Classification of Software Projects using k-Means, Discriminant Analysis and Artificial Neural Network

R. Chandrasekaran and R. Venkatesh Kumar

Abstract—An attempt is made in this paper to identify the groups of software development projects which demonstrate the significance of comparable characteristics based on various parameters associated with the Source Lines of Code (SLOC). Initially, software projects are clustered by using k-means cluster analysis. These groups are investigated further with the Discriminant Analysis (DA) and compared with Artificial Neural Network (ANN). For this purpose, we collected the historical data of software development projects from one of the major information technology (IT) company and this provided essential progressive information such as Planned Value (BCWS), Actual Cost (ACWP), Earned Value (BCWP), Cost Performance Index (CPI), Average Team Size (Team_Size) and Source Lines of Code (LOC). The results of this comparison study indicate that a Statistical model based on discriminant analysis is marginally better for prediction of the effort, than a non-parametric model based on artificial neural network. The classification method proposed in this paper may be used to identify the similar category of projects and for forecasting the software development cost and time effort. Hence, this approach would be useful for planning and preventive actions in the process of software development. The Statistical Software Package IBM SPSS 19.0 is used for the present analysis.

Index Terms — Artificial Neural Network, Cluster Analysis, Discriminant Analysis, k-Means, Lines of Code, software development estimate and Software Effort Estimation.

1 INTRODUCTION

It is of utmost important for both customers and the software companies to plan and predict certain parameters for any new project at the proposal stage to get beneficial results. Hence any Software Project Manager would be interested to determine the software development cost and effort, which will help them to allocate the budget, decide the time line, plan the resource and risks, so that an optimal operational metrics is obtained (Lionel, Victor, and Yong, 1994). Software Development Life Cycle (SDLC) is presented in Figure 1. It is essential to get the accurate cost and effort estimation for the project proposal to obtain mutual benefits. In order to evaluate cost and effort, we have to accumulate metrics from the previous projects and compare the similar characteristics with the current projects. For this purpose, many software effort estimation models have been proposed in the software industry. In general, Software Effort Estimation Models fall under following two categories: (1) Algorithmic models such as SLIM, COCOMO, Multiple Regression, Statistical models, etc. and (2) Non-Algorithmic models such as Fuzzy Logic (FL) Neural Network (NN), Case-Base Reasoning (CBR), Regression Trees, etc., (Wittig and Finnic, 1994) in order to identify the optimal estimation model and helping software project managers to take the correct decision.

Estimation is a crucial activity for successful software project management, since this is very fundamental for budgeting and project planning. The quality of the estimation is one of the factors, which determines the success of the project and facilitate to avoid the risks. Inaccurate estimations have frequently to lead to the failure of the projects. In general, within budget and On-time deliverables are important concern for the software organizations. Under-estimated projects can have unfavourable outcome on business, performance and competitiveness. Over-estimated projects can result in poor resource allocation and missed business opportunities to take on new project.

---

Figure 1. Software Development Life Cycle

The main objective of this research is to identify the similar groups of software development projects which share similar
characteristics based on the Effort of Lines of Code (LOC) by performing k-means clustering method. The data set used in this research is based on the historical development projects data from one of the major information technology (IT) company. The clusters obtained through the k-means clustering method is further investigated and validated through the use of Discriminant Analysis (DA) and Artificial Neural Network (ANN). These methods are implemented using Statistical Software Package IBM SPSS 19.0.

This paper is structured as follows:

- **Section 2** Provides the literature review about the Software effort estimation, Cluster analysis, Discriminant Analysis and Artificial Neural Network.
- **Section 3** Includes a description about Cluster analysis
- **Section 4** Provides a brief explanation about Discriminant Analysis.
- **Section 5** Includes a brief account of Artificial Neural Network.
- **Section 6** Exhibits a method for classification of software development projects and Comparison of Discriminant Analysis and Artificial Neural Network study.
- **Section 7** Presents conclusion of the present research.

## 2 LITERATURE REVIEW

Shepperd et al. (1996), classification of estimation and prediction techniques can be categories into three ways, which are algorithmic models, expert judgment and machine learning. This work mainly focuses on combining clustering using machine learning techniques. The different types of clustering methods have already been applied to several aspects of software project management, including software effort, schedule estimation, software metrics and software quality.

Discriminant Analysis (DA) can be classified as three types, namely, direct, hierarchical and stepwise. Kinnear and Gray (2001) has highlighted that direct Discriminant Analysis involves all the variables entering in the equations at once, whereas, in hierarchical Discriminant Analysis, the variables enter in the equation according to a schedule set by the researcher, and in stepwise Discriminant Analysis statistical criteria are used in determining when the variables will enter the equations.

