Logistic Model Trees based Approach for Prediction of Reusability of Object Oriented Software Components

Sarabpreet Singh, Pushpinder Singh, Neeraj Mohan and Parvinder S. Sandhu

Abstract—In this research paper, objective is the reuse of software components in software development. An approach which puts this idea central is called reuse-based software engineering. Software professionals have recognized reuse as a powerful means to potentially overcome the situation also known as software crisis and promises significant savings in software productivity and quality. Multiple systems. In this way a software component is developed only once, and can save out development effort multiple times. LMT (Logistic Model Trees) based approach is used to economically determining reusability of software components in existing systems as well as the reusable components that are in the design phase.

Keywords—Reuse, Logistic Model Trees, Object Oriented, Metrics.

I. INTRODUCTION
A great deal of research over the past several years has been devoted to the development of methodologies to create reusable software components and component libraries, where there is an additional cost involved to create a reusable component from scratch. That additional cost could be avoided by identifying and extracting reusable components from the already developed large inventory of existing systems. But the issue of how to identify good reusable components from existing systems has remained relatively unexplored. Our approach, for identification and evaluation of reusable software, is based on software models and metrics [1].

II. REUSE PROCESS
The process of reuse consists of four major activities:
- manage the reuse infrastructure (MRI).
- produce reusable assets (PRA).
- broker reusable assets (BRA) and
- Consume reusable assets (CRA).
Function of Manage the Reuse Infrastructure (MRI) is to establish the reuse rules, roles, and goals in the infrastructure to support reuse. The Produce Reusable Assets (PRA) activities develop, generate, or reengineer assets with the specific goal of reusability. PRA includes domain analysis and domain engineering. The Broker Reusable Assets (BRA) activity aids the reuse effort by qualifying or certifying, configuring, maintaining, promoting and brokering reusable assets. The Consume Reusable Assets (CRA) activity occurs when systems are produced using reusable assets [6].

III. OBJECT ORIENTED PARADIGM
A major factor in the invention of Object-Oriented approach is to remove some of the flaws encountered with the procedural approach. In OOP, data is treated as a critical element and does not allow it to flow freely. It bounds data closely to the functions that operate on it and protects it from accidental modification from outside functions. OOP allows decomposition of a problem into a number of entities called objects and then builds data and functions around these objects. A major advantage of OOP is code reusability.

IV. PROBLEM FORMULATION
Two popular methods for classification are linear logistic regression and tree induction, which have somewhat complementary advantages and disadvantages. The former fits a simple (linear) model to the data, and the process of model fitting is quite stable, resulting in low variance but potentially high bias. The latter, on the other hand, exhibits low bias but often high variance: it searches a less restricted space of models, allowing it to capture nonlinear patterns in the data, but making it less stable and prone to overfitting. So it is not surprising that neither of the two methods is superior in general—earlier studies [3] have shown that their relative performance depends on the size and the characteristics of the dataset (e.g., the signal-to-noise ratio). It is a natural idea to try and combine these two methods into learners that rely on simple regression models if only little and/or noisy data is available and add a more complex tree structure if there is enough data to warrant such structure. For the case of predicting a numeric variable, this has lead to ‘model trees’, which are decision trees with linear regression models at the leaves. These have been shown to produce good results [7]. A more natural way to deal with classification tasks is to use a...
combination of a tree structure and logistic regression models resulting in a single tree. Another advantage of using logistic regression is that explicit class probability estimates are produced rather than just a classification. Hence, with the objective of taking advantage of the features of the LMT (Logistic Model Trees) technique, in this study LMT (Logistic Model Trees) based approach is used to economically determining reusability of software components in existing systems as well as the reusable components that are in the design phase[11]. Inputs to the system, are provided in form of five object oriented metric values as representation of the attributes of the software component and output is be obtained in terms of reusability.

V. METHODOLOGY
Reusability evaluation System for Object Oriented Software Components can be framed using following steps:

A. Selection and refinement of metrics targeting the reusability of Object Oriented software system and perform parsing of the software system to generate the Meta information related to that Software.

The proposed five metrics for Object-Oriented Paradigm are as follows:

1) Weighted Methods per Class (WMC)
   According to this metric if a Class C, has n methods and c1, c2 …, cn be the complexity of the methods, then WMC\(C\) = c1 + c2 +… + cn. McCabe’s complexity metric is chosen for calculating the complexity values of the methods of a class; the value is normalized so that nominal complexity for a method takes on a value of 1.0. If all method complexities are considered to be unity, then WMC = n i.e. the number of methods existing in that class.
   “Tuned WMC” (TWMC) measure is used as input to the NF inference engine by restricting the WMC value in between 0 and 1 with help of sigmoid function shown below:

   \[
f(x, a, c) = \frac{1}{1 + e^{-a(x-c)}}
   \]

   Where a=10 and c=0.5.

