ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Obtaining Optimal Software Effort Estimation Data Using Feature Subset Selection

Abirami.R[1], Sujithra.S[2], Sathishkumar.P[3], Geethanjali.N[4]

Student, Department of Computer Science and Engineering, SNS College of Technology, Coimbatore[1, 2, 3]

Assistant Professor, Department of Computer Science and Engineering, SNS College of Technology, Coimbatore[4]

Abstract: Effort estimation is usually an odd job and it should be done with utmost care. It is basically carried out with large set of Software Effort Estimation data. But the problem with estimating such a large data set is very hard and there are many chances of errors. To avoid drawbacks in effort estimation, various techniques have been widely used. But no method produces absolute result for effort estimation. Since visualization is not possible in these cases of estimation, no method can be the best for estimating efforts efficiently. There are some problems like (a)some methods will be suitable for some kind of projects only and (b)some kind of methods will be suitable for smaller projects only (c)some methods can eliminate some necessary data and it leads to wrong estimations. It affects other processes like time and cost estimation, resource allocation, scheduling, etc. So the Software effort estimation data should be reduced using feature subset selection methods to make effort estimation in a better way. Feature selection is very important in various pattern classification problems. The feature selector is applied to select a subset of features from the large set of features. And also the selected subset should be sufficient to perform the estimation process. By using Ant Colony Optimization (ACO) method as a feature selector for obtaining optimized and reduced essential data, optimal solution can be produced for estimating the effort.

Key Terms: Effort, Estimation, Feature Subset Selection, Ant Colony Optimization

I. INTRODUCTION

In Software Engineering, effort estimation is an important process since it is a precursor for many other processes involved in project. So it should be done in a proper and efficient way. Estimated values are usually expressed in units such as man-day, man-month and man-year. The reason for estimation varies according to the project. Some of the reasons can be (a)approving the project: the organization will proceed the project by approving the successful completion of the project through effort estimation, (b)managing the project: the project managers will estimate the effort in order to manage and control the project, (c)understanding the project: the team members should understand the requirements and features of the software, so that they can work well accordingly, (d)defining the project tasks: time, cost, staff allocation and other resource allocation for the tasks can be made efficiently.

But the issue is, the accuracy and correctness of effort estimation methods is unreliable and there is no significant proof for a wide range of estimation techniques. The complexity of effort estimation should be reduced by reducing the enormous effort estimation data into reduced, essential and optimized data. So as to obtain optimal estimated values. From the whole set of features given, only the required subset of data should be abstracted by using feature subset selector. Using the selectors the effort must be estimated in a possible short duration with optimized data obtained through feature subset selection method.

Instead of using complex search methods, meta heuristic methods like Ant Colony Optimization(ACO) can be used to obtain the optimized data subset. It is a meta heuristic inspired by the real ant system. The ants search for the shortest paths to reach their food sources. It looks for the optimal solutions by considering both local heuristics and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

previous knowledge. It is found that a substance called pheromone is used as a communication mode for the ants in an optimized food search. When a source of food is identified by an ant, it will lay some pheromone to mark the path between their place and the food founded. The quantity of the laid pheromone depends on the distance, quality and quantity of the food. When an isolated ant moves randomly detects pheromone, it is very likely that the ant will decide to follow its path. The ant will itself lay a certain amount of pheromone. The main job here is to update the pheromone instead of accumulating pheromones.

Then the path that has been used by more ants will be more likely to be followed. This process can also be defined as a positive feedback loop. But the ants should not focus on only one path so that the pheromone in one path will get increased. The ants should go for new paths and so the ants can find out more optimal path and shorter path. In order to solve optimization problems, a number of artificial ants are used to iteratively construct optimal path. In each iteration, an ant would deposit a certain amount of pheromone proportional to the quality of the food founded. In each step, all the ants will compute a set of feasible solutions to obtain its current partial and incomplete solution based on the local heuristics. Finally they will arrive at an optimal solution to reach their places. The pheromone intensity formula is used to reduce the chance of being struck in local minima. By considering both local importance of features and the overall performance of the subsets, optimal feature space can be identified for sure.