A cluster analysis is used to identify the small, medium and high risk of software projects. Risk measurements across the levels exposed that, even low risk projects have a high level of complexity risk (Linda et al., 2004). It is associated with requirements and planning. The results suggest that, Project scope have an effect on all dimensions of risk. There are past studies about software project effort estimation on cost overrun (Procaccino, et al., 2002 and Verner, et al., 2007). The logistic regression is also used to predict the delay delivery of software projects (Takagi, et al., 2005).

The Software defect prediction models will change from environment to environment pertaining to specific defect patterns, but it does not mean building common defect prediction model (Koru and Liu, 2005). In fact, such models would be really useful and provide a starting point for development environments with no historical data. Stockburger (2007), specified the use of discriminant function to forecast group membership of a project.

Cluster analysis is an unsupervised classification method. Generally, it is useful in many research areas such as Statistics, data mining, data classification, pattern recognition and machine learning. The main objective of the cluster analysis is to partition large dataset into smaller groups such that objects within a group are similar while objects between the groups are dissimilar, in some sense. Even though there are different methods available to solve the clustering problem, the most commonly used method is k-means clustering, which is a simple and efficient partition method developed by Mac Queen in 1967 (Zhang, et al. (2006); Li and Wu, 2007 and Ye, et al., 2008).

Software Testing is the most important part in the Software Development Life Cycle. Software projects testing activities used at least 30 percent of the project overall effort. Software project testing is an essential part to ensure the software product quality (Sonali and Shaily, 2010). There is always a rigid Schedule during the software system development, thereafter reducing efforts of performing software testing management. Ogunsanmi, et al., (2011), conducted a study for various risk sources in Design and demonstrated that the projects can be classified into three risk groups of cost, time and quality using the discriminant analysis technique.

## 3 CLUSTER ANALYSIS

Cluster analysis is a statistical technique for identifying groups of objects using measurements on them (Li and Wu, 2007). The main purpose of a cluster analysis is to classify a sample of objects based on the measured variables into a number of different groups such that similar objects are placed in the same group (Manly, 2005). The clustering algorithms are generally classified into two types, which are hierarchical and non-hierarchical algorithms (Anderson, 2003), as depicted briefly in Fig 2.
3.1 Hierarchical Methods

Hierarchical clustering is one of the most popular methods and it can be either agglomerative or divisive (Anil and Richard, 1998). Agglomerative hierarchical clustering is called as bottom-up clustering method. It starts with each case being a cluster into itself and similar clusters are merged. The algorithm ends with all objects in a same cluster. Divisive Hierarchical Clustering is called as top-down clustering method. It works in the opposite direction of agglomerative clustering method. It starts with a single cluster holding all the objects, then sequentially splitting objects until every object is in a separate cluster.

3.2 Non-Hierarchical Method (k-means)

In general, Non-hierarchical cluster analysis is used when huge quantity of data is involved (Anil and Richard, 1998). It is a most common method because it allows subjects to move from one cluster to another cluster and these features are not possible in the hierarchical cluster method. The number of clusters (k) should be specified in advance in the k-means clustering technique.

- Initially, select k objects randomly from the dataset, which correspond to k-cluster centers.
- Assign each object to the cluster with which it has minimum distance.
- Calculate the mean value of the objects for each cluster, to update the centroids.
- Repeat these steps until no changes in the centroid/no relocation of objects occur.

Once achieved a “high-quality” partition of objects corresponding to optimizing a measure function defined either locally or globally, the algorithm stops (Rencher, 2002). The k-means algorithm tries to minimize the squared error function (Jain and Dubes, 1998). The squared sum error (SSE) of a partition K = (k1, k2, …, km) is defined as:

\[ \text{SSE}(K) = \sum_{j=1}^{m} \sum_{i=1}^{n_j} d^2(Q_j, f_j) \]

where the cluster k_j is a set of objects (Q_{j1}, Q_{j2}, …, Q_{jn_j}) and f_j is the centroid of k_j. Therefore, the k-means algorithm minimizes the distance between the intra-cluster (Anil et al., 1998).

The main disadvantages of K-means cluster are:
- The performance of the algorithm depends on the initial centroids. So, there is no assurance for an optimal solution.
- The user has to identify the number of clusters in advance.

4 DISCRIMINANT ANALYSIS

Discriminant Analysis (DA) is used for predicting group membership based on a linear combination of independent variables. The method of combining independent variables into a single new variable is called as discriminant function (Ogunsanmi, et al., 2011). The concepts of discriminant function is well documented by Stockburger (2007), and Kinnear and Gray (2001).

