Asian Journal of Science and Applied Technology
ISSN: 2249-0698 Vol. 6 No. 1, 2017, pp.38-41

Programming Fault Prediction Using Quad Tree-Based Fluffy C-Means Clustering Algorithm

S. Ravichandran[1], M. Umamaheswari[2] and R. Benjohnson[3]
[1]Research Scholar in Department of Computer Science,
Bharathiar University, Coimbatore, Tamil Nadu, India
[2]Professor in Department of Information Technology,
RRASE College of Engineering, Chennai, Tamil Nadu, India
[3]Assistant Professor in Department of Computer Applications,
Coimbatore Institute of Management and Technology,
Coimbatore, Tamil Nadu, India
E-mail: ravi17raja@gmail.com & karpagaravi15@gmail.com
druma_cs@yahoo.com, benjohnsonr@gmail.com

Abstract - **Software measurements and blame information having a place with a past programming form are utilized to construct the product blame expectation show for the following arrival of the product. Unsupervised procedures like bunching might be utilized for blame expectation as a part of programming modules, all the more so in those situations where blame marks are not accessible. In this paper a Quad Tree-based Fuzzy C-Means calculation has been connected for anticipating deficiencies in program modules. The points of this paper are twofold. In the first place, Quad Trees are connected for observing the underlying group focuses to be contribution to the Fuzzy C-Means Algorithm. An information edge parameter oversees the quantity of introductory bunch focuses and by shifting the limit the client can create wanted beginning group focuses. The idea of grouping increase has been utilized to decide the nature of bunches for assessment of the Quad Tree-based introduction calculation when contrasted with other instatement procedures. These bunches got by Quad Tree-based calculation were found to have most extreme pick up qualities. Second, the Quad Tree based calculation is connected for anticipating shortcomings in program modules. The general blunder rates of this forecast approach are contrasted with other existing calculations and are observed to be better in the vast majority of the cases.**
Keywords: **Quad Tree, C-Means Algorithm, Fuzzy logic,**

I. INTRODUCTION

Blames in programming frameworks keep on being a noteworthy issue. Programming bug is a blunder, defect, misstep, disappointment, or blame in a PC program that keeps it from carrying on as proposed. Product blame is a deformity that causes programming disappointment in an executable item. In programming designing, the non-conformance of programming to its necessities is usually called a bug. Most bugs emerge from slip-ups and blunders made by individuals in either a program's source code or its outline, and a couple are brought on by compilers delivering naccurate code.

Knowing the reasons for conceivable imperfections and in addition recognizing general programming process ranges that may require consideration from the introduction of a venture could spare cash, time and work. The likelihood of early assessing the potential defectiveness of programming could help on arranging, controlling and executing programming improvement exercises. Expectation of blame inclined modules in programming improvement prepare and generally utilized the metric based approach with machine learning strategies to show the blame forecast in the product modules. Measurements is characterized as "The persistent use of estimation based systems to the product advancement process and its items to supply important and convenient administration data together with the utilization of those methods to enhance that procedure and its items". Programming measurements is about estimation and these are relevant to every one of the periods of programming advancement life cycle from start to support. As the lion's share of flaws are found in a couple of its modules so there is a need to research the modules that are influenced extremely when contrasted with different modules and appropriate support should be done in time particularly for the basic applications.

Exhibit Software Fault Prediction framework needs to apply quad tree for observing the underlying group focuses to be contribution to the Fuzzy C-Means Algorithm. An info edge parameter that oversees the quantity of starting group focuses and by shifting the client can produce coveted introductory bunch focuses. The idea of bunching addition has been utilized to decide the nature of groups for assessment of the Quad Tree-based instatement calculation when contrasted with other introduction systems. The groups acquired by Quad Tree-based calculation were found to have most extreme pick up qualities. The Quad Tree-based calculation is connected for anticipating hortcomings in program modules.

The Quad Tree-based calculation partitions an underlying information space into pails and proceeds until all cans are either dark or white leaf basins as showed in Fig. 1 and Fig. 2.

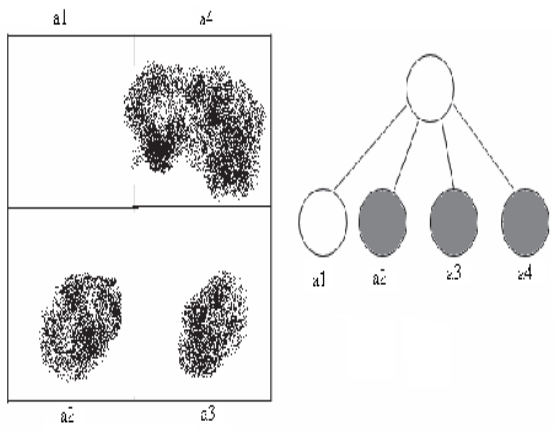

Fig.1Quad Tree implementation for 4 – quadrants

In Fig.1 the first division into four buckets is done. Out of these, three buckets are gray while one is white. In Fig. 2 the gray buckets are further subdivided, while the white one is left as such. At this stage, one of the sub buckets is labelled as a black leaf bucket.

