Feature Subset Optimization through the Fireworks Algorithm

Tad Gonsalves
Department of Information and Communication Sciences, Faculty of Science & Technology, Sophia University, Tokyo, Japan
t-gonsal@sophia.ac.jp

Abstract - Software estimation models are vital in the software industry, given the fact that most software development projects over-run the time and budget limits. Recently, data mining algorithms have been applied to further improve the prediction accuracy of the software cost estimation models that are routinely used in industry. This paper introduces a novel Swarm Intelligence technique to fine-tune such models. It chooses the Feature Selection Method (FSS) to reduce the number of input parameters in the dataset. Further, it applies the Fireworks Algorithm (FWA) to deal with the combinatorial explosion problem in determining the optimal subset of features. The FWA-FSS method improves the prediction accuracy when tested on publicly available software development project datasets.

Keywords – Feature Subset Selection, Fireworks Algorithm, Software Cost Estimation, Optimization

I. INTRODUCTION

Software estimation models are vital in the software industry, given the fact that most software development projects over-run the time and budget limits. Underestimating software costs can have detrimental effects on the quality of the software, while overestimation of software cost can result in missed opportunities to use funds in other projects [1]. Project managers and software development organizations need accurate estimates during the early stages in the lifecycle of the project. Expert judgment and algorithmic models are the two kinds of approaches common in the estimation of software development costs. In the expert judgment models expert knowledge is used to estimate the cost of software development. In the parametric models, the relationship between the project development effort and the project characteristics is expressed in the form of equations. However, neither the parametric nor the non-parametric cost estimation models are found to be perfect. The Standish group study revealed that 53% of the US software projects ran over 189% of the original estimate [2]. The off the shelf approach of plugging in the cost drivers is also reported to be unsuccessful [3], [4].

The COCOMO 81 software cost estimation model was proposed in 1981[5] and further developed into COCOMO II in 1992 [6]. One of the prominent reasons for selecting COCOMO 81 for this study is the availability of data online for experimentation. The availability of datasets in the public domain is more indicative for COCOMO’s use than its representativeness or other properties [7]. Owing to its success in predicting software cost estimates coupled with the public availability of data, the COCOMO model has been extensively used in research studies [8-12].

Many machine learning and data mining techniques have been applied to improve the prediction accuracy of the existing parametric models [13-14]. However, there are two major limitations to the current data mining practice. It relies heavily on expert heuristics in determining the attributes that influence the predicting power of the model. Secondly, it relies on the the hill-climbing optimization strategy which invariably gets trapped in local optima. Feature Subset Selection is a kind of data-mining pre-processing, in which only a subset containing suitable number of the original features is selected for learning the predicting model [15-18]. Many meta-heuristic algorithms like the Genetic Algorithm [19], the Ant Colony Algorithm [20] and the Particle Swarm Optimization [21] have been used in Feature Subset Selection.
This study introduces the Fireworks Algorithm (FWA) and shows how to use it to perform FSS to fine-tune the prediction accuracy of the COCOMO model. FWA is a recently developed Swarm Intelligence Optimization technique which mimics the explosion of the fireworks in the night sky to achieve function optimization [22-24]. Although the algorithm is fairly new, it has already been subjected to extended developments and variations [25-29] to improve its performance.

Although the applications of this algorithm are not yet widespread, some of the prominent applications mentioned in the literature are as follows: Power loss minimization in electrical networks [30], mass minimization of trusses [31], spam detection [32], orientation coding in identification [33], selective harmonic elimination in PWM inverter [34], digital filter design [35], non-negative matrix factorization [36-38], etc. Zeng et al. have extended the algorithm to multi-objective optimization. They present a multi-objective fireworks optimization algorithm (MOFOA) for oil crop fertilization. It takes into consideration not only crop yield and quality but also energy consumption and environmental effects [39].

The rest of the paper is organized as follows. The Fireworks Algorithm is introduced in Section II. Software development cost estimation parametric models are described in Section III. Application of the FWA algorithm for Feature Subset Selection in optimizing the parametric models is explained in Section IV and the experimental results are presented in section V. Concluding remarks are given in section VI.

II. FIREWORKS ALGORITHM

The Fireworks Algorithm (FWA) belongs to the class of Swarm Intelligence optimization algorithms. The algorithm deriving its inspiration from the fireworks exploding in the night sky, was first proposed by Tan Ying and Yuanchun Zhu in 2010 [15]. The authors state that the performance of SFWA is superior to that of the Standard PSO and the Clonal PSO in all the nine benchmark test functions they experimented with [16]. The global convergence and time complexity of the algorithm are tested by Liu et al. [17].

