and the simple methods are still able to perform well on the essential content, Table 1, presents the data collected from the promised dataset. The original dataset contains data 106 data with 94 attributes; here we have worked out on 15 data with 4 attributes such as KLOC, people, effort and time for generating sample input and output. All the projects we implemented three estimation techniques and found various outputs. This output is not similar to one to one. The main reason is COCOMO model helped to estimate the project at early stage but functional point model is not like that.

METHODOLOGY

ACO is a methodology involving computing that exhibits an ability to gain knowledge and deal with new situations, such that the system is perceived to own one or more attributes of reason, such as generalization, finding, association and abstraction. Particle Swarm Optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. ACO is a technique for solving computational problems by providing the approach to resolve a problem by the shortest path. Our approach is a member of the ACO family, which is in swarm intelligence methods and consist of metaheuristic optimization. The aim of the first algorithm is to search for an optimal path in a graph, based on the behavior of ant travels in the path from the source to the destination. ACO algorithm has been applied on many combinatorial optimization problems.

In natural world, ants moves randomly, to find the food and return to their colony while laying down pheromone trails. Other ants find such a way, based on the pheromone concentration deposited on the path. They are likely not to keep travelling at random instead follow the trail, to reach the food destination. The equation one represents that to find the shortest path:

$$p = \frac{abs(a-b)+\alpha}{c} \tag{1}$$

However, when the pheromone trail starts to evaporate, the path reduces its attractive strength by the ant. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. In the shorter path, the pheromone density becomes higher than the longer ones. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. The Eq. (2) and (3) helped to find the pheromone concentration which is deposited on the path can be calculated using the formula:

$$Ph = \frac{(effort)^{\alpha}\left(\frac{1}{2people}\right)^{\beta}(time)^{\gamma}}{total\ system\ pheromone} \tag{2}$$

where,

$$TSP = \frac{no\ of\ projects * update\ constant}{pheromone\ evaporation\ (\rho)} \tag{3}$$

TSP = Total system pheromone

When one ant finds a good path from the colony to reach the food source, other ants are more likely to follow that path and positive feedback eventually leads to all the ants following a single path. The idea of ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph and obtaining the best solution.

Analogy technique has a major role in estimation in which they compare previous projects' data with the proposed project to have better solutions for all challenges encountered during software cost estimation. Analogy based cost estimation takes us through lots of ambiguities while trying to pick up and match between our project and that project in the data set. There are two different matching techniques we could choose to implement for our project, namely, probabilistic matching and sematic matching. Probabilistic matching is a statistical analysis based technique to determine the overall probability value using more than one record. Semantic matching is a technique used to identify similarity between projects based on semantics. From

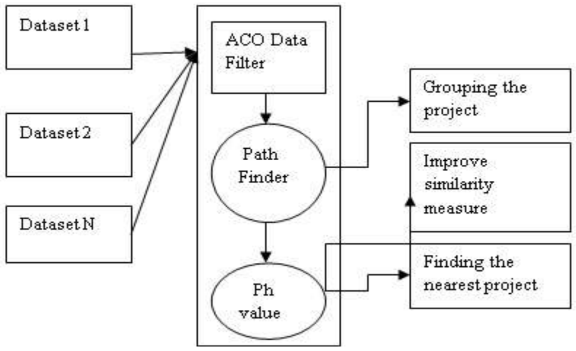

Fig. 1: ACO analogy architecture diagram

analysis, we find that sematic matching does not produce accurate results because of it's related to ontology matching. Probabilistic matching technique on the other hand, helps overcome human errors that occur during dataset retrieval. ACO is a technique based on probabilistic matching where statistical reports are derived/present for each data in the data set. Using this probabilistic value, we arrive at the exact value with the help of ACO.

Figure 1 explains that each dataset contains 70 projects and each project contains more number of attributes. ACO Data filter collects the requirements from requirement specification and filter the projects from data set based on two parameters:

- Domain name
- KDSI

ACO path finder helps to group projects based on sizing. Pheromone value improves the degree of similarity measure and helps to find the nearest matching data. The proposed algorithm as follows:

Step 1: Read data from dataset which are not null.

