ISSN:0975-9646

Comparision between Quad tree based K-Means and EM Algorithm for Fault Prediction

Swapna M. Patil
Dept.Of Computer science and Engineering,Walchand Institute Of Technology,Solapur,413006

R.V.Argiddi
Assistant Professor
Dept.Of Computer science and Engineering,Walchand Institute Of Technology,Solapur,413006

Abstract: Fault prediction will give one more chance to the development team to retest the modules or files for which the defectiveness probability is high. 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. Software fault prediction decreases the total cost of the project and increases the overall project success rate. The perfect prediction of where faults are likely to occur in code can help direct test effort, reduce costs and improve the quality of software. This Paper shows specific methods of fault prediction for software safety that directly address the root causes of software Faults and improve the quality of software.

Keywords: Quad Tree, K-Means clustering, Expectation Maximization Algorithm, Iris Dataset, Clustering, Classification.

1. 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.

This paper focuses on clustering by partition based method namely K-Means algorithm and model based method namely, EM algorithm. Fig.1 [4] indicates a sample scatter plot diagram clustered using the partitioning based K-Means algorithm. Fig.2 [4] indicates a sample scatter plot clustered using the EM algorithm.

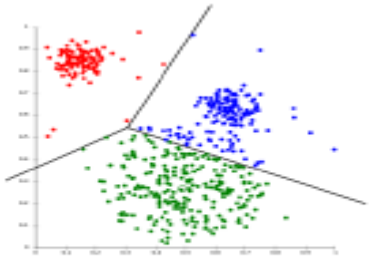

Fig.1

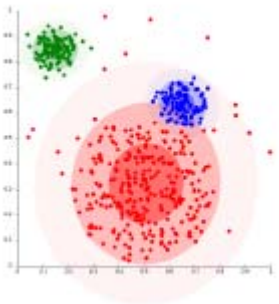

Fig.2

2. 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 ap-

ply 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 .

.

3. COMPARISION BETWEEN QUAD TREE BASED K-MEANS AND QUAD TREE BASED EM ALGORITHM

3.1. 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.

Figure3. Shows the simple Quad tree representation.

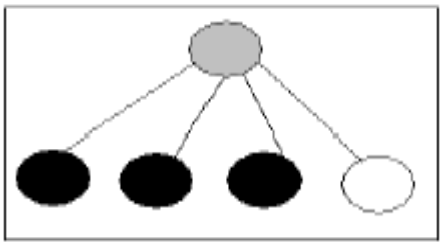

Figure3. Simple Quad Tree.

3.1.1. 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 bucket.
- White leaf bucket: A sub bucket having less than MIN number of data points of the parent bucket. Fig. shows an illustration of a white leaf bucket.
- 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.
- User specified distance for finding nearest neighbors.

3.1.2. 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 K-MEANS ALGORITHM

3.2.1 Description

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.

3.2.2 Working of K-Means Algorithm

1. Input the centroid points obtained using the quad tree algorithm as the initial cluster centers for the first iteration.

2**.** Compute distance between each data point and each centroid using the distance formula:

|(x2-x1)|+|(y2-y1)| = distance

3. Repeat

- (Re)assign each data point to the cluster with which it has the minimum distance.
- Update the cluster means for every iteration.
- Until clustering converges.

3.2.3. 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.

3.3 EXPECTATION MAXIMIZATION(EM) ALGORITHM

EXPECTATION MAXIMIZATION(EM) is a well established clustering algorithm in the statistics community. EM is a distance based algorithm that assumes the dataset can be modeled as a linear combination of multivariate normal distribution and the algorithm finds the distribution parameter that maximize a model quality measure, called log likelihood. The EM algorithm is an extension of the K-Means algorithm [3][7].

3.3.1 *Implementation of E-M Algorithm*

The general E-M algorithm is comprised of the following simple steps:

1. **Initialization**
   Initialize the distribution parameters, such as the means, covariances and mixing coefficients and evaluate the initial value of the log-likelihood (the goodness of fit of the current distribution against the observation dataset)';
2. **Expectation**
   Evaluate the responsibilities (i.e. weight factors of each sample) using the current parameter values;

$$p(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right)$$

3. **Maximization**
   Re-estimate the parameters using the responsibilities found in the previous step;
4. **Repeat**
   Re-evaluate the log-likelihood and check if it has changed; if it has changed less than a given threshold, the algorithm has converged.

3.3.2 Advantages of Fault Prediction using Quad Tree and Expectation Maximization Algorithm

The benefits of using EM as a replacement to K-Means algorithm are observed as follows:

1. The algorithm meets the convergence criterion faster and hence it results in lesser number of iterations.
2. Reduction in time and computational complexity
3. Will work despite limited memory (RAM)
4. Better throughput with lower error rates of classification

4. EXPERIMENTAL DESIGN

4.1 Dataset

The dataset that has been used for the purpose of experimental design in this paper is the popular Iris dataset [2]. It is a multivariate dataset. This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. Predicted attribute is the class of iris plant. This is an exceedingly simple domain. There are 4 attributes in this dataset. The attribute information are as follows:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. Class: Iris setosa, Iris versicolor, Iris virginica.

