Operation Management for validating the software quality using enhanced Hungarian method

Dr. V.Jayaraj [1], N.A. Bhaskaran [2]

[1] Associate Professor, Dept. of CSE, Bharathidasan University, Tiruchirapalli, India

[2] Research Scholar, Bharathiyar College of Engineering and Technology, Karaikal India

Abstract

The main endeavor of software development is to produce optimal efficiently and effectively. Software estimation model means of a mathematical model to estimate software size (in terms of size of coding), effort, or scheduled from various entities such as the number of screens and reports which estimating the effort required to develop for vital importance. Software estimation is the process of predicting the effort and cost required to develop effective software. The problem instance has a number of agents and a number of tasks the agent can be assigned to perform any task, incurring some cost that may vary depends on task assignment is required to perform all tasks by assigning exactly one agent to each task and exactly one task to each agent in such a way that the total cost of the assignment is minimized. By this we distinguish the essential content of Software Effort Estimation (SEE) data, the least number of features and instances required to capture the information within SEE data if the crucial entity is very small.

Keywords: Effort Estimation (EE), Cost Estimation(CE), Machine Learning (ML), Assignment Problem, Hungarian Method.

1. Introduction

An important goal of software engineering community is accurate software cost Estimation. Software Cost Estimation with resounding reliability, productivity and development effort is challenging and tough task. In software engineering effort is used to denote measure of use of workforce and is defined as total time that takes members of a development team to perform a given task. It is usually expressed in units such as man-day, man-month, and man-year. Any improvement in the accuracy of prediction of development effort can significantly reduce the cost from the inaccurate estimation. Accurate modeling can also assist in scheduling resources and evaluating risk factors. The most critical activity in software project management during the initiation phase of a project to estimate the effort related cost essential to complete the project tasks.

The cost estimate of software development is in focusing of much research over in recent past years. Many methods have been applied to clarify software development cost as a function of a large number of its relevant cost factors. The static modeling techniques chosen from statistics, such as machine learning, and knowledge acquisition have been used for various rates of degrees of success for small as well as big data set. Effort estimation is a odd job and should be done with utmost care. It is essentially carried out a complex action that requires knowledge of a number of key attributes. Estimation analogy appears is suited to effort estimation, when the software product is scantily implicit it is concerned with finding solutions for a known problem and applies its solution from set of similar projects. The key activities for estimating by analogy is identification of a problem as a new case, the retrieval of similar cases from a repository to reuse of knowledge consequential from previous cases and the suggestion for the new case.

Estimation means “prediction”. It neither is used in plan nor in commitment to calculate approximate results estimation techniques are used; to make uncertain or incomplete data become usable. Software effort estimation is used to estimate of effort required to develop software.

Feature selection is also called variable selection or attributes selection. It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem we are working. The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.

The assignment problem is one of the fundamental combinatorial optimization problems of operations research in mathematics. It consists of finding a maximum weight matching (or minimum weight perfect matching) in a weighted bipartite graph. The **Hungarian method** is a combinatorial optimization algorithm which helps to solve the assignment problem in polynomial time and which anticipated later primal-dual methods.[1][2]

2. Related Work

There is a wealth of active learning studies in machine learning literature. For example, Das Gupta [3] seeks generalizability guarantees in active learning. He used a greedy active learning heuristic and showed that it is able to deliver performance values as good as any other heuristic in terms of reducing the number of required labels [3].Supervised algorithms require instance labels, while unsupervised ones execute on unlabeled data. Instance selection algorithms generate prototypes, i.e., a subset of the data which best represents the predictive properties of the whole data. A standard result in instance selection is that most of the rows in a matrix of data can be removed without damaging the predictive power of rules learned from the remaining data Ekrem[4]. Shepperd et al. [5], expresses EbA in an automated environment known as ANaloGy softwarE tool (ANGEL) that supports the collection, storage and identification of the most analogous projects from the repository in order to estimate the cost and effort. It uses Euclidean distance as the distance measure to reduce the amount of computation involved.

2.1. Theorem:

If a number is added to or subtracted from all of the entries of any one row or column of a cost matrix, then on optimal assignment for the resulting cost matrix are also an optimal assignment for the original cost matrix.