The efficiency of discriminant function is tested with a Statistic, known as Wilks Lambda (λ). This measure indicates the significant difference between the target groups (Kinnear and Gray, 2001). The Discriminant function is expressed as follows.

\[ D = w_1X_1 + w_2X_2 + w_3X_3 + … + w_nX_n + a \]  

Where D = Discriminate function;  
\( w_i = \) Weight for the i-th variable  
\( X_i = \) Score for the i-th variable; a = Constant

Discriminant function is similar to the regression equation. The w_i’s are unstandardized discriminant coefficients. This w_i’s maximize the distance between the means of the dependent variable. Good predictors tend to have large weights.

5 ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks are relatively crude electronic models similar to the neural structure of the brain. In other words, Artificial Neural Network (ANN) is immensely parallel adaptive network of simple nonlinear computing elements called Neurons (Fig. 3). An artificial neural network comprises of eight basic components : (i) neurons, (ii) activation function, (iii) signal function, (iv) pattern of connectivity, (v) activity aggregation rule, (vi) activation rule,
l) learning rule and (viii) environment (Jagannath and Bibhudatta, 2011).

Steps in effort estimation:

1. **Data Collection**: Collect development project data from previous projects like Planned Value (BCWS), Actual Cost (ACWP), Earned Value (BCWP), Cost Performance Index (CPI), Average Team Size (Team_Size) and Source Lines of Code (LOC).

2. **Division of dataset**: Divide the number of data into two parts – Training (75%) & Testing (25%).

3. **Design**: Design the neural network (NN) with number of neurons in input layers same as the number of characteristics/attributes of the project.

4. **Training**: Feed the training data set to train the neural network.

5. **Validation**: After training is over then validate the NN with the validation data set.

6. **Testing**: Finally test the created NN by feeding test dataset.

7. **Error calculation**: Check the performance of the ANN. If acceptable then stop, else again go to step (3), make some changes to the network parameters and proceed.

### 5.1 RADIAL BASIS FUNCTION (RBF)

The RBF procedure fits a radial basis function neural network, which is a feed forward, supervised learning network with an input layer, a hidden layer called the radial basis function layer, and an output layer (Figure 4). The hidden layer transforms the input vectors into radial basis functions. The RBF procedure performs prediction and classification similar to the MLP procedure. To classify the software projects, the paper focuses on Radial Basis Function (Jagannath and Bibhudatta, 2011).

The RBF procedure trains the network in two stages:

- The procedure determines the radial basis functions using clustering methods. The center and width of each radial basis function are determined.
- Given the radial basis functions, the procedure estimates the synaptic weights. Sum-of-squares error function with identity activation function for the output layer is used for both prediction and classification. Ordinary Least Squares regression is used to minimize the sum-of-squares error.

Due to this two-stage training approach, the RBF network is, in general, trained much faster than MLP.
6 METHODOLOGY

The main object of this research is to identify the similar group of effort spent for software development. We performed k-means cluster analysis, which is used to identify the underlying groups of software projects. Further, these groups are subjected to discriminant analysis and Neural Network. For this purpose, we obtained 810 historical software development project data, of which 75% has been used for training and 25% for testing purposes. Various parameters which are considered in this research are Planned Value, Actual Cost, Earned Value, Cost Performance Index (CPI), Average Team Size and Source Lines of Code (SLOC). The raw data set is normalised before subjecting it to analysis. The best possible 3 groups of projects (called Small, Medium and Large) has been identified using k-means algorithm and the results are presented in Table 1 with the Final Cluster Centers. These two methods are implemented using Statistical Software Package IBM SPSS 19.0.

These 3 groups (1-LARGE, 2-MEDIUM, 3-SMALL) obtained were further investigated with discriminant analysis (DA) and Neural Network (NN). It could assist the decision makers to identify similar types of development projects, with best effort estimation, to develop the code and test the project.

Table 1. Final Cluster Centers

<table>
<thead>
<tr>
<th>Variables</th>
<th>Cluster</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planned Value</td>
<td>0.03</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>Actual Cost</td>
<td>0.06</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>Earned Value</td>
<td>0.04</td>
<td>0.02</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>Cost Performance Index</td>
<td>0.18</td>
<td>0.18</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>Average Team Size</td>
<td>0.18</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>Source Lines of Code</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

The results are shown in Table 2. Canonical Discriminant Function 1 has Eigen value $2.031$ and Function 2 has Eigen values $0.984$. This implies that the function 1 has more variance between the groups and function 2 has lesser variance between the groups. The third column of this table indicates that the discriminant function 1 explains 67.4% of the variance between the groups while discriminant function 2 accounts for 32.6% of the variance. The last column of this table shows that the Function 1 has canonical correlation $r = 0.82$; by squaring the canonical correlation for this discriminant function $(0.82^2 = 0.67)$, we obtain the eta square that would result from a one-way ANOVA on the first discriminant function. According to the result 67% of the variability of the scores for the first discriminant function is accounted by differences among the 3 groups which are Large, Medium and Small level projects. Similarly, Functions 2 has canonical correlation $r = 0.70$, therefore, 49% $(0.70^2 = 0.49)$ of the variability of the scores for the second discriminant function is accounted for by the effort factors.