Step 2: Assign a variable for relevant parameters in the data.
Let a be effort
Let b be time
Let α, β and γ be a positive constant
Let p be percentage of pheromone concentration
Let c be the number of projects taken from dataset for our study.

Step 3: Arrive at the feasible path which is used to choose data from dataset that matches Current project using the formula from Eq. (1).

Example: Applying the values effort = 3855.46 complexity = 3 time = 214.19 KDSI = 27:

$$p = \frac{abs(a-b)+\alpha}{c} = p = \frac{abs(3855.46-214.19)+2}{20}$$
$= 182.1635$

Step 4: Arrive at the percentage of pheromone concentration 'ph' using the formula from Eq. (2) and (3) TSP = 350. α = 1 β = 2.5 γ = 3.

Example: Applying values on 3.2, effort = 3855.46 complexity = 3 time = 214.19 KDSI = 27:

$$Ph = \frac{(3855.46)^{1}\left(\frac{1}{2^{3}}\right)^{2.5}(214.19)^{3}}{350}$$

We find (based on ACO technique), from Table 2 that minimum path value and maximum value for pheromone have optimal values for effort, time and people.

RESULTS AND DISCUSSION

Analogy based cost estimation is the technique used at early stages of estimation. Sometimes analogy fails at early stage of estimation itself for reasons like assumption error, measurement error and skill of estimator. Our study here helps to minimize the assumption and measurement error. Using evolution criteria we define the following three steps, complete analogy based estimation will satisfy these three steps:

Step 1: Find the nearest project effort value with the help of ACO path value.

Step 2: Identify and group projects based on sizing.

Step 3: Improve similarity measure and extract the correct project from our requirements.

Step 1 and 2 helps resolve measurement error and step 3 help minimize the assumption error.

Step 1: Identify and group projects based on sizing: Grouping the project from dataset is not easy because each project contains 72 attributes and all the attributes are related to one another. Two important parameters justify grouping, namely, domain name and size of the project. We collect appropriate values for KLOC and domain from business requirement specification based

Table 2: Eliminated data using ACO algorithm

KLOC	People	Effort	Time	Path
116	11	9200.06	511.13	434.5465
116	8	8477.57	470.97	400.4300
108	13	7139.35	396.63	337.2360
112	12	10534.59	585.25	497.5670
55	17	8170.75	453.98	385.9385
127	15	6476.65	359.79	305.9430

Table 3: Retrieve the path and pheromone values

KLOC	No. of people	Time (b)	Path (p)	Pheromone (ph)
101	15	345.55	293.8200	0.003772478
116	11	511.13	434.5465	18.492339180
116	8	470.97	400.4300	2413.143561000
108	13	396.63	337.2360	0.209542659
95	6	743.47	632.0420	479516.387000000
52	5	504.10	428.5880	573322.751500000
190	19	784.65	667.0570	9.79459E-05
112	12	585.25	497.5670	5.619193540
384	23	1073.12	912.2265	3.34627E-07
55	17	453.98	385.9385	0.000351180
127	15	359.79	305.9430	0.004434093
159	9	1058.13	899.5085	10868.879070000
256	28	855.11	726.9355	2.3291E-11

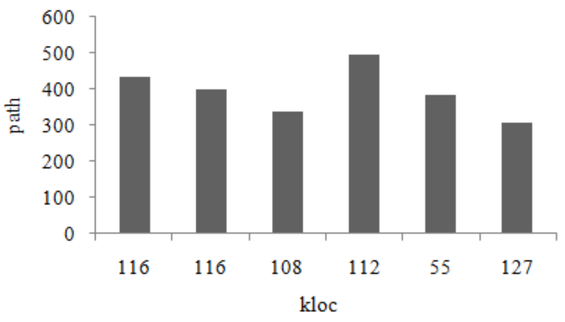

Fig. 2: Relationship between KLOC and path

on expert opinion. Some grouping algorithm fails because grouping data are not directly based on a grouping algorithm. ACO grouping technique helps group the project based on sizing using path and pheromone values. Following steps are involved in ACO grouping:

- Read the input data from dataset which are not null.
- Check each KLOC value is assigned path and pheromone values.
- Referring Table 1, set boundary value for KLOC as, upper boundary value and lower boundary value.

