ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 3, Issue 2 (Apr. - Jun. 2015)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2015)

Prediction of Defects in Software Using K-Nearest Neighbour Algorithm for Cost Reduction

[I]Shivangi Govil, [II]Nupur Chugh
[I]Student, M.tech CSE, Galgotias University, Greater Noida, U.P. India.
[II]Assistant Professor, M.tech CSE, Galgotias University, Greater Noida, U.P. India

Abstract
Software Reliability is becoming an important part of any software system. It is an important factor in software quality since it calibrates software failures. Clustering is a process of partitioning a group of data objects into a small number of clusters on the basis of similarity. K-nearest neighbor is a clustering method that aims to find the defects in the new software based on the previously developed software. In this paper, we will examine the defects of the software with the help of k-means algorithm. Finally, we propose a defect prediction system with high performance which manages to decrease the cost of software system simultaneously.

Keywords
Clustering, Software Reliability, K-nearest neighbor algorithm..

I. Introduction

Software defect can be defined as variations from the expected operation of the software that might lead to software failures or any imperfections related to software, leading to economic loss, is an important issue to consider in software development life cycle. As software development is a human process, its quite obvious that many errors are to be generated during the software development lifecycle. It is a real challenge to develop a error free and quality software because of increasing complexity and constraints which occurs when the software is being developed. Defective software possess higher risk by increasing the development and maintenance cost and customer disapproval. Moreover, software development companies cannot risk their company's prestige by developing low quality software.

It is therefore, considerable to locate fault prone software modules at an early stage of the software development. Tracing the fault as early as possible in software development process will not only help in improving the cost of the software, but also helps in attaining the customer satisfaction and reliability of the software developed. Developing reliable, fault-free and high quality software system is a complex and expensive task[1]. It is important to predict the fault because it helps in developing a high-quality and reliable software with reduced cost. Software defect prediction is the process of finding the defective modules in the software[1].

Defects can be introduced in several phases of the software development life cycle and are classified on the basis their origin. The four types of defect artifacts based on this classification are Requirements Defects (e.g. leaving out a required Cancel option in an Input screen), Design Defects (e.g. error in the algorithm), Coding Defects (e.g. looping 9 instead of 10 times) and Documentation Defects (e.g. incorrect instructions in User's manual to Cancel an operation). Capers Jones [10] in the glossary of his book under "Defect Origins" also has a category called "bad fixes".

A Software defect prediction models consists of independent variables collected and measured during software development life cycle and dependent variable which can be faulty and non-faulty. A primary model is being developed using the training data i.e. dependent and independent variables of previously developed Software. Then this model can be used in future to predict the defect of software.

Data mining is a process to convert raw data into useful information. It is a process designed to explore and analyze large amounts of data to find consistent patterns, trends or relationships among variables and then to validate the findings by applying the observed patterns to new set of data.

Data mining is a process to convert raw data into useful information. It is a process designed to explore and analyze large amounts of data to find consistent patterns, trends or relationships among variables and then to validate the findings by applying the observed patterns to new set of data [7].

A. Data mining algorithms

Data mining can be divided into two tasks- predictive tasks and descriptive tasks. . The ultimate aim of data mining is prediction and predictive data mining is the most common type of data mining and has the most direct applications to business [3]. Predictive data mining depends on formulas that compares the past successes and failures and then predicts the future behavior of the system.

Data Mining is about explaining the past and predicting the future by means of data analysis. It is Multidisciplinary field which combines statistics, machine learning, artificial intelligence and database technology [4]. Predictive modeling is a process by which a model is developed to predict the future outcome. If the outcome is categorical, it is called as classification and if it is numerical it is called regression. Descriptive mining tasks distinguish properties of the data in a target data set.

