Implementing Decision Tree for Software Development Effort Estimation of Software Project

Sonam Bhatia, Varinder Kaur Attri
Student, Dept. of CSE, GNDU, RC, Jalandhar, India
Assistant Professor, Dept. of CSE, GNDU, RC, Jalandhar, India

ABSTRACT: Effort estimation is one of the biggest problems faced by software industry. In software planning estimation of the effort is one of the most critical responsibilities. It is necessary to have good effort estimation in order to conduct well budget. The accuracy of the effort estimation of software projects is vital for the competitiveness of software companies. For the forecasting of software effort, it is important to select the correct software effort estimation techniques. Inaccurate effort estimation can be risky to an IT industry’s economics and certainty due to poor quality or trait and stakeholder’s disapproval with the software product. This paper presents M5P decision tree Technique, for effort evaluation in the field of software development.

KEYWORDS: Effort estimation, Decision tree, M5P, Machine learning

I. INTRODUCTION

Software effort estimation is the forecasting about the amount of effort needed to make a software system and its duration [1] Good estimates play a very important role in the management of software projects.[2] . The effort is the most important cause that affecting the budget of a project. Estimating the effort with a high degree of accuracy is a issue which has not yet been solved and even the project manager has to deal with it since the beginning. Several parameters can affect the effort estimation. These parameters Incorporate Size, Category, Personnel Attributes, Complexity [3] Most of the effort estimation metrics takes the input as the software size, which can be measured with function point, LOC, object point. A number of models have been enlarged to provide the relation between size and effort. [16]

SLOC is typically used to predict the amount of effort that will be needed to establish a program, as well as to valuation programming productivity or maintainability once the software is developed[11] Effort is measured in terms of person months and duration.[4]. More recently attention has turned to a variety of machine learning techniques to predict software development effort [7][8]. Most of the projects are break down due to imprecise estimated effort, so the success of any software project depends on an initial and accurate effort estimation.[9] The purpose of Machine Learning is to provide increasing levels of automation in the knowledge engineering process, replacing much time consuming human activity with automatic techniques that improve reliability or efficiency by observing and manipulating regularities in training data.[5]

There are many reason for vary of effort estimation. These are Project approval, project management, defining of project task etc. The field of Machine Learning (ML) is devoted to develop computational methods that implement various forms of learning, in particular mechanisms capable of inducing knowledge from examples or data [10]. An important requirement is that the learning system should be able to deal with imperfections of the data. Many methods have been explored for software effort estimation, consisting traditional methods such as the COCOMO and, more recently, machine learning techniques such Linear regression, Multi-Layer Perceptron, Decision tree[2] Machine learning methods have been exploited to generate better software products ,to be part of software product and to make software development process more convenient and adequate .[6]
II. IMPORTANCE FOR EFFORT ESTIMATION AND PERFORMANCE MEASURES

A. Importance

Effort estimation is necessary for many people and different departments in an organization. At various points in project lifecycle well-defined effort estimation is essential. The computation of the effort might be used as input to project plans, determining the budget and other important procedure needed for the successful release of the software. The progress or failure of projects depends on the authenticity or reliability of effort and schedule evaluations, among other things. Early effort estimation also assists the project manager to investigate whether the available resource is effective to complete the project. As software applications have grown in size and significance, the need for reliability in software cost estimating has grown, too.

B. Performance measures

Correlation measures of the strength of a relationship between two variables, Mean Absolute error measures of how far the estimates are from actual values, Relative absolute Error (RAE) takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor. Root Mean Square Error (RMSE): RMSE evaluates the difference between value estimated by a model and the value actually observed. [9]
IV PROPOSED MODEL

Proposed model includes four steps. The steps are identifying the problem domain, scanning data, partition data into test and training or classification.

A. Data preprocessing

There are 17 attributes used in our data set and a brief description about each is presented in table 1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELY</td>
<td>Required software reliability</td>
</tr>
<tr>
<td>DATA</td>
<td>Data base size</td>
</tr>
<tr>
<td>CPLX</td>
<td>Process complexity</td>
</tr>
<tr>
<td>TIME</td>
<td>Time constraint for CPU</td>
</tr>
<tr>
<td>STOR</td>
<td>Main memory constant</td>
</tr>
<tr>
<td>VIRT</td>
<td>Machine volatility</td>
</tr>
<tr>
<td>TURN</td>
<td>Turnaround time</td>
</tr>
<tr>
<td>ACAP</td>
<td>Analyst capability</td>
</tr>
<tr>
<td>AEXP</td>
<td>Application experience</td>
</tr>
<tr>
<td>PCAP</td>
<td>Programmers capability</td>
</tr>
<tr>
<td>VEXP</td>
<td>Virtual machine experience</td>
</tr>
<tr>
<td>LEXP</td>
<td>Language experience</td>
</tr>
<tr>
<td>MODP</td>
<td>Modern programming practice</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of software tools</td>
</tr>
<tr>
<td>SCED</td>
<td>Schedule constraint</td>
</tr>
<tr>
<td>LOC</td>
<td>Lines of code</td>
</tr>
<tr>
<td>ACT-EFFORT</td>
<td>Actual effort</td>
</tr>
</tbody>
</table>

WEKA is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms. WEKA is a state-of-the-art facility for developing machine learning (ML) techniques and their application to real-world data mining problems. It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset. WEKA implements algorithms for data preprocessing, classification, regression, clustering, and association rules.