Suppose there are **n** facilitates and **n** jobs it is clear that in this case, there will be **n** assignments. Each facility or say worker can perform each job, one at a time. But there should be certain procedure by which assignment should be made so that the profit is maximized and cost or time is minimized.

Projects	**Project Data Set**					
	1	**2**	**3**	**4**	**j^{th}**	**n**
P_1	Pd_{11}	Pd_{12}	Pd_{13}	Pd_{14}	……	Pd_m
P_2	Pd_{21}	Pd_{22}	Pd_{23}	Pd_{24}	……	Pd_{2n}
P_3	Pd_{31}	Pd_{32}	Pd_{33}	Pd_{34}	……	Pd_{3n}
i^{th}				Pd_{ij}	……	
n^{th}	Pd_{n1}	Pd_{n2}	Pd_{n3}	Pd_{n4}	……	Pd_{nm}

In the table, P_1, P_2, P_3 are the projects and Pd_{ij} is defined project variables of (n x n) matrix. It is a special case of balanced transportation problem when the number of rows is equal to number of columns (R=C).

In the table, Pd_{ij} is defined as the cost when j^{th} job is assigned to i^{th} worker. It maybe noted here that this is a special case of transportation problem when the number of rows is equal to number of columns.

2.2. Mathematical Formulation

The basic feasible solution of an Assignment problem may consists (2n – 1) variables out of its (n – 1) variables are zero; n is number of jobs or number of facilities. Due to this high degeneracy, the solution of this problem is complex and time consuming transportation method. Thus a separate technique is derived for it. Before going to the absolute method it is very important to formulate the problem.

Suppose x_{ij} is a variable which is defined as

1 if the i^{th} job is assigned to j^{th} machine or facility

0 if the i^{th} job is not assigned to j^{th} machine or facility.

Now as the problem forms one to one basis or one job is to be assigned to one facility or machine.

Therefore

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ and } \sum_{j=1}^{n} x_{ij} = 1$$

The total assignment cost will be given by

$$U = \sum_{j=1}^{n}\sum_{i=1}^{n} x_{ij} c_{ij}$$

The above definition can be developed into mathematical model as follows:
Determine $x_{ij} > 0$ (i, j = 1,2, 3…n) in order to
Minimize

$$U = \sum_{j=1}^{n}\sum_{i=1}^{n} x_{ij} c_{ij}$$

Subjected to constraints

$\sum_{i=1}^{n} x_{ij} = 1$ When j = 1,2,3 . . . . n.

$\sum_{j=1}^{n} x_{ij} = 1$ When i = 1,2,3 . . . . n.

And x_{ij} is either zero or one.

2.3. Hungarian Technique :

Consider the objective function for minimization type. The algorithm flow for solution of Assignment problem,

1. Allocate the smallest cost element in each row of the given cost table starting with the first row then subtracted from each element of that row least one zero will assign in each row this new table.
2. Allocate the smallest cost element in each column of step-I starting with the first column then subtracted from each element of that column least one zero will assign in each column of the new table
3. Now, the assignments are made for the reduced table in following manner.

i. Rows are examined successively, until the row with exactly single (one) zero is found. Assignment is made to this single zero by putting square □ around it and in the corresponding column, all other zeros are crossed out (x) because these will not be used to make any other assignment in this column. Step is conducted for each row.

ii. Step 3 (i) in now performed on the columns as follow:- columns are examined successively till a column with exactly one zero is found. Now , assignment is made to this single zero by putting the square around it and at the same time, all other zeros in the corresponding rows are crossed out (x) step is conducted for each column.

iii. Step 3, (i) and 3 (ii) are repeated till all the zeros are either marked or crossed out. Now, if the number of marked zeros or the assignments made are equal to number of rows or columns, optimum solution has been achieved. There will be exactly single assignment in each or columns without any assignment. In this case, we will go to step

4. Draw the minimum number of lines (horizontal and vertical) necessary to cover all zeros in the matrix obtained in step 3, Following procedure is adopted:

i. Tick mark (✓) all rows that do not have any assignment.

ii. Now tick mark (✓) all these columns that have zero in the tick marked rows.

iii. Now tick marks all the rows that are not already marked and that have assignment in the marked columns.

iv. All the steps i.e. (4(i), 4(ii), 4(iii) are repeated until no more rows or columns can be marked.

v. Now draw straight lines which pass through all the un marked rows and marked columns. It can also be noticed that in an n x n matrix, always less than 'n' lines will cover all the zeros if there is no solution among them.