4.2 Comparison Metric/ Evaluation Parameter

Classification of the dataset using both the clustering algorithms proves that EM is a better fit [6] for clustering than K-Means. The two algorithms are compared for higher accuracy and efficiency using the metric "Error Rate". The evaluation parameters are the correctly classified and incorrectly classified data points. Based on these parameters, the error rate is evaluated using the following formula:

Per Species:

Actual Classified–Correctly Classified = Incorrectly Total

Overall Error Rate,

Error= (TI) / (TC+TI)

Table 1

Actual Label	Predicted Label (Correctly Classified)	Predicted Label (Incorrectly Classified)
Species 1	Precisely Labeled	Mislabeled
Species 2	Precisely Labeled	Mislabeled
Species 3	Precisely Labeled	Mislabeled
TOTAL	Summation	Summation

Table 1 illustrates the blueprint of the table used to calculate the error rates for each algorithm. The actual number of instances in every species of flower is listed in the left corner of the table. After classification using any one of the clustering algorithms, the number of correctly classified and incorrectly classified data points are determined and noted in the right hand side of the table. This enables the computation of the error rates.

4.1.1 Parameters and Definitions

TI - total number of incorrectly classified species

TC - total number of correctly classified species

5. THE PROPOSED SYSTEMS

The proposed system is ―Software fault prediction using clustering approach that classify given data using Quad-tree algorithm. The system consists of 3 modules

- Create dataset parser
- Data set is given as input to the 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 Quad-tree is given as an input to the K-Means to get 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 Centroid points obtained using the Quad-tree is also given as an input to the EM algorithm 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
- Observe the Faults in terms of Graphical representation.

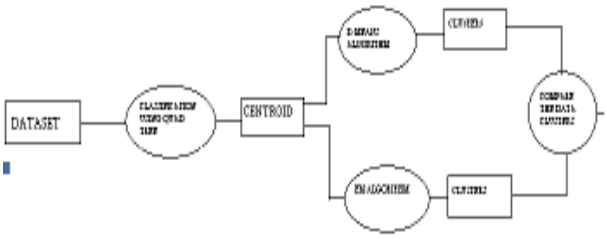

Fig 4:System Archetecture

6. RESULTS

Using the formulae outlined above, error rates are calculated for both the clustering algorithms separately. The computed result is shown via charts for comparison purposes. Fig.5 indicates a bar chart that shows the comparison of error rates for both the algorithms. It proves that EM algorithm is more accurate than K-Means owing to lower error rates as shown. Visibly lower error rates are seen for the EM clustering algorithm, when compared to K-Means.

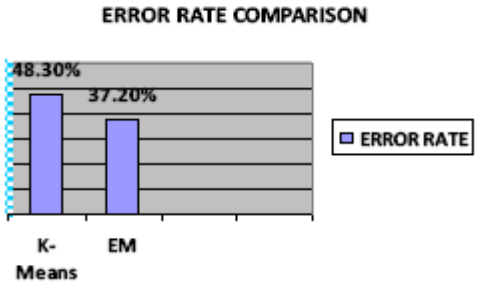

Fig.5

7.CONCLUSION

Combining the Quad Tree approach and the EM algorithm gives a clustering method that not only fits the data better in the clusters but also tries to make them compact and more meaningful. Using EM along with Quad Tree makes the classification process faster. With K-means, convergence is not guaranteed but EM guarantees elegant convergence. The proposed approach starts with a huge set (the popular Iris dataset [2]). The proposed system obtains the appropriate initial cluster centers through Quad Tree. These centroids serve as input to the EM algorithm, thus increasing the chances of finding the best clusters. The overall error rates of the proposed system are found comparable to other existing approaches. In fact, in the case of the Iris dataset, the overall error rates of the proposed approach have considerably reduced and are fairly acceptable. Results are shown, via charts indicating the effectiveness of the proposed approach.

8. FUTURE WORK

An extension of this paper would be to use a HQ Tree based EM clustering model [10]. The HQ tree is used as a replacement to the traditional Quad Tree approach so as to obtain even more precise cluster centers/centroids. A HQ tree is a D-dimensional analogue of a quad tree. Every node of a HQ tree is associated with a bounding hyper box and every non leaf node has 2D children. Thus HQ Trees are expected to yield better cluster centers as compared to the Quad Tree approach.

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] http://archive.ics.uci.edu/ml/datasets/Iris

[3] J. Han and M. Kamber, Data mining Concepts and techniques, 2nd edition, Morgan Kaufmann Publishers, pp. 401-404, 2007.

[4] http://en.wikipedia.org/wiki/Cluster_analysis

[5] J. Han and M. Kamber, Data mining Concepts and techniques, 2nd edition, Morgan Kaufmann Publishers, pp. 401-404, 2007.

[6] Osama Abu Abbas, Computer Science Department, Yarmulke University, Jordan, "Comparisons between data clustering algorithms" The international Arab Journal of Information Technology,Vol.5,No.3,July 2008.

[7] P.S.Bishnu and V. Bhattacharjee, "Software Fault prediction using Quad tree based K-Means method," IEEE transactions on Knowledge and Data Engineering ,Vol. PP, No.99, May 2011

[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] M. Laszlo and S. Mukherjee, "A Genetic Algorithm Using Hyper-Quad trees for Low-Dimensional K-Means Clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no4, pp. 533-543, 2006.