B. Decision Tree

A decision tree is a logical model that contributes in operations research, specifically in decision analysis [9]. Decision tree is a kind of tool to come out with a decision on the basis of some conditions and their possible consequences [16]. Decision tree is a procedure used for classification and regression [15]. Decision tree is a flowchart like tree structure, where each internal node stand for a test on an attribute, each branch express an outcome of the test, and each leaf node holds a class label. The root node is the topmost node in a tree [15].

Decision trees are generated from training data in a top down, general to specific direction. The initial state of tree is root node that is assigned all examples from training the training set. If it is case that all the examples belong to same class then no further decision need to be made to partition the examples and the solution is complete. If example at this node belongs to two or more classes then test is made at node that will result in split. The process is recursively repeated for each intermediate node until completely discriminating tree is obtained. M5P is powerful because it implements as much decision trees as linear regression for predicting a continuous variable. This algorithm is a multivariate tree algorithm which is appropriate for noise removal and also applies for large database. The M5P introduced by Quinlan, the model tree technique (M5) can be recognized as an extension to CART. A model tree will
fit a linear regression to the observations at each leaf rather of allowing a single value like CART. The M5P algorithm has three stages: building a tree, pruning the tree and smoothing. [4]

V. RESULT

A. M5P Implementation result

Using weka tool the classification algorithm is used to perform the experiments on COCOMO dataset. From the given dataset by using M5P algorithm, the following rules were produced on the basis of data

--- Classifier model (full training set) ---
MD pruned model tree:
(using smoothed linear models)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOC &lt;= 106.5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LOC &gt; 106.5</td>
<td></td>
</tr>
</tbody>
</table>

LM num: 1

ACT_EQRT =
86.816 * TIME=Very_High,High
+ 37.0446 * VINT=Low,High
+ 97.0746 * VEX=Nominal,High
+ 24.7876 * TOOL=High,Very_High,Low,Very_Low
+ 9.9701 * LOC
- 157.5237

LM num: 2

ACT_EQRT =
84.816 * TIME=Very_High,High
+ 37.0446 * VINT=Low,High
+ 162.7623 * VEX=Nominal,High
+ 52.9084 * TOOL=High,Very_High,Low,Very_Low
+ 4.6844 * LOC
- 161.764

LM num: 3

ACT_EQRT =
520.4254 * TIME=Very_High,High
+ 865.9092 * VINT=Low,High
+ 166.2373 * VEX=Nominal,High
+ 91.1874 * LOC
- 961.138

Number of Rules : 3

Time taken to build model: 0.17 seconds

Fig 1: RULES GENERATED BY TREE

B. Performance Measure

By using algorithm correlation coefficients, mean absolute error, root mean squared error, relative absolute error, and root relative absolute squared error were measured

--- Cross-Validation ---
--- Summary ---

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation coefficient</td>
<td>0.922</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>150.9641</td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>262.1864</td>
</tr>
<tr>
<td>Relative absolute error</td>
<td>36.0178 %</td>
</tr>
<tr>
<td>Root relative squared error</td>
<td>37.9721 %</td>
</tr>
<tr>
<td>Total Number of Instances</td>
<td>60</td>
</tr>
</tbody>
</table>

Fig2: Performance measures

On the basis of rules the following tree was generated for our data set.
VI. DISCUSSION OF RESULTS

In this analysis, we used M5P Classification tree on 60 instances of COCOMO dataset. Correlation measures was 0.922, Mean Absolute error measures was 150.9841, Relative absolute Error (RAE) was 35.0178% and Root Mean Square Error (RMSE) was 252.8864, root relative square error was 37.9721%

VII. CONCLUSION

Effort estimation is greatest problem faced by software industry. Effort is very closely related to size of the software and cost of the software hence it is very important for software industry to reduce the effort. Data mining technique like decision tree can be used to study the effort of software. The result of our paper focuses on the implementation of M5P decision tree. It is assumed that with better characteristic of decision tree can generate a specialized method to monitor the effort or performance measures and hence take necessary steps to reduce the effort.

ACKNOWLEDGEMENT

We would like to thank acknowledge almighty for his constant blessings. Then we like to thank our family and friends for helping and supporting us throughout the making of this paper.

REFERENCES


Jiawei Han Data Mining: Concepts and Techniques Second Edition, 2011


http://www.qualitymanagementconference.com/effort_estimation.php

Sonam Bhatia received B.Tech degree in Information Technology from Punjab Technical University, Jalandhar, India, in 2013, pursuing M.Tech in Computer Science Engineering from Guru Nanak Dev University, Amritsar, India. Her research area includes software engineering.