5. In step 4, if the number of lines drawn are equal to n or the number of rows, then it is the optimum solution if not, then go to step 6.
6. Select the smallest element among all the uncovered elements. Now, this element is subtracted from all the uncovered elements and added to the element which lies at the intersection of two lines. This is the matrix for fresh assignments.

7. Repeat the procedure from step (3) until the number of assignments becomes equal to the number of rows or number of columns.

The optimal assignment can be found using the Hungarian algorithm. The Hungarian algorithm has worst case run-time complexity of O(n^3).

This section offers a small example to find out minimal cost of each project by choosing one feature from one row (Project)

The original cost matrix:

8	10	17	9
3	8	5	6
10	12	11	9
6	13	9	7

Subtract row minima
We subtract the row minimum from each row:

0	2	9	1	(-8)
0	5	2	3	(-3)
1	3	2	0	(-9)
0	7	3	1	(-6)

Subtract column minima
We subtract the column minimum from each column:

0	0	7	1
0	3	0	3
1	1	0	0
0	5	1	1
	(-2)	(-2)	

Cover all zeros with a minimum number of lines

There are 4 lines required to cover all zeros:

0	0	7	1	**x**
0	3	0	3	**x**
1	1	0	0	**x**
0	5	1	1	**x**

The optimal assignment
Because there are 4 lines required, the zeros cover an optimal assignment:

0	0	7	1
0	3	0	3
1	1	0	0
0	5	1	1

This corresponds to the following optimal assignment in the original cost matrix:

8	10	17	9
3	8	5	6
10	12	11	9
6	13	9	7

So from the above example Project 1 Column 2 feature is chosen, from Project 2 column 3 feature is chosen, from Project 3 column 4 feature is chosen and from Project 4 column 1 feature is chosen and hence the result

Project	Selected features
P_1	10
P_2	5
P_3	9
P_4	6

3. Solution by Enhanced Hungarian Technique

Consider the objective function for minimization type. The algorithm flow for solution of Assignment problem,

1. Check smallest and next smallest element in each row and column then subtracted from each element of that row, column have least one zero will assign in each row, column of new table so least one zero will assign in each row, column.
2. Check rows, column with exactly single (one) zero this is allocation by square □ around it and cross the corresponding column and row and allocate if all allocation is assigned then stop the process else repeat step I crossed rows and column be neutral no operation is required.
3. Draw the minimum number of lines (horizontal and vertical) necessary to cover all zeros in the matrix obtained in step 2,
4. In step 4, if the number of lines drawn are equal to the number of rows, then it is the optimum solution if not, then repeat step-I.
5. Repeat the procedure from step (3) until the number of assignments becomes equal to the number of rows or number of columns.

The optimal assignment can be found has worst case run-time complexity is less than of Hungarian Algorithm run time complexity O(n^3).

This is the original cost matrix:

8	10	17	9
3	8	5	6
10	12	11	9
6	13	9	7

Step –I

0	0	7	1
0	3	0	3
1	1	0	0
0	5	1	1

Step-II

Cover all zeros with a minimum number of lines

The optimal assignment

0	0	7	1
0	3	0	3
1	1	0	0
0	5	1	1

Project	Selected features
P_1	10
P_2	5
P_3	9
P_4	6

4. Results and Discussion

In the existing method the system chooses row minima and subtract from all the elements until all the rows are completed. Similarly the system chooses the column minima and subtract from all the elements of the column until all the columns are completed. But in the proposed algorithm the system simultaneously chooses the row minima and column minima and subtract from the others and hence the result. So this has reduced the time complexity of the overall computational process.

Supervised algorithms require instance labels, while unsupervised ones execute on unlabeled data. Instance selection algorithms generate prototypes, i.e., a subset of the data which best represents the predictive properties of the whole data [4]. It is well known that selecting a subset of the SEE dataset features can improve the estimation performance. Feature selection methods aids in mission to create an accurate predictive model. They help us by choosing features that will give us as good or better accuracy whilst requiring less data.

