Analyzing and Improving Object Oriented Software Design using CK Metric & Neural Network

Rashmi Singla¹ Amrita Chaudhary²
¹,²Department of Computer Science & Engineering
¹,²Krukshtara University/D.I.E.T, India

Abstract—Object Oriented Software Engineering (OOSE) is used to build a new quality software system by taking the help of existing classes. Chidamber and Kemerer (CK) metrics is calculated on OOSE model to analyze reusability. The research work presents a general model to analyze, evaluate and improve CK metric values for object oriented software using Matlab’s Neural Network Toolbox. Object Oriented Design Patterns are used to analyze various metrics and number of useful conclusions can be drawn by evaluation of these metrics that may effect on quality factors of OOSE. Various metrics have been calculated that affects the performance of Object Oriented Software Engineering and proposed a model that analyzes and improves CK metric by automating the calculation of CK metric values. Finally, assessment is performed of the theoretical and empirical evaluation procedures to determine the extent of producing useful findings. Here we are proposing a model that will use CK metric values of Object Oriented Design Patterns (obtained from Rational Rose) as input and Learning Vector Quantization (LVQ) & Back Propagation Neural Network (BPNN) for processing of input values in order to calculate and automate CK metric values. Finally, Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) have been calculated to measure accuracy as well as improvement in CK metric values.

Key words: CK Metrics, OOSE, Design Patterns, Neural Network

I. INTRODUCTION

Object Oriented Software Engineering (OOSE) is beneficial for software development as it promotes reusability, decreasing development’s time, effort, cost and hence managing complexity. OOSE is most preferably used in software industry as the mainstream approach to software engineering. Object based software development approach focuses on reuse of already existing software artifacts. Object oriented software development can increase the Software engineer’s productivity. Existing reusable classes are combined to form Object oriented software that boost up the process of developing software from beginning. Object oriented software development offers higher quality, more reliable software. The main reason is that reusable classes have been tested and therefore their quality can be assured. Object oriented technology can ease software maintenance. Object oriented software means that a large software application can be made of many small classes.

Here, in this paper we are considering Design patterns as object oriented classes that can be divided basically into 4 categories as shown in the figure 1.

A. Artificial Neural Network

An Artificial Neural Network (ANN) is an information-processing model which has been derived from biological nervous systems. It is made up of large number of highly interconnected processing elements (neurons) that works collectively to solve specific problems. Supervised learning is based on the system trying to predict outcomes for known examples and is a commonly used training method. It compares its output with the target answer and gets learnt from its mistakes. The data is feeded as input to the input layer neurons which are passed to next nodes. The free parameters - weight and bias is applied and on reaching the inputs to the next node, these are summed. The process continues until the data reach the output layer where the model gives outcome. In a supervised learning system, the predicted output is compared with the actual output. If the obtained output is same as actual output, no further change is required in the weights. But, if the predicted output is higher or lower than the actual outcome, the error is back propagated and the weights are adjusted accordingly. This process of error feeding in backward direction through the network is called "back-propagation".

B. MATLAB

MATLAB is a high-level technical computing language and interactive environment for developing algorithms, visualizing & analysis of data, and numerical computation. MATLAB product can be used to solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran. MATLAB can be used in various applications like signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology etc [29, 30].

II. PROPOSED WORK

The goal of the research work is to analyze a series of metrics proposed by various researchers e.g. Design
Patterns. A systematic analysis of the values for various metrics will be carried out and several outputs will be resulted from them. We will figure out various metrics that affects the performance of Object Oriented based Software Engineering Model and will try to propose a model that will evaluate the reusability of the system based on CK metric analysis. In the research work, CK metric suit i.e. WMC, DIT, NOC, CBO and LCOM has been used to analyze the structure of Object Oriented software model. CK metric suite does not account for potential complexity that arises from certain other Object Oriented design factors such as encapsulation and polymorphism [5, 6].

The research work analyzes, evaluate apply mechanisms to an example set of Object Oriented models. Finally, assessment is performed of the theoretical and empirical evaluation procedures to determine the extent to which they satisfy defined requirements and produce useful findings.

A. CK Metric

The metrics for object-oriented system proposed by Chidamber and Kemerer which was further improvised by other authors can be summarized as follows [1, 3]:
1) Weighted Methods per Class (WMC): This is a weighted sum of all the methods defined in a class.
2) Coupling Between Object classes (CBO): It is a count of the number of other classes to which a given class is coupled and, hence, denotes the dependency of one class on other classes in the design.
3) Depth of the Inheritance Tree (DIT): It is the length of the longest path from a given class to the root class in the inheritance hierarchy.
4) Number of Children (NOC): This is a count of the number of immediate child classes that have inherited from a given class.
5) Response for a Class (RFC): This is the count of the methods that can be potentially invoked in response to a message received by an object of a particular class.
6) Lack of Cohesion of Methods (LCOM): A count of the number of method-pairs whose similarity is zero minus the count of method pairs whose similarity is not zero.

B. Software Design Pattern Analysis

Design patterns represent a recurring solution to a software development problem within a particular context. In the software design phase, they are frequently used to create abstractions which are needed to accommodate future changes and to maintain architectural integrity. De-coupling of the major components from the system is made easy in order to vary each component independent of other.

Here we are going to discuss software design patterns in detail.