The output for significance tests and strength of relationship statistics for the discriminant analysis is shown in Table 3. The Eigen value of function 1 through 2 and function 2 have the $p$-values very close to zero $(p = 0.000)$, indicating that the functions discriminate the groups.

### 6.1 Discriminant Functions Classification

The output for classification of groups is shown in Table 4. The classification results allow us to determine how well we can predict group membership using a classification functions.

The results indicate that:

<table>
<thead>
<tr>
<th>Test of Function(s)</th>
<th>Wilks’ Lambda</th>
<th>Chi-square</th>
<th>df</th>
<th>Sig. (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 through 2</td>
<td>0.166</td>
<td>1078.907</td>
<td>12</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.504</td>
<td>411.961</td>
<td>5</td>
<td>0.000</td>
</tr>
</tbody>
</table>

The output for group classification is shown in Table 5. The
classification results allow us to determine how well we can predict group membership using a classification functions. Based on this study,

**Table 5. Classification of projects using ANN**

<table>
<thead>
<tr>
<th>Sample</th>
<th>Observed</th>
<th>Predicted</th>
<th>Percent Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Training</td>
<td></td>
<td>36</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
<td>3</td>
<td>399</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>Overall</td>
<td>Percent</td>
<td>6.40%</td>
<td>75.10%</td>
</tr>
</tbody>
</table>

| Testing |         | 7         | 9      | 0     | 43.80%  |
|         | Medium   | 3         | 139    | 0     | 97.90%  |
|         | Small    | 0         | 15     | 30    | 66.70%  |
| Overall | Percent  | 4.90%     | 80.30% | 14.80%| 86.70%  |

*Dependent Variable: Cluster Number of Case*

- For the training data set, about 90.10% of the original group cases are correctly classified by this ANN.
- For the testing data, around 86.70% of the original group cases are correctly classified by this ANN.

**6.3 Comparison of Discriminant Analysis and Neural Network**

There are few interesting and important facts about classification of Discriminant Function and Artificial Neural Networks (RBF).

It illustrates the predictive capability of each model. In the below table, out of 810 software development projects, 203 (25%) projects used for testing purpose.

**Table 6. Training grouped cases correctly classified**

<table>
<thead>
<tr>
<th>Models</th>
<th>High Accepted</th>
<th>Medium Accepted</th>
<th>Small Accepted</th>
<th>Correct Classified Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discriminant Analysis</td>
<td>53</td>
<td>396</td>
<td>137</td>
<td>96.50%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>36</td>
<td>399</td>
<td>112</td>
<td>90.10%</td>
</tr>
</tbody>
</table>

Training data classified for the Discriminant Function at 96.50% and Artificial Neural Networks (RBF) at 90.10%, which implies that discriminant analysis shows better results than neural network analysis.

**Table 7. Testing grouped cases correctly classified**

<table>
<thead>
<tr>
<th>Models</th>
<th>High Accepted</th>
<th>Medium Accepted</th>
<th>Small Accepted</th>
<th>Correct Classified Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discriminant Analysis</td>
<td>15</td>
<td>136</td>
<td>39</td>
<td>93.60%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>7</td>
<td>139</td>
<td>30</td>
<td>86.70%</td>
</tr>
</tbody>
</table>

Testing data classified for the Discriminant Function at 93.60% and Artificial Neural Networks (RBF) at 86.70%, which implies that discriminant analysis gives better results than neural network analysis.

In the Comparison study a statistical method of Discriminant Analysis is slightly better results than the non statistical method Neural Network.

**7. CONCLUSION**
Estimation of Lines of Code (LOC) and required team size are essential parameters in the tasks of software project management. In the software development life cycle, project planning, effort/cost estimate, team size are made based on the code size. In this paper, we performed k-means clustering on software development projects and identified 3 groups, namely, Large, Medium and Small projects, of software development projects based on the Effort of Lines of Code (LOC). The clusters obtained are investigated with the discriminant analysis and Artificial Neural Network. The results of this comparison study indicate that a Statistical model based on discriminant analysis is marginally better for prediction of the effort, than a non-parametric model based on artificial neural network.

REFERENCES