Predicting software fault using Clustering Approach

Submitted By: Swati Shree Mohanty,Asst. Prof In Raajdhani Engineering College, Bhubaneswar
Niva Tripathi, Asst.Prof in Raajdhani Engineering College, Bhubaneswar
Pravat Routray,Asst.Prof in NM Institute of Engineering and Technology
Yogamaya Mohapatra, Asst. Prof in Capital Engineering College.

ABSTRACT

Predicting defect gives chance to the development team to again test the modules or files which are likely to be defective. If more time is given to the defective modules, the project resources would be efficiently utilized. Software defect prediction decreases the total cost of the project with an increase in overall project success rate. Detecting faults perfectly can help direct test effort, reduce the cost with an increase in quality of software. This Paper shows specific methods of fault prediction for software safety and improvement in the quality of software.

I INTRODUCTION

The main objective of this paper is to predict the fault that tends to occur while classifying the dataset also tries to improve the quality of software. Developing a defect free software system is very difficult task and sometimes there are some unknown faults or deficiencies found in software projects where there is a need of applying carefully the principles of the software development methodologies. By spending more time on the defective modules and no time on the non-defective ones, the resources of the project would be utilized better and as a result, the maintenance phase of the project will be easier for both the customers and the project owners.

When we look at the publications about Fault prediction we saw that in early studies static code features were used more. But afterwards, it was understood that beside the effect of static code metrics on Fault prediction, other measures like process metrics are also effective and should be investigated. For example, Fenton and Neil (1999) argue that static code measures alone are not able to predict software Faults accurately.

To support this idea if software is Faulty this might be related to one of the following:

- The specification of the project may be wrong due to differing requirements or missing features.
- Because of improper documentation realization of the project is too complex.
- Scarce and incorrect requirements results in poor design.
- Developers are not qualified enough for the project.
- The software life cycle methodologies might not be followed very well.
- Improper and incomplete testing of software.

Such faulty software classes may increase development & maintenance cost, due to software failures and decrease customer's satisfaction.

The main objective of this paper is to predict the fault that tends to occur while classifying the dataset.

- Hyper Quad-Trees are applied for finding initial cluster centers for K-Means algorithm.
- The overall error rates of this prediction approach are compared to other existing algorithms and are found to be better in most of the cases.

II .RELATED WORK

Previous work of faulty software components enables verification experts to concentrate their time and resources on the problem areas of the software system under development. One of the main purposes of these models is to assist in software maintenance budgeting.

Among various clustering techniques available in literature K-means clustering approach is most widely being used? Different authors apply different clustering techniques and expert-based approach for software fault prediction problem. They applied K-Means[8][9] and Neural-Gas techniques on different real data sets and then an expert explored the representative module of the cluster and several statistical data in order to label each cluster as faulty or non faulty. And based on their experience Neural-Gas-based prediction approach performed slightly worse than K-Means clustering-based approach in terms of the overall error rate on large data sets. But their approach is dependent on the availability and capability of the expert. Seliya and

Khoshgoftaar proposed a constrained based semi-supervised clustering scheme. They showed that this approach helped the expert in making better estimations as compared to predictions made by an unsupervised learning algorithm. [1] a Quad Tree- based K-Means algorithm has been applied for predicting faults in program modules. The aim of their topic is twofold. First, Quad- Trees are applied for finding the initial cluster centers to be input to the K-Means Algorithm. Bhattacherjee and Bishnu [1] have applied unsupervised learning approach for fault prediction in software module. An input threshold parameter delta governs the number of initial cluster centers and by varying delta the user can generate desired initial cluster centers. The clusters obtained by Quad Tree-based algorithm were found to have maximum gain values. Second, the Quad-tree based algorithm is applied for predicting faults in program modules. Supervised techniques have however been applied for software fault prediction and software effort prediction There is no solution to find the optimal number of clusters for any given data set in K-Means. The overall error rates of this prediction approach are compared to other existing algorithms and are found to be better in most of the cases. In this paper I try to find the better centroid than Quad-tree algorithm by using Hyper Quad-tree which will give as a input to the K-Means algorithm for lowers the error rate and effective software fault prediction. Due to some defective software modules, the maintenance phase of software projects could become really painful for the users and costly for the enterprises. That is why predicting the defective modules or files in a software system prior to project deployment is a very crucial activity, since it leads to a decrease in the total cost of the project and an increase in overall project success rate .

III. OVERVIEW

This paper shows the study of K-Means clustering algorithm Quad tree algorithm and Hyper Quad-tree algorithm then proposed system architecture for software fault prediction, expected result using confusion matrix and conclusion.

3.1 K-Means clustering algorithm

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group page is done. At this point we need to re- calculate k new centroids as bar centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done.

3.1.1. The algorithm is composed of the following steps

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated

3.1.2. Limitations of K-Means

The cluster centers, thus found, serve as input to the clustering algorithms. However, it has some inherent drawbacks-

- The user has to initialize the number of clusters which is very difficult to identify in most of the cases.
- It requires selection of the suitable initial cluster centers which is again subject to error. Since the structure of the clusters depends on the initial cluster centers this may result in an inefficient clustering.
- The K-Means algorithm is very sensitive to noise.