In our case, where KLOC = 27, we set 127 as the upper boundary value (27+100) and 27-100 being the lower boundary value So that the output will be the represented in Table 2.

Table 2 shows that using ACO algorithm and based on requirement we have eliminated unrelated data from Table 1. Collected requirement data from business requirement specification we have to match the data from dataset and extract the correct project with path value. We identified the similar projects addition and subtraction of boundary value 100 with path value. These values displayed in Table 2 and we can find the duplicate projects with various path values easily.

Through Fig. 2 we understand and conclude that KLOC and path are directly proportional.

Step 2: Improving similarity measure and extract the correct project from our requirements: Finding the similarity measure plays on important role in analogy cost estimation. Each project contains more than 72 attributes; we can hence not find the similarity measure easily after finding the similarity projects using 72 attributes. It's very difficult to find maximum and minimum values in the projects.
Steps followed to arrive at similarity measure:

```
1. Read the input data
2. Retrieve the similar projects based on KLOC and domain
3. If (Input KLOC value = = similar project)
   {
   Retrieved project = choose the minimum path value row or max pheromone value row
   }
```

In ACO algorithm follows the concept of when the path value is low it follows the pheromone values is high use this concept and implemented in similarity measure algorithm. If the value of KLOC matches a similar project from Table 3 then we can choose to go with the project that has minimum for pheromone.

Step 3: Finding the nearest project effort value with the help of ACO path value.

Data availability is important parameter for analogy estimation. Some estimators skip the analogy estimation at early stages because of data unavailability. Data available in projects we classified two types, first one requirement data is not available in dataset, we can't handle this situation secondly requirement is available in dataset not exactly related to that but We can handle this situation. Finding the nearest project based on close surrounding of path value related to requirement path. So we identify the related project information.

CONCLUSION

Early stage estimation plays a vital role in industry because the estimation if not accurate. Sometimes overestimation leads to lose the project at biding time or under estimation leads to project failure. In order to avoid this situational we improve the quality of analogy with the help of ACO. Our study here helps to solve common errors like measurement error and data unavailability. Using analogy In future we will minimize the estimator knowledge error and The project is going to be implemented in any of the CMM level 5 companies. Based on the expert's feedback we will minimize the errors.

REFERENCES

Albrecht, A.J., 1979. Measuring application development productivity. Proceeding of the Joint SHARE, GUIDE, and IBM Application Development Symposium. Monterey, California, October 14-17, IBM Corporation 1979, pp: 83-92.

Barry, B., 1981. Software Engineering Economics. Prentice-Hall, Englewood Cliffs, NJ, ISBN: 0-13-822122-7.

Chiu, N.H. and S.J. Huang, 2007. The adjusted analogy-based software effort estimation based on similarity distances. J. Syst. Software, 80(4): 628-640.

Keung, J.W., B.A. Kitchenham and D.R. Jeffery, 2008. Analogy-X: Providing statistical inference to analogy-based software cost estimation. IEEE T. Software Eng., 34: 471-484.

Kocaguneli, E., T. Menzies, A. Bener and J.W. Keung, 2012. Exploiting the essential assumptions of analogy-based effort estimation. IEEE T. Software Eng., 38(2): 425-438.

Kocaguneli, E., T. Menzies, J. Keung, D. Cok and R. Madachy, 2013. Active learning and effort estimation: Finding the essential content of software effort estimation data. IEEE T. Software Eng., 39(8): 1040-1053.

Mittas, N. and L. Angelis, 2010. Visual comparison of software cost estimation model by regression error characteristic analysis. J. Syst. Software, 83(4): 621-637.