2) Depth of Inheritance Tree (DIT)
   According to this metric Depth of inheritance of a class is "the maximum length from the node to the root of the tree". More is the depth of the inheritance tree greater the reusability of the class corresponding to the root of that tree as the class properties are shared by more derived classes under that class. Greater depth dilutes the abstraction and there is a need to set the minimum and maximum DIT value for a class as a contribution towards the reusability [19].
   “Lack of Tuned Degree of Inheritance” (LTDIT) measure is used as input to the NF inference system, in order to restrict the input value between 0 and 1.

3) Number of Children (NOC)
   According to this metric, Number of children (NOC) of a class is the number of immediate sub-classes subordinated to a class in the class hierarchy. Thus, greater is the value of NOC, greater will be the reusability of the parent class. Hence, there should be some minimum value of NOC for a parent class for its reusability [22].
   Theoretical basis of NOC metric relates to the notion of scope of properties. It is a measure of how many sub-classes are going to inherit the methods of the parent class. The definition of NOC metric gives the distorted view of the system as it counts only the immediate sub-classes instead of all the descendants of the class. The NOC value of a class (say class ‘i’) should reflect all the subclasses that share the properties of that class as shown in the following equation:

   \[
   NOC(i) = N + \sum_{i} NOC(i)
   \]

   Where N is the total number of immediate subclasses of class i.
   In order to restrict the input value between 0 and 1, we have used “Lack of Tuned Number of Children” (LTNOC) measure as input to the NF inference system.

4) Coupling Between Object Classes (CBO)
   According to this metric, “Coupling Between Object Classes” (CBO) for a class is a count of the number of other classes to which it is coupled. Theoretical basis of CBO relates to the notion that an object is coupled to another object if one of them acts on the other. Here, we are restricting the unidirectional use of methods or instance variables of another object by the object of the class whose reusability is to be measured. As Coupling between Object classes increases, reusability decreases and it becomes harder to modify and test the software system. So, there is a need to set some maximum value of coupling level for its reusability and if the value of CBO for a class is beyond that maximum value then the class is said to be non-reusable [19].
   In order to restrict the input value between 0 and 1, we have used “Lack of CBO” (LCBO) measure as input to the NF inference system.

5) Lack of Cohesion in Methods (LCOM)
   Consider a Class C 1 with n methods M 1, M 2 ..., M n. Let \(\{I_j\}\) = set of instance variables used by method M i .There are n such sets \(\{I_1\}, \{I_2\}, ..., \{I_n\}\). Let P = \(\{(I_1, I_j) | I_1 \cap I_j = \emptyset\}\) and Q = \(\{(I_1, I_j) | I_1 \cap I_j \neq \emptyset\}\). If all n sets \(\{I_1\}, \{I_2\}, ..., \{I_n\}\) are \(\emptyset\) then P = \(\emptyset\) [4]. Lack of Cohesion in Methods (LCOM) of a class can be defined as:
   \[
   LCOM = |P| - |Q|, \text{ if } |P| > |Q|
   LCOM = 0 \text{ otherwise}
   \]
   The high value of LCOM indicates that the methods in the
class are not really related to each other and vice versa. It means that low value of LCOM depicts high internal strength of the class which results into high reusability [19]. So, there should be some maximum value of LCOM after that class becomes non-reusable.

“Tuned LCOM” (TLCOM) measure is used as input to the NF inference engine by restricting the LCOM value between 0 and 1 with help of sigmoid function as shown in the following equation:

\[ f(x, a, c) = \frac{1}{1 + e^{-(a(x-c))}} \] (3)

Where \( a = 4 \) and \( c = 1.5 \)

B. Calculate the metric values of the sampled software components.

In this phase, the metric value is generated corresponding to the different selected attributes. Different metric value is evaluated according to the statistic of the attribute selected.

C. Use LMT (Logistic Model Trees) based prediction system for the Reusability Prediction

D. The performance criterion taken is the classification Accuracy%. It is the percentage of the predicted values that match with the expected values of the reusability for the given data. The best system is that having the high Accuracy value.

The number of folds is fixed to 10, as long as the number of instances in the training set is not smaller than 10. If this is the case, the number of folds is set equal to the number of instances.

Deduce the results on the 10 fold cross validation accuracy, precision and recall values [24].

In case of the two-level output value based problem, the confusion matrix has four categories: True positives (TP) are modules correctly classified as Reusable modules. False positives (FP) refer to non-reusable modules incorrectly labeled as reusable modules. True negatives (TN) correspond to non-reusable modules correctly classified as such. Finally, false negatives (FN) refer to reusable modules incorrectly classified as non-reusable modules as shown in table I.