The various data mining techniques used for predictions are discussed as:

1. Regression: It is a statistical process to examine the relationship among variables. It examines the relationship between the dependent variable and independent variables. The relationship is expressed in the form of an equation that predicts the dependent variable as a linear function of independent variable. Linear Regression: Y=a+bX+u.
2. Association Rule Mining: It is a method for discovering interesting relationships between variables in large databases. It is concerned about finding association or correlations among sets of items in a database. It basically deals with finding rules that will predict the occurrence of item based on the occurrence of other items in the system.
3. Clustering: Clustering is a way to categorize a collection of items into groups or clusters whose members are similar in some manner. It is task of categorizing a set of items in such a way that items in the same cluster are similar to each other and dissimilar to items in other clusters.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

4. Classification: It consists of predicting a certain outcome based on a given input. Classification technique uses input data, also called training set where all objects are already tagged with known class labels. The objective of classification is to analyze and learns from the training data set and develop a model accordingly. This model is then used to classify test data for which the class labels are not known.

a. Neural Networks: Neural Networks are the non linear predictive models which can learn through training and resemble biological neural networks in structure. A neural network consists of inter-connected processing elements called neurons that work together in parallel within a network to produce output.

b. Decision Trees: A decision tree is a predictive model which can be used to represent both regression and classification models in the form a tree structure. It is also called as a hierarchical model because of decisions and their consequences. It is a tree comprising of decision nodes and leaf nodes. A decision node has at least two branches. Leaf nodes represents a classification or decision.

c. Naive Bayes: It is based on Bayes theorem with independence assumption between independent variables. Naïve Bayes Classifier is based on an assumption that the presence or absence of a particular feature of a class in not related to the presence or absence of any other feature.

d. Support Vector Machines: Support vector machines (SVM) are based on the concept of decision planes that define decision boundaries. A decision plane is the one that separates a set of objects that are having different class membership[1]. SVM is basically a classifier method that performs classification task by constructing hyper plane in a multi-dimensional space that separates cases of different class labels. It supports both regression and classification.

e. Case Based Reasoning: Case based reasoning refers to solving new problems based on the similar past problems and using old cases to explain new situations in a more accurate way. It works by comparing new unclassified records with previously known patterns and examples. A simple example of a case based learning algorithm is k-nearest neighbor algorithm. It is simple and easy algorithm that stores all available cases and classifies new cases based on a similarity measure i.e. a distance function.

B. K-Nearest Neighbor Algorithm

We consider each of the characteristics in our training set as a different dimension in some empty space, and take the value an observation has for this characteristic to be its coordinate in that dimension, so getting a set of points in that empty space. We can then consider the similarity of two points to be the distance between them in this space under some appropriate metric.

The way in which the algorithm decides which of the points from the training set are similar enough to be considered when choosing the class to predict for a new observation is to pick the k closest data points to the new observation, and to take the most common class among these.[8]

1 NN :

- Predict the same value/class as the nearest instance in the training set.[9]

K-NN:

- Find the k closest training points (small |xi – xo|) according to some metric, for ex. Euclidean, Manhattan, etc.)
- Predicted class: majority vote
- Predicted value: average weighted by inverse distance This is why it is called the k Nearest Neighbors algorithm.

The Algorithm:

The algorithm can be summarized as:

1. A positive integer k is specified, along with a new sample
2. We select the k entries in our database which are closest to the new sample
3. We find the most common classification of these entries
4. This is the classification we give to the new sample

II. Related Work

In [2], software defect prediction work focuses on the number of defects remaining in a software system. A prediction of the number of remaining defects in an inspected artifact can be used for decision making. An accurate prediction of the number of defects in a software product during system testing contributes not only to the management of the system testing process but also to the estimation of the product's required maintenance. Defective software modules cause software failures, increase development and maintenance costs, and decrease customer satisfaction. It strives to improve software quality and testing efficiency by constructing predictive models from code attributes to enable a timely identification of fault-prone modules. In this paper, they discussed data mining techniques that are association mining, classification and clustering for software defect prediction. This helps the developers to detect software defects and correct them. Unsupervised techniques may be used for defect prediction in software modules, more so in those cases where defect labels are not available.

In [3], the framework consists of two components: (i) scheme evaluation and (ii) defect prediction. Historical data are divided into two parts: training set for building learners with the given learning schemes, and a test set for evaluating the performances of the learners. It is very important that the test data are not used in any way to build the learners. This is a necessary condition to assess the generalization ability of a learner that is built according to a learning scheme, and further to determine whether or not to apply the learning scheme, or select one best scheme from the given schemes. defect prediction stage, according to the performance report of the first stage, a learning scheme is selected and used to build a prediction model and predict software defect

In [4], software fault prediction based on mining of code and design metrics has been considered by many researchers. Fault detection systems predict faults by using software metrics and data mining techniques. Various classifiers have already been used in this case; however Naïve Bayes classifier is the most commonly used. According to the results of a study performed by Lessman, no significant performance difference could be detected among the top 17 classifiers. In this paper, they extended that study by examining the performance of 37 different classifiers in fault detection systems. They review the results and aim to choose an appropriate classifier (Bagging) which depicts a higher performance and accuracy compared to the others. Finally, they proposed a fault detection system with higher performance which manages to decrease the cost of software fault detection simultaneously. They investigate classifier selection by evaluating the methods

on a number of other datasets. There results indicate that Bagging classifier has the highest performance in fault detection.