Consider the example of Abstract Factory design pattern as shown in figure 2. CK metric value analysis is shown in Table 1 as shown in figure 2. CK metric analysis for all other design patterns may be obtained by following same process.

![Fig. 2: Decorator design](image_url)

Table 1: CK Metric Analysis for Decorator Design

<table>
<thead>
<tr>
<th>Class/Matrix</th>
<th>NOM</th>
<th>DIT</th>
<th>NOC</th>
<th>CBO</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Concrete Component</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Decorator</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Concrete Decorator A</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Concrete Decorator B</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

C. Artificial Neural Network

Input vectors are needed to be classified according to given targets using LVQ network training. LVQ (learning vector quantization) neural networks consist of two layers. The first layer maps input vectors into clusters that are found by the network during training. The second layer maps merges groups of first layer clusters into the classes defined by the target data. The total number of first layer clusters is determined by the number of hidden neurons. The larger the hidden layer the more clusters the first layer can learn, and the more complex mapping of input to target classes can be made [2, 4].

In order to improve the CK Metric, problem of automating CK metric evaluation on the basis of some attributes has to be resolved. The automation can be done using Feed Forward Back Propagation Neural Network which is a supervised neural network. With the help of Back Propagation network we will feed some metric values and train the network on the basis of target data. After training, simulation can be performed on the same values or other test values to get Mean Square Error (MSE) and PSNR (Peak Signal to Noise Ratio) to find the accuracy of got values.

D. MATLAB

MATLAB (MATrix LABoratory) is a numerical computing environment and fourth-generation programming language, which is developed by MathWorks [16]. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB have many features for add-on
application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems [29, 30].

III. IMPLEMENTATION & RESULT ANALYSIS

A. Neural Network Implementation

This is divided into 2 stages:

- Developing a Learning Vector Quantization Neural Network and train the network to classify the input vectors according to targets.
- Back Propagation Neural Network creation and train it to automate the evaluation of CK metric calculation.

A feed-forward back-propagation network is created using the function newff. It requires three arguments and returns the network object. The first argument is a matrix of sample R-element input vectors. The second argument is a matrix of sample S-element target vectors. The sample inputs and outputs are used to set up network input and output dimensions and parameters. The third argument is an array that contains sizes of each hidden layer. More optional arguments can be provided. For instance, the fourth argument is a cell array containing the names of the transfer functions to be used in each layer. The fifth argument contains the name of the training function to be used. If only three arguments are supplied, the default transfer function for hidden layers is tansig and the default function for the output layer is purelin. The default training function is trainlm. newff command creates the network object and also initializes the weights and biases of the network; therefore the network is made ready for training [29]. The command used is given as under:

```matlab
net = newff(inBack',outBack',4,{},'trainlm');
```

Before training a feedforward network, initialization of the weights and biases is carried out. To automatically initializes the weights the newff function is used, but it may needed to be reinitialized which can be done using init function that takes a network object as input and returns a network object with all weights and biases initialized.

No. of Layers taken: 2 (Input layer + hidden layer+ Output Layer)
No of epochs: 200
Learning Rate Parameter: 0.01
Goal defined to achieve: 0.01

The steps that are followed to obtain the improvement in CK metric are followed as stated earlier. Number of training data is 21 x 6 i.e. 126 elements. The experiment results for training of neural network in the MatLab are shown in figure 3 & 4.

The usefulness of back propagation neural network to generalize and insensitive to the missing data benefits in our work. For training purpose all 21 design patterns are considered with 4 metrics as input and 2 metrics as output. Number of epochs taken is 200 to achieve high accuracy as shown in figure 4.

IV. CONCLUSION

Object Oriented Software Development act as best means to achieve the benefit in programmer productivity, system flexibility, and overall system quality required for IT revolution. High quality software development is getting more important than before with the increase of software applications.

Here we have used 2 neural networks i.e. Learning Vector Quantization used for classification of CK metric values (which are used as input) obtained from Design Patterns. After the classification, we have used Back Propagation Neural Network in which any 4 CK metric values are given as input and 2 other CK metric values are used as output. Here we have provided flexibility to choose any 4 values as input and other 2 values as output. After training the neural network, we may provide the same 4 CK metric values as input for test data and respective to that 2 CK metric values will be obtained. Thereby improving the CK metric in terms of evaluating time & calculation effort by automating the calculation of only 4 CK metric values. However, still we have to calculate 4 CK metric values. By using the example of design patterns and neural network
techniques, we have proposed a model that provides improvement for CK metric values of Object Oriented Software model, but from the above results, it may be said that it can be more improved if more realistic data is available to act as training set and also if neural network techniques with fuzzy techniques are considered. Also, the model is concise to applicability of software design patterns. Wherever software design patterns are not used, the model may not be applicable there.

V. LIMITATIONS & FUTURE SCOPE
The technique implemented for evaluating and improving performance of CK metric values for Object Oriented Software Engineering is using design patterns. The proposed model can give better values if good amount of data is provided with some real values. Proposed model analyzes, evaluates and improves the CK metric values calculation based on historical results of project values based on Object Oriented behavior, a software company may provide error free data. We can improve the performance of our system by tuning the inputs from software industry.

This model has been designed keeping in mind software design patterns for evaluation of CK metric so in the area where software design patterns are not applicable, the model may not provide correct results. The model can be made generic not concise to applicability of software design patterns by using CK metric values obtained from some legalized software.

REFERENCES


[29] MatLab Neural Network Tool Box Product Help