The links represents the connection between the features of the dataset and the nodes are the features of the project. Each and every ant in the ant system constructs a local solution in the search space by traversing the path of nodes and links in the dataset. This path is actually the subset of the feature set. After an ant has completed its search, the correctness and accuracy of the selected features is checked. Then the average accuracy validation gives the optimal feature subset. It is used to update the pheromone values and this is the most important step in ACO effort estimation. These feature selection algorithms must decide when to stop searching through the space of the feature subsets. After the termination of search, the best feature set will be returned as the final optimal solution.

II. SYSTEM ANALYSIS

Existing System:

There are many expert-based, analogy-based and model based methods widely used for effort estimation. In most of the projects model-based methods are used rather then choosing some other methods. These methods range from relatively simple nearest neighbor methods to iterative dichotomization methods. For example, Classification and regression trees (CART) to complex search-based methods like tabu search, genetic algorithms, etc. Though there is an issue of reducing the complexity of the feature datasets and reducing the errors. The effort estimation method should also be less complex because the more complex the estimation methods become, the more prone they become to operator flaws.

Some active learning methods are used to analyze and reduce the data sets into essential data subsets. For instance, QUICK is one of the active learning method that computes the Euclidean Distance between the rows and columns of the software effort estimation datasets and determines the essential subset of data from the larger dataset. The rows are the instances of related old projects and the columns have the features of the project.

It undergoes reduction of datasets by the following steps:

- Grouping the rows and columns by their similarity
- Prune the synonyms in the columns which are too similar
- Prune the outliers in the rows which are too distant
- Now generate the estimate from the remaining data

Drawbacks:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

- It can not produce an accurate solution
- Sometimes essential data can be neglected
- The project may or may not be successful
- The project may end up with failures

Proposed system:

Feature subset selector (FSS) can be used to reduce the given dataset into a reduced and essential subset of data. That is, this feature subset selection process is used to select the smaller subset of features from the large possible feature set and it reduces the occurrence of errors in effort estimation. It should form good subsets and by adding the right features it will become the best subset. Ant Colony Optimization is a feature subset selector which can be used to obtain an optimal dataset for estimation. The logic of real ant system is applied in this method.

The ACO is a meta heuristic inspired by the behavior of real ants in their search for the shortest paths to food sources. It looks for optimal solutions by considering both local heuristics and previous knowledge. The local importance of a given feature is measured using the Mutual Information Evaluation Function (MIEF), which is a filter evaluation function.

General ACO can be implemented as follows:

- A stochastic construction procedure
- Probabilistically build a solution
- Iteratively adding solution components to partial solutions
- Heuristic information
- Trace/Pheromone trail
- Reinforcement Learning reminiscence
- Modify the problem representation at each iteration
- Ants work concurrently and independently
- Collective interaction via indirect communication leads to good solutions

Ant colonies, and more generally social insect societies, are distributed systems that, in spite of the simplicity of their individuals, it presents a highly structured social organization. As a result of this organization, ant colonies can accomplish complex tasks in some cases exceeding the individual capabilities of a single ant. In estimating the effort by reducing the large datasets Ant Colony Optimization can be implemented as follows:

- Ants: Simple computer agents
- Move ant: Pick next feature in the construction of solution

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

- Trace: Pheromone, a global type of information
- Memory: Memory Space to hold the selected features
- Next move: Use probability to move ant

In the first iteration, each ant will randomly choose a feature subset of m features. Only the best k subsets, k < na, will be used to update the pheromone trial and influence the feature subsets of the next iteration. In the second and following iterations, each ant will start with m – p features that are randomly chosen from the previously selected k-best subsets, where p is an integer that ranges between 1 and m – 1. In this way, the features that constitute the best k subsets will have more chance to be present in the subsets of the next iteration. However, it will still be possible for each ant to consider other features as well. For a given ant j, those features are the ones that achieve the best compromise between pheromone trails and local importance with respect to S_j, where S_j is the subset that consists of the features that have already been selected by ant j. The Updated Selection Measure (USM) is used for this purpose and defined as:

$$USM_i^{S_j} = \begin{cases} \dfrac{(\tau_i)^{\eta}\left(LI_i^{S_j}\right)^{\kappa}}{\sum_{g \notin S_j}(\tau_g)^{\eta}\left(LI_g^{S_j}\right)^{\kappa}} & \text{if } i \notin S_j \\ 0 & \text{Otherwise} \end{cases}$$