3.2. Quad Tree

This data structure was named a Quad tree by Raphael Finkel and J.L. Bentley in 1974. A similar partitioning is also known as a Q-tree. The Quad Tree-based method assigns the appropriate initial cluster centers and eliminates the outliers hence overcoming the second and third drawback of K-Means clustering algorithm.

Common features of quad tree

- They decompose space into adaptable cells.
- Each cell (or bucket) has a maximum capacity.
- When maximum capacity is reached, the bucket splits.
- The tree directory follows the spatial decomposition of the Quad tree.

Figure1. Shows the simple Quad tree representation.

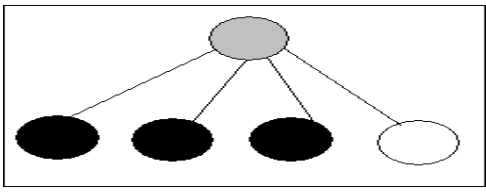


Figure1. Simple Quad Tree.

3.2.1. Some definitions of notations and parameters

- Minimum: User defined threshold for minimum number of data points in a sub bucket.
- Maximum: User defined threshold for maximum number of data points in a sub bucke

- Black leaf bucket: A sub bucket having more than MAX number of data points of the parent bucket.
- Gray bucket: a sub bucket which is neither white nor black.
- δ: User specified distance for finding nearest neighbors.

3.2.2. Quad Tree Algorithm [8] [9]:

- For each class:
- Find the minimum and maximum x and y co-ordinates.
- Find the midpoint using the values obtained in the previous step.
- Divide the spatial area into four sub regions based on the midpoint.
- Plot the points and classify regions as white leaf buckets or black leaf buckets.
- The white leaf buckets are left as such.
- The Center data-points of each black leaf bucket are calculated for all black leaf buckets.
- The mean of all the center points obtained in the previous step is calculated.
- The computed mean gives the centroid point necessary for that class.

3.2.3. Limitations of Quad Tree

- The user has to initialize the number of clusters which is very difficult to identify using quad tree algorithm.
- It is not providing the exact centroid.

3.3 Hyper Quad Tree

The Hyper Quad Tree-based method assigns the appropriate initial cluster centers and eliminates the outliers hence overcoming the second and third drawback of K-Means algorithm that is

- Hyper Quad-Trees are applied for finding initial cluster centers for K-Means algorithm. User can generate desired number of cluster centers that can be used as input to the simple K-Means.
- Second, the centroid obtained by the Hyper Quad Tree is more accurate than Quad tree.

Figure2 shows a simple hyper quad tree representation of data set dots denotes the data:

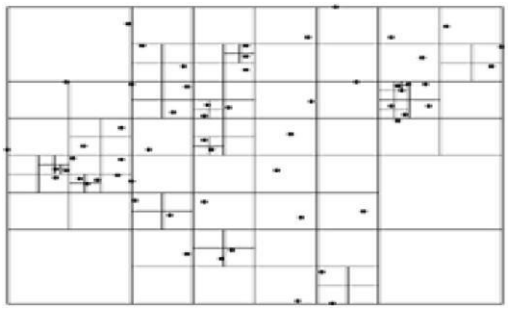


Figure2. Hyper Quad Tree [7]

Hyper Quad-Trees are expected to give better cluster centers than the Quad-tree because

- It has an eight-way branching tree whose nodes are associated with axis- parallel boxes.
- The d-dimensional analogue is known variously as a multidimensional Quad-tree and a hyper quad tree.
- It divides the regions recursively so that no region contains more than one data point.

Algorithm to generate a hyper quad tree is as follows

1. Select Random Node
2. Initialize current node = root node
3. While current node is not a leaf node do
4. Generate a random number n
5. If n<p then //n= Number of data points
6. Break the while loop
   //P=Random path termination probability
7. Else
8. Randomly select one children
9. current node=selected node
10. Require: Hyper Quad Tree
11. end if
12. end while
13. select the current node

Input: Dimension, Data set, Min,

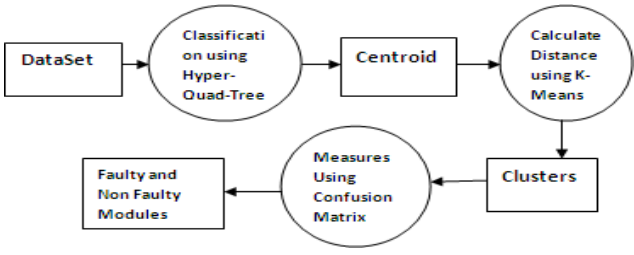
Max Output: Centroid

IV. THE PROPOSED SYSTEMS

The proposed system is ―Software fault prediction using clustering approach‖ that classify given data using Hyper Quad-tree algorithm.