The original FWA has gone through some significant changes over the years. For instance, Liu et al. have designed a transfer function to calculate the explosion amplitude and the number of sparks generated by a given firework. Further, the control parameter of the transfer function is effectively used to balance the local and global search in every iteration [21]. Extended developments and variations of the algorithm are found in [22-24]. The algorithm is hybridized with biogeography-based algorithm [25]. Zheng et al. have combined it with Differential Evolution algorithm. They report that the hybrid algorithm’s performance is better than the two individual algorithms from which it is made [26].

The FWA algorithm proceeds as follows. It begins with random initial positions of \( N \) fireworks. Before the fireworks explode generating sparks, the amplitude and the number of the explosion sparks are calculated. Fireworks with higher fitness values will have a smaller explosion amplitude and a larger number of explosion sparks, while fireworks with lower fitness values will have a larger explosion amplitude and a smaller number of explosion sparks. The amplitude \( A_i \) and the number of sparks \( S_i \) for the \( i \)th firework \( X_i \) are respectively given by:

$$A_i = A \frac{f(x_i) - f_{min} + \epsilon}{\Sigma(f(x_i) - f_{min}) + \epsilon} \quad (1)$$

$$S_i = M \frac{f_{max} - f(x_i) + \epsilon}{\Sigma(f_{max} - f(x_i)) + \epsilon} \quad (2)$$

where, \( A \) and \( M \) are constants. \( A \) is for controlling the explosion amplitude and \( M \) for the number of explosion sparks. \( f_{min} \) and \( f_{max} \) are respectively, the minimum and the maximum values of the objective function. The parameter \( \epsilon \) is introduced to prevent the denominator and the numerator from falling to zero. Further, the explosion amplitude parameter \( M \) is bounded by:

$$S_i = \text{round}(aM), \quad \text{if } S_i < aM$$
$$\quad = \text{round}(bM), \quad \text{if } S_i > bM$$
$$\quad = \text{round}(S_i), \quad \text{otherwise} \quad (3)$$
where, \( a \) and \( b \) are some pre-determined constants.

The "regular" sparks are generated according to the above equations. After the explosion, different kind of sparks, called "Gaussian" sparks are generated based on a Gaussian mutation process. A new population of \( N \) fireworks is selected at the end of each iteration. This may include the original fireworks, as well as the regular and Gaussian sparks. The elitist strategy is maintained by always inserting the current best location in the new population. The algorithm goes through a pre-determined number of iterations. The pseudo-code of FWA algorithm is shown in Figure 1.

![Figure 1. Fireworks algorithm pseudo-code](image)

### III. PARAMETRIC COST MODELS

The Cost Constructive Model (COCOMO) was introduced by Barry Boehm in 1981 as a model for estimating effort, cost, and schedule for software projects [5]. The model was defined based on the analysis of 63 completed projects from diverse domains. COCOMO 81 consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO, is good for quick, early, rough order of magnitude estimates of software costs. Intermediate COCOMO takes 15 different cost drivers or effort multipliers into account. Detailed COCOMO additionally accounts for the influence of individual project phases. The COCOMO 81 model has been extensively used in industry as well as academia.

In 1997, COCOMO 81 was extended to COCOMO II to include the newer paradigms in software development and finally published in a book in 2000 [6]. COCOMO II is better suited for estimating modern software development projects. Among the main upgrades are the introduction of new functional forms that use scale factors and new cost drivers and a set of parameter values. However, unlike COCOMO 81, COCOMO II datasets are not publicly available for verification and further research.

In both the COCOMO models, the effort required to develop a project can be written in the following general form:

\[
PM = a \times (KLOC)^b \times (\Pi_j EM_j) \tag{4}
\]

where,

PM = effort estimate in person months
Feature Subset Optimization through the Fireworks Algorithm

a = productivity coefficient
b = economies (or diseconomies) of scale coefficient.
KLOC = kilo lines of code
EM = effort multipliers or cost drivers

IV. FEATURE SUBSET SELECTION USING FWA

This section describes the objective function that is to be optimized using the Fireworks Algorithm. It introduces the publicly available datasets used to conduct the study and explains the use of the FWA algorithm to perform Feature Selection.