where LI_i^{Sj} is the local importance of feature f_i given the subset S_j. The parameters η and κ control the effect of pheromone trail intensity and local feature importance respectively. LI_i^{Sj} is measured using the MIEF measure and defined as:

$$LI_i^{S_j} = I(C; f_i) \times \left[\frac{2}{1+\exp(-\alpha D_i^{S_j})} - 1\right]$$

the parameters α, β, and γ are constants, $H(f_i)$ is the entropy of f_i, $I(f_i; f_s)$ is the mutual information between f_i and f_s, $I(C; f_i)$ is the mutual information between the "class labels" and f_i, and $|S_j|$ is the cardinal of S_j.

Each ant will produce partial solution until the stop criteria is reached. For example, if the number of iterations is more than the maximum allowed iteration exit the search, otherwise continue. Then the features are updated throughout the ant system and atlast the best solution is selected.

Merits:

- It facilitates data visualization and provides better data understanding
- Reduces the complexity of training data that leads to reduced training times of the learning algorithm
- Reduce the curse of dimensionality and improve the performance of prediction
- Positive Feedback accounts for rapid discovery of good solutions

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

- Distributed computation avoids premature convergence
- The greedy heuristic helps find acceptable solution in the early solution in the early stages of the search process.
- The collective interaction of a population of agents.

III. CONCLUSION AND FUTURE WORK

ACO is a relatively new meta heuristic approach for solving hard combinatorial optimization problems. Artificial ants implement a randomized construction heuristic which makes probabilistic decisions. The cumulated search experience is taken into account by the adaptation of the pheromone trail. By following this method can find best solutions on small problems. On larger problems converged to good solutions. It is not the best way for estimating the effort for all kind of projects but it can yield best result for most of the projects and atleast a good solution for other projects. On "static" problems like TSP it is hard to beat specialist algorithms. But static problems do not have difficult issues like dynamic problems. Ants are "dynamic" optimizers and the suggestion is that, coupling ACO with local optimizers can give world class results. Also trying combination of other methods with ACO can yield better results. COCOMO datasets can be used to implement and visualize the effect of using this method.

REFERENCES

1. E. Kocaguneli, T. Menzies, and J. Keung, "On the Value of Ensemble Effort Estimation," IEEE Trans. Software Eng., vol. 38, no. 6, pp. 1403-1416, Nov./Dec. 2012
2. T. Menzies, A. Bener, and J.W. Keung, "Exploiting the Essential Assumptions of Analogy-Based Effort Estimation," IEEE Trans. Software Eng., vol. 38, no. 2, pp. 425-438, Mar./Apr. 2012
3. Sanjoy Dasgupta, University of California, "Analysis of a greedy active learning strategy", 2010
4. A.Corazza, S.Di Martino, F.Ferrucci, E. Mendes, "How Effective is Tabu Search to Configure Support Vector Regression for Effort Estimation?", May 2011
5. Huis ter Duin, Noordwijk, The Netherlands, 2CEE, "A Twenty First Century Effort Estimation Methodology", May 2013
6. M. Dorigo and L. M. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1):53–66, 1997
7. L. M. Gambardella, ` E. D. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pages 63–76. McGraw Hill, London, UK, 1999
8. G. Di Caro and M. Dorigo. Mobile agents for adaptive routing. In H. El-Rewini, editor, Proceedings of the 31st International Conference on System Sciences (HICSS-31), pages 74–83. IEEE Computer Society Press, Los Alamitos, CA, 1998
9. M. Jorgensen and T. Gruschke, "The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments," IEEE Trans. Software Eng., vol. 35, pp. 368-383, May/June 2009
10. Lederer, A.L. and J. Prasad, " Software Management and Cost Estimating Error", Journal of Systems and Software, pp. 33-42, 2000
11. T. Menzies, O. Jalali, J. Hihn, D. Baker, and K. Lum, "Stable Rankings for Different Effort Models," Automated Software Eng., vol. 17, pp. 409-437, Dec. 2010
12. T. Menzies, Z. Chen, J. Hihn, and K. Lum, "Selecting Best Practices for Effort Estimation," IEEE Trans. Software Eng., vol. 32, no. 11, pp. 883-895, Nov. 2006