Software Cost Estimation using Particle Swarm Optimization and Neural Network

Sheena Goyal¹, Sheilly Padda²

¹Student (Pursuing M.Tech in CSE), ²Assistant Professor, Department of computer science and engineering, Chandigarh Group of colleges, Landran, Mohali, Punjab, India

Abstract – In today’s time, cost estimation of the project is the major part. Whenever a project is required by any company or organization, then the second major thing along with project requirements comes the cost they can spent on the project. In project management, planning and resources are allocated according to the cost it takes. Only then resources are described and assigned which includes the effort in allocating the resources and time in allocation of resources. In our research we are going to minimize the effort estimation non-linearity using clustering technique called K-means clustering. Then parameters of each cluster are tuned using PSO. These clusters and tuned parameters are further trained using back propagation.

Keywords – PSO, Software cost estimation, K-means clustering.

I. INTRODUCTION

For the software industry, software cost estimation has been a challenge from a long time. Accurate cost estimation helps software companies making correct cost estimation and avoiding loss of money due to the wrong cost estimation. Software cost estimation can be defined as the estimated judgment of the expenses on the resources and the effort required allocating the resources for an undertaking. Cost estimation is typically measured regarding effort. The most widely recognized metric used is individual months or years (or man months or years). The effort is the measure of time for one individual to work for a certain time on the project. It is essential to record the attributes of a particular in the advancement environment into light when contrasting the effort of two or more undertakings in light of the fact that no two situations that are improved are same.

II. LITERATURE REVIEW

G. Sivanageswara Rao, Ch.V. Pani Krishna and K. Rajasekhar Rao et al (2013) [1] worked on COCOMO and MARE and used Multi Objective PSO to tune the parameters. The parameters of model tuned by using MOPSO considered two objectives, Mean Absolute Relative Error and Prediction. They observed that the model they have proposed had given better results when compared with the standard COCOMO model.

Ali Idri et al [10] stated that web Effort Estimation is a procedure of anticipating the endeavors and cost regarding cash, timetable and staff for any product venture framework. Numerous estimation models have been proposed in the course of the most recent three decades and it is accepted that it is an absolute necessity with the end goal of: Budgeting, danger examination, venture arranging and control, and task change speculation investigation.

Haigang Li et al (2015) [12] demonstrated that PSO calculation is a smart streamlining calculation in view of swarm knowledge. Molecule swarm enhancement calculation is straightforward, simple to execute, and it has a wide application prospect in investigative research and building applications. In actuality, most of the enhancement issue is the advancement issue of some nonlinear discrete with the presence of neighborhood. PSO calculation additionally has a few deformities in treating advancement issue. The ideal execution of the PSO calculation is proficiency; the quality weights are streamlined, which is the same as to enhance the precision of case recovery. The utilization of case is based thinking in the streamlining of weight vessel model outline. Through the examination comes about, the improvement of the execution of PSO calculation is better; the consequence of forecast is more rough to the real esteem, which can address the issues of reasonable applications in designing.

III. SOFTWARE COST ESTIMATION

All people in software industry know that there are lots of uncertainties involved in software cost estimation. So, it is difficult to expect perfect effort estimates even in perfect estimation process. The software development effort of a software project is more frequently estimated by project managers using the estimated effort they calculate the cost and duration associated with the project. Accurate development effort estimation at the earlier stage of a software development cycle is the most important to plan, monitor and control the allocated resources appropriately.

The uncertainties and factors on which software development effort depends on the following factors:

- The amount of Implemented functionality [11]
• Number of Errors made by programmers [11]
• Quality of code produced [11]
• Availability of development tools [11]
• Probabilistic factors (absence of staff members due to sickness) [11]
• Availability of skilled persons [11].