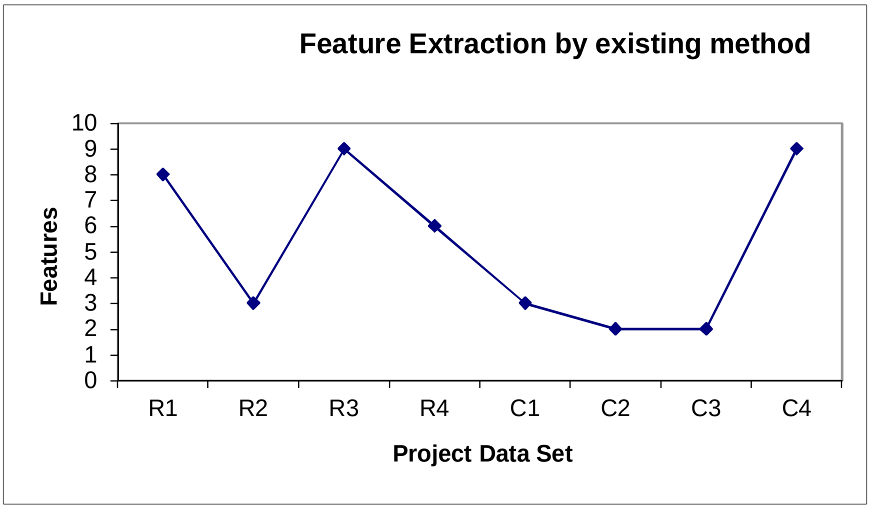

Fig – 1 feature extraction by existing method

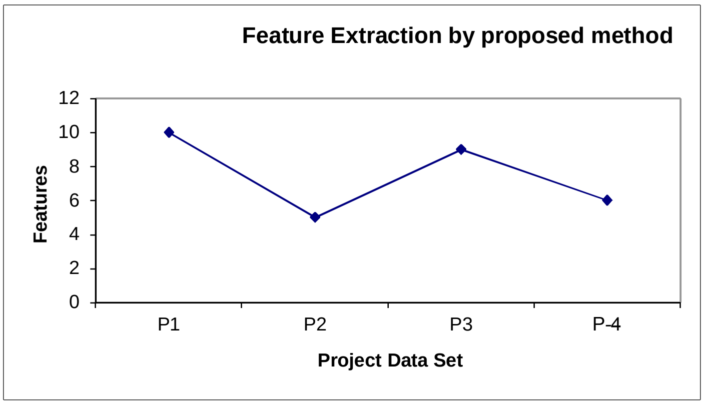

Fig – 2 feature extraction by proposed method

While depicting the graph using the existing method, with a given sample set of data, the cost of the reduced matrix is calculated row wise first and then the column wise and shown in figure 1. Similarly depicting the graph by using the proposed method using the same set of data, we are reducing the cost of the adjacent matrix by row and column wise simultaneously. And the total effort is successfully shown in figure 2.

5. Conclusion

We consider the case of square metric of order 4 as consider the balanced (row=column) problem in Hungarian method more mathematical operation is required and also the run time complexity is O(n^3). But by applying the new algorithm we found less mathematical calculation and the run time complexity is less than O(n^3). The ultimate objective of this research paper is to study the essential content of SEE datasets and make recommendations regarding which estimation methods (simple or complex) should be favored.

Reference

[1]. Harold W. Kuhn, "The Hungarian Method for the assignment problem", Naval Research Logistics Quarterly, 2: 83–97, 1955. Kuhn's original publication.

[2]. Harold W. Kuhn, "Variants of the Hungarian method for assignment problems", Naval Research Logistics Quarterly, 3: 253–258, 1956.

[3]. S. Dasgupta, "Analysis of a Greedy Active Learning Strategy," Proc. Neural Information Processing Systems, vol. 17, 2005.

[4]. Ekrem Kocaguneli, Tim Menzies, Jacky Keung, David Cok, and Ray Madachy, "Active Learning and Effort Estimation: Finding the Essential Content of Software Effort Estimation Data, IEEE Transactions on Software Engineering, vol. 39, no. 8, pp. 1040-1053, August 2013.

[5]. Shepperd M., Schofield C., and Kitchenham B., "Effort Estimation Using Analogy," in Proceedings of the 18th International Conference on Software Engineering, Berlin, Germany, pp. 170 - 178, 1996.