In [5],an oversampling method is proposed that using the number of fault per modules and distribution of fault among modules. Two prediction models Naïve Bayes and Logistic regression are applied to two dataset from NASA MDP project .Sampling and over sampling method are used. The result of T test and the nonparametric method of Wilcoxon test showed that oversampling method significant influence on the prediction of both LR and NB model. Author proposed a human-in-the-loop CBR tool that ad business knowledge to the data mining algorithm. CBR build better prediction model that detect the lower bound on the number of instances. Using three sub sampling techniques (over, under and micro sampling) to find the lower bound the number of training instances. Naive Bayes and j48 methods are used in case of over and under sampling and Naive Bayes is used in case of micro sampling.

In [6],software defect detection has been an important topic of research in the field of software engineering for more than a decade. This research work aims to evaluate the performance of supervised machine learning techniques on predicting defective software through data mining algorithms. This paper places emphasis on the performance of classification algorithms in categorizing seven datasets (CM1, JM1, MW1, KC3, PC1, PC2, PC3 and PC4) under two classes namely Defective and Normal. In this study, publicly available data sets from different organizations are used. This permitted us to explore the impact of data from different sources on different processes for finding appropriate classification models. We propose a computational framework using data mining techniques to detect the existence of defects in software components. The framework comprises of data pre-processing, data classification and classifier evaluation. In this paper; we report the performance of twenty classification algorithms on seven publicly available datasets from the NASA MDP Repository. Random Tree Classification algorithm produced 100 percent accuracy in classifying the datasets and hence the features selected by this technique were considered to be the most significant features. The results were validated with suitable test data.

III. Conclusion

A defect shows the undesirable aspects of the software quality. Defects may be occurs in three phases of software at the time of requirement analysis, Design and in coding .In proposed methods we first identify defects from the given set of projects for improving quality of the process then train all defects via neural network for later use , than classify these defects and on the basis of these defects we predict defects using K-Nearest neighbor algorithm, for causing failures before occurring .Identification of defects at the early stages so that wastage in process and product development can be eliminated . This algorithm is able to classification and prediction itself hence reduce complexity and cost of the research.

References

[1]. *Ravinder Kaur et al, "A review on software defect prediction model based on different data mining techniques", IJCSMC,Vol: 3, Issue :5, May- 2014.*

[2]. *P.J.Kaur , Pallavi , "Data Mining Technique for Software Defect Prediction", International journal of Software and Web Science, 12-317,2013*

[3]. *K.B.S.Sastray, B.V.Subba Rao,K.V.Sambasiva Rao,"Software Defect Prediction From Historical Data,IJARCSSE-2013,Volume-3.Issue 8,Aug 2013.*

[4]. *A.A.Shahrjooi Haghighi,"Applying Mining Schemes to Software Fault Prediction:A Proposed Approach Aimed at Test Cost Reduction .WCE-2012,July 4-6,2012 London,UK*

[5]. *N.Azeem, S.Usmani, Analysis of data mining based software defect prediction techniques,Global Journal of Computer Science and Technology, Volume 11, Issue 16,Version 1.0 ,Sep-2011.*

[6]. *Ramani,R.G.,Prediction Fault Prone Software Modules using feature selection and classification through data mining algorithm,IEEE-2012*

[7]. *www.statsoft.com/Textbook.Data-Mining- Techniques/ mining*

[8] *Oliver Sutton, "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction",Feb-2012.*

[9] *L. Kozma, k Nearest Neighbours Algorithm. Helsinki University of Technology, Available: http://www.lkozma.net/knn2.pdf, 2008*

[10] *Jones C., Assessment and Control of Software Risks, 1994, Yourdon Press Computing Series*