A. Objective functions

The mean magnitude of the relative error (MMRE) and prediction (PRED) are the most widely used metrics for evaluation of the accuracy of cost estimation models [12]. These metrics are calculated based on a number of actuals observed and estimates generated by the model. They are derived from the basic magnitude of the relative error (MRE, which is defined as:

\[
MRE_i = \frac{|Actual\ Effort_i - Predicted\ Effort_i|}{Actual\ Effort_i} \tag{5}
\]

\[
MMRE(\%) = \frac{1}{T} \sum_i^{T} MRE_i \times 100 \tag{6}
\]

where, T is the total number of instances.

Yet another metric common among project managers used for model evaluation is PRED. PRED(l) is defined as the percentage of estimates where MRE is not greater than l, that is PRED(l) = k/n, where k is the number of estimates with MRE falling within l, and n is the total number of estimates. For example, PRED(30) = 75% would mean that three-quarters of the estimates are within 30% of the actual. For single optimization problems, PRED value is considered as a standard in reporting COCOMO calibration and model.

\[
PRED_N(\%) = \frac{100}{T} \sum_i^T \begin{cases} 
1 & \text{if } MRE_i \leq N/100 \\
0 & \text{otherwise}
\end{cases} \tag{7}
\]

B. Datasets

This study uses the COCOMO I datasets for experimentation because they are open models with published data. The datasets are publicly available online in the PROMISE repository [40]. The datasets are as follows.

• Projects 02, 03, 04 (p02, p03, p04)
  These three are subsets of NASA. There are 22 projects in p02, 12 in p03 and 14 in p04.
• NASA
  There are 60 projects coming from aerospace applications.
• COCOMO
  There are 63 projects in this data set. They are collected from a variety of domains including science, engineering and finance.
C. FSS through FWA

The FW algorithm performs FSS as shown in the flowchart (Figure 2). The dataset is first divided into a training set (70% of the instances) and a test set (30% of the instances). N initial bit strings of 0s and 1s of length 15 (corresponding to the COCOMO effort multipliers) are randomly generated. A 0 signifies that the particular feature is not selected, while a 1 signifies that the corresponding feature is selected. The 15-bit string is called a “firework”. The MMRE (or PRES(30)) for each of the fireworks and accordingly their amplitude and number of sparks are calculated using equations (1), (2), and (3). The regular and the Gaussian sparks are then generated and all the solutions are then evaluated. This algorithm is iterated over 100 iterations. The optimized model is then tested using the test data.

V. EXPERIMENTAL RESULTS

The five datasets (p02, p03, p04, nasa, and cocomo) from the promise repository were used to conduct test studies with the FW algorithm. Each of the datasets were divided into a training set and a corresponding test set. The training set contained 70% of the instances, while the test set contained the remaining 30%. This is a common procedure followed in statistical studies. For each training set the FW algorithm was iterated 100 times to optimize the number of the features selected. For each of the datasets, feature subset selection containing less than 11 features did not show any improvement in the prediction accuracy. However, the accuracy began to rise consistently for subsets containing 11 and 12 features. The peak for all the datasets was for subsets of 12 features. It then began to decline steadily. The experimental results are shown in Table 1 and in Figure 3.
Feature Subset Optimization through the Fireworks Algorithm

Table 1 COCOMO 81: PRED(30) averages

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td>p02</td>
<td>76.48</td>
</tr>
<tr>
<td>p03</td>
<td>41.33</td>
</tr>
<tr>
<td>p04</td>
<td>67.98</td>
</tr>
<tr>
<td>nasa</td>
<td>71.45</td>
</tr>
<tr>
<td>cocomo</td>
<td>56.43</td>
</tr>
<tr>
<td>Mean</td>
<td>62.734</td>
</tr>
</tbody>
</table>

Figure 3. COCOMO 81: PRED(30) v/s the number of selected features

VI. CONCLUSION

Software estimation models are vital in the software industry, given the fact that most software development projects over-run the limits of time budget. The Constructive Cost Model, which has been in use in industry for a relatively long time is a subject of many studies. Recently, data mining algorithms, in particular, have been applied to further improve its prediction accuracy. This paper introduces a novel approach in fine-tuning the Constructive Cost Model. It chooses the Feature Selection Method to reduce the number of input parameters in the dataset. Further, it applies...
the FWA algorithm to deal with the combinatorial explosion problem in determining the optimal subset of features. The FW algorithm conducts parallel search without getting trapped in local minima like the hill-climbing algorithms employed in data mining. The experimental results imply that accurate cost estimates for a new project can be made with fewer cost drivers. In other words, the developers need not put in all the attributes (features) prescribed by the classical parametric model. With fewer data points the complexity of the model decreases and the cost involved in collecting data is indirectly reduced.

REFERENCE

Feature Subset Optimization through the Fireworks Algorithm