IV. COCOMO MODEL

COCOMO stands for Constructive Cost Model; it is a software cost estimation model that was first published in 1981 by Barry Boehm. It is an algorithmic approach to estimating the cost of a software project. By using COCOMO you can calculate the amount of effort and the time schedule for projects. From these calculations you can then find out how much staffing is required to complete a project on time. COCOMO’s main metric used for calculating these values is lines of code (denoted KLOC for COCOMO II, or KDSI for COCOMO 81 and measured in thousands), function points (FP), or object points (OP). COCOMO 81 was the first form of COCOMO. It has been discovered that generally speaking it has the capacity produce gauges that are inside 20% of the real values 68% of the time.

COCOMO model has three different modes
1. Organic
2. Embedded
3. Semi detached

V. PSO (PARTICLE SWARM OPTIMIZATION)

PSO is a stochastic global optimization method which is based on simulation of social behavior. As in GA and ES, PSO exploits a population of potential solutions to probe the search space. In contrast to the aforementioned methods in PSO no operators inspired by natural evolution are applied to extract a new generation of candidate solutions. Instead of mutation PSO relies on the exchange of information between individuals, called particles, of the population, called swarm. In effect, each particle adjusts its trajectory towards its own previous best position, and towards the best previous position attained by any member of its neighborhood. In the global variant of PSO, the whole swarm is considered as the neighborhood. Thus, global sharing of information takes place and particles profit from the discoveries and previous experience of all other companions during the search for promising regions of the landscape. To visualize the operation of the method consider the case of the single objective minimization case; promising regions in this case possess lower function values compared to others, visited previously [3].

1. ALGORITHM OF PSO

Numbers of birds are searching for food randomly in an area. There is only one piece of food is available. All birds do not know where the food is, but they know how far it is in iteration. So the best solution to find out the food is to follow the bird nearest to the food.

PSO is used for optimization problem. Each single solution in the search space is a bird. We call it particle.

All the particles have
1. Fitness values, evaluated by fitness function
2. Velocities which direct the flying of the object

Particles fly through the problem space by following current optimization particles.

PSO is initialized with a group of random particles (solutions) and searches for optima by updating generations.

In each iteration, each particle is updated by following two “best” values. The first solution is the best solution it has achieved so far. The value is called “pbest”. Another best value that is tracked by PSO is the best value obtained by any particle in the population. This best value is a global best and called “gbest”. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called “lbest”.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

\[
v[i] = v[i] + c1 * \text{rand}( ) * (pbest[i] - \text{present}[i]) + c2 * \text{rand}( ) * (gbest[i] - \text{present}[i]) \\
\text{present}[i] = \text{present}[i] + v[i]
\]

\[v[i] \text{ is the particle velocity, present}[i] \text{ is the current particle (solution), pbest}[i] \text{ and gbest}[i] \text{ are defined as stated before, } \text{rand}( ) \text{ is a random number between } (0, 1). c1, c2 \text{ are learning factors. Usually } c1 = c2 = 2.\]

VI. BACK PROPAGATION ALGORITHM

The back propagation learning algorithm is one of the most widely used methods in neural network. The network associated with back-propagation learning algorithm is called as back propagation network. While training a network a set of input-output pair is provided the algorithm provides a procedure for changing the weight in BPN that
helps to classify the input output pair correctly. Gradient descent method of weight updating is used.

Figure: Architecture of a Back propagation Network

The aim of the neural network is to train the network to achieve a balance between the ability of net to respond and its ability to give reasonable responses to the input that is similar but not identical to the one that is used in training. Back propagation algorithm differs from the other algorithm by the method of weight calculation during learning. The drawback of Back propagation algorithm is that if the hidden layer increases the network become too complex [11].

VII. CONCLUSION AND FUTURE SCOPE

In our research we are going to minimize the effort estimation non-linearity using clustering technique called K-means clustering. Then parameters of each cluster are tuned using PSO. These clusters and tuned parameters are further trained using back propagation. We are using clustering techniques as well as back propagation neural network algorithm as drivers for the improvement. For further work, any genetic algorithm can be used for improving the linearity of the results.

VIII. REFERENCES