The system consists of 3 modules

- Create dataset parser
- Data set is given as input to the Hyper Quad-tree algorithm in which we Create cells, insert cell, label bucket, split cell, spatial decomposition
  Input: Dimension, Data
  set Output: Centroid
- Centroid points obtained using the Hyper Quad-tree is given as an input to the K-Means to get better clusters it Calculates the distance, Shuffle data points according to distance, If centroids are stable then stop. The output of this will be set of clusters
- Measure the Faults in terms of FPR, FNR and ERROR using confusion matrix.



are placed along the columns. For example, a False Actual label implies that the module is not faulty. If a not faulty module (Actual label—False) is predicted as non-faulty (Predicted Label—False) then there is the condition of cell A, which is True Negative, and if it is predicted as faulty (Predicted label—True) then there is the condition of cell B, which is False Positive. Similar definitions hold for False Negative and True Positive. The False positive rate is the percentage of not faulty modules labeled as faulty by the model and the false negative rate is the percentage of faulty modules labeled as fault free and Error is the percentage of mislabeled modules. The following equations are used to calculate these FPR, FNR, Error, and Precision [1]

$$FPR = \frac{B}{A+B} \quad (1)$$

$$FNR = \frac{C}{D+C} \quad (2)$$

$$ERROR = \frac{B+C}{A+B+C+D} \quad (3)$$

TABLE 1.Confusion Matrix

	Predicted Labels		
Actual Labels		False (Non-Faulty)	True (Faulty)
	False (Non-Faulty)	True Negative A	False Positive B
	True (Faulty)	False Negative C	True Positive D

The above performance indicators should be minimized. A high value of FPR would lead to wasted testing effort while high FNR value means error prone modules will escape testing.

In this paper, for calculating the measures, if any metric value of the centroid data point of a cluster was greater than the threshold, that cluster was labeled as faulty and otherwise it was labeled as non-faulty. After this the predicted fault labels will compare with the actual fault labels. Also the clusters can be label according to the majority of its members (by comparing with metrics thresholds) but this increases the complexity of the labeling procedure since all the modules in the cluster need to be examined.

CONCLUSION AND FUTURE SCOPE

Hyper Quad-tree based K-Means clustering algorithm evaluates the effectiveness in predicting faulty software modules as compared to the original Quad-tree based K-Means algorithm. Also it will find better the initial cluster centers for K-Means algorithm. By using hyper quad I try to meet the convergence criterion faster and hence it results in lesser number of iterations. Also there will be reduction in time and computational complexity by reducing NOI. Better throughput with lower error rates of classification.

In this paper I am not focusing on automatic initialization of number of clusters this will be the future work for better software fault prediction using clustering Approach.

ACKNOWLEDGMENT

This is a small review of my post graduate project work that I am going to start to implement .I specially thank to my Guide for his assistance.

REFERENCES

[1] P.S. Bishnu and V. Bhattacherjee, Member, IEEE‖ Software Fault Prediction Using Quad Tree-Based K-Means Clustering Algorithm‖ *IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 6, June 2012*

[2] P.S. Bishnu and V. Bhattacherjee, —Outlier Detection Technique Using Quad Tree,‖ *Proc Int'l Conf. Computer Comm. Control and Information Technology, pp. 143-148, Feb. 2009.*

[3] P.S. Bishnu and V. Bhattacherjee, —Application of K-Medoids with kd- Tree for Software Fault Prediction,‖ *ACM Software Eng. Notes, vol. 36, pp. 1-6, Mar. 2011.*

[4] V. Bhattacherjee and P.S. Bishnu, —Software Fault Prediction Using KMedoids Algorithm,‖ *Proc. Int'l Conf. Productivity, Quality, Reliability, Optimization and Modeling (ICPQROM '11), p. 191, Feb. 2011.*

[5] J. Han and M. Kamber, —*Data Mining Concepts and Techniques*‖, second ed, pp. 401-404. Morgan Kaufmann Publishers, 2007.

[6] Parvinder S. Sandhu, Jagdeep Singh, Vikas Gupta, Mandeep Kaur, Sonia Manhas, Ramandeep Sidhu‖ A K-Means Based Clustering Approach for Finding Faulty Modules in Open Source Software Systems‖ ,*World Academy of Science, Engineering and Technology 48 2010*

[7] Michael Laszlo and Sumitra Mukherjee, Member, IEEE, —A Genetic Algorithm Using Hyper-Quadtrees for Low-Dimensional K-means Clustering‖, *IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 4, april 2006*

[8] Leela Rani.P, Rajalakshmi.P.,‖ Clustering Gene Expression Data using Quad-tree based Expectation Maximization Approach‖ *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA,Volume 2– No.2, June 2012 – www.ijais.org*

[9] Meenakshi PC, Meenu S, Mithra M, Leela Rani.P,‖ Fault Prediction using Quad-tree and Expectation Maximization Algorithm‖, *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 2– No.4, May 2012 – www.ijais.org*

[10] P.S. Bishnu and V. Bhattacharjee, —A New Initialization Method for KMeans Algorithm Using Quad Tree,‖ *Proc. Nat'l Conf. Methods and Models in Computing (NCM2C), pp. 73-81, 2008.*