### Table I
**Confusion Matrix of Prediction Outcomes**

<table>
<thead>
<tr>
<th>Predicted Value</th>
<th>Real Data Value</th>
<th>Reusable</th>
<th>Non-Reusable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusable</td>
<td>TP</td>
<td></td>
<td>FN</td>
</tr>
<tr>
<td>Non-Reusable</td>
<td>FN</td>
<td></td>
<td>TN</td>
</tr>
</tbody>
</table>

VI. RESULTS AND DISCUSSION

The object oriented dataset considered have the output attribute as Reusability value. The Reusability in the dataset is expressed in terms of two numeric labels i.e. 1 and 2. The label 1 represents Non-Reusable and the label 2 represents the Reusable Label. The statistics of the count of the number of examples of certain reusability label is shown in the table II.

### Table II
**Statistics of the Reusability Output Attribute in the Dataset**

<table>
<thead>
<tr>
<th>Number of Instances</th>
<th>Class1</th>
<th>Class2</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>39</td>
<td></td>
</tr>
</tbody>
</table>

The statistics shows that in the dataset, there are 48 examples of label 1 and 39 examples of label 2. The input attribute-wise statistical details of the count of the examples of the labels are shown in table III, table IV, table V, table VI, table VII. The input attributes are expressed in the three linguistic labels i.e. 1, 2, and 3. The label 1 corresponds to the Low value, label 2 corresponds to the Medium value and label 3 corresponds to the high value.

### Table III
**Statistics of the Reusability Output Attribute in the Dataset**

<table>
<thead>
<tr>
<th>No.</th>
<th>Label</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>

The graphical display of the above statistics is as shown in the Fig 1:

![Fig. 1 Bar Chart of count of examples with different the reusability level](image-url)
The given data is with five Input Attributes i.e. TWMC, LTNO, LTNC, LCBO, TLCOM, and one Output attributes named as Reusability Level of the Software Component. Then SVM clustering based algorithm is implemented in Matlab 7.4.

First of all, Randomly selection of training and test sets are made. Thereafter, Training of support vector machine classifier is performed with the training dataset created. The training data is provided to SVM in form of Matrix, where each row corresponds to an observation or replicate, and each column corresponds to a feature or variable. Groups are also provided to the SVM as Column vector, character array, or cell array of strings for classifying data in Training into two groups. It has the same number of elements as there are rows in Training. Each element specifies the group to which the corresponding row in Training belongs [19].

The trained SVM is now used to classify the test dataset and the performance of the classification is recorded in terms of the Correct Rate and Error Rate. The snapshot of the output structure named as cp is shown in the figure below:

The proposed algorithm is run for 10 times the classification Accuracy is recorded as shown in table VIII below:

<table>
<thead>
<tr>
<th>Run</th>
<th>Accuracy Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>88.3721</td>
</tr>
<tr>
<td>2nd</td>
<td>79.0698</td>
</tr>
<tr>
<td>3rd</td>
<td>83.7209</td>
</tr>
<tr>
<td>4th</td>
<td>88.3721</td>
</tr>
<tr>
<td>5th</td>
<td>74.4186</td>
</tr>
<tr>
<td>6th</td>
<td>81.3953</td>
</tr>
<tr>
<td>7th</td>
<td>79.0698</td>
</tr>
<tr>
<td>8th</td>
<td>83.7209</td>
</tr>
<tr>
<td>9th</td>
<td>88.3721</td>
</tr>
<tr>
<td>10th</td>
<td>88.89</td>
</tr>
</tbody>
</table>

As evident from the table above the Average Correct Rate or the Accuracy of the proposed system is 83.54%.

VII. CONCLUSION

In this study LMT (Logistic Model Trees) based classification approach is evaluated for Reusability Prediction of Object based Software systems using the metric based approach are used for prediction. Reusability value is expressed in the six linguistic values. Five Input metrics are used as Input and the training of the proposed system is performed, thereafter performance of the system is recorded for prediction of reusability of the software modules. After the 10 fold cross validation the Correctly Classified Instances are 65 i.e. 74.7126 %. It means Incorrectly Classified Instances are 22 i.e. 25.2874 %. The Mean absolute error and Root mean squared error calculated are 0.1182 and 0.272 respectively. The precision value of the “Reusability level 1” is the highest and the Recall value of “Reusability level 6” is the maximum among other classes and overall the F-measure value of all classes is satisfactorily high. It means that LMT algorithm can satisfactorily predict reusability level of object.
oriented software components and this can further help in improving the value of the reusability of software components.

VIII. FUTURE SCOPE

The proposed approach is applied on the C++ based software modules/components and it can further be extended to the Artificial Intelligence (AI) based software components e.g. Prolog Language based software components. It can also be tried to calculate the fault-tolerance of the software components with help of the proposed metric framework.

The research work can be extended in the following directions:

- Intelligent Component Mining or Extraction algorithms can be developed
- Early prediction of the quality of component based system
- Characterization of Software Components for easy retrieval

REFERENCES