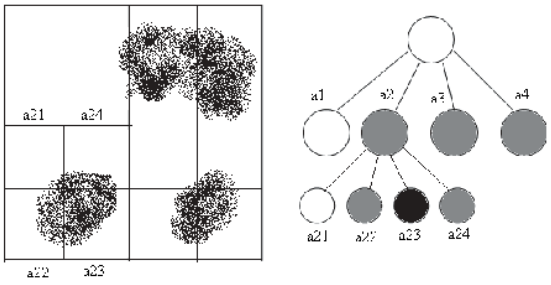

Fig.2 Quad Tree implementation for 16 – quadrants

II. RELATED WORK

The master based approach for programming issue expectation issue applies K-Means and Neural-Gas methods on various genuine information sets and afterward a specialist investigated the agent module of the group and little factual information with a specific end goal to name every bunch as blame inclined or not blame inclined. Also, in view of their experience Neural-Gas-based expectation approach performed somewhat more regrettable than K-Means grouping based approach as far as the general blunder rate on vast information sets. Be that as it may, their approach is subject to the accessibility and ability of the master.

It proposed an obliged based semi-administered bunching plan. They demonstrated that this approach helped the master in improving estimations when contrasted with forecasts made by an unsupervised learning calculation.

It proposed a grouping and measurements edges based programming flaw forecast approach and investigated it on three datasets. The fundamental commitment of their paper is the utilization of measurements edges with or without grouping strategies and the evacuating of the commitment of a specialist help. In any case, the determination of the bunch number is done heuristically in this grouping based model as well. In this study, we utilize x-implies grouping strategy and our model does not require the choice of bunch number. Rather than a correct bunch number, an interim is given to the x-implies calculation.

It has connected unsupervised learning approach for blame expectation in programming module in. In their work, the false negative rates (FNR) for the grouping based approach are not as much as that for measurements based approach, while the false positive rates (FPR) are better for the measurements based approach. The general mistake rates for both methodologies continue as before.

Hereditary calculation has been utilized for advancing focuses as a part of the K-Means calculation furthermore to find a decent dividing. The k-implies calculation is broadly utilized for grouping as a result of its computational effectiveness. Given n focuses in d-dimensional space and the quantity of craved group's k, k-implies looks for an arrangement of k bunch focuses in order to minimize the entirety of the squared Euclidean separation between every point and its closest bunch focus. Nonetheless, the calculation is extremely delicate to the underlying choice of focuses and is probably going to meet to segments that are altogether substandard compared to the worldwide ideal. This study introduce a hereditary calculation (GA) for advancing focuses in the k-implies calculation that all the while recognizes great allotments for a scope of qualities around a predetermined k. The arrangement of focuses is spoken to utilizing a hyper quad tree built on the information. This representation is misused in our GA to create an underlying populace of good focuses and to bolster a novel hybrid operation that specifically passes great subsets of neighboring focuses from guardians to posterity by swapping sub trees. Test comes about demonstrate that GA finds the worldwide ideal for information sets with known optima and discovers great answers for substantial re-enacted information sets.

III. QUAD TREE BASED INSTATEMENT ALGORITHM

A. Quad Tree

A Quad Tree in two dimensional spaces is a 4-way stretching tree that speaks to recursive disintegration of space utilizing separators parallel to the facilitate hub. At every level a square subspace is isolated into four equivalent size squares. This information structure was named as Quad. The meaning of a Quad Tree for a set O of information focuses inside a n dimensional hyper 3D square µ is as per the following:
Let µ= [d 1µ : d' 1µ] x [d 2µ : d' 2µ] x - x [d : d'].

On the off chance that the quantity of information focuses in any pail is not as much as limit then the Quad Tree comprises of a solitary leaf where the set O and the hypercube µ are put away. At every stage each pail gets

subdivided into 2 nμ sub pails. Give us a chance to consider the division of cans for n = 2. Let μd μd 1L d 2L, μd 1R d 2L 1L d 2R, indicate the four quadrants of μ. B. Parameters and Definitions μd n 1R nμ.

B. Parameters and Definitions

MIN: client characterized edge for least number of information focuses in a sub container.
MAX: client characterized limit for most extreme number of information focuses in a sub pail.
δ: client determined separation for finding closest neighbors.
White leaf basin: a sub pail having less than MIN percent of information purposes of parent.
Dark leaf basin: a sub pail having more than MAX percent of information purposes of the parent basin.
Dark can: a sub basin which is not one or the other white nor dark.
R_k**:** neighborhood set of focus ck of a dark leaf basin.
C: set of bunch focuses utilized for introducing K-Means calculation.

C. Estimation of Metric Thresholds

With a specific end goal to decide adequate measurements edges, there are three techniques depicted as takes after: 1) Experience and Indications from writing: The edge values are indicated by experimental specialists, already presented in the writing. 2) Tuning machine: This approach utilizes a vault of dangerous things (defective modules). In like manner, there are picked edge values that expand the number of effectively recognized things. 3) Investigation of numerous adaptations: This strategy does not parameterize a procedure with a few limits, yet includes an imperative time perspective for each presumed element. The limits are LoC, CC, UOp, UOpnd, Top, TOpnd, NOI and SSE.

D. Assessment of Fault-inclined Parameters

For computing the assessment parameters, if any metric estimation of the centroid information point of a bunch was more noteworthy than the edge, that bunch was named as broken and else it was marked as non-broken. After this the anticipated blame names were analysed with the genuine blame marks. The accompanying conditions are utilized to compute these FPR, FNR, and Error.

$$FPR = \frac{B}{A+B'}$$

$$FNR = \frac{C}{D+C'}$$

$$Error = \frac{B+C}{A+B+C+D}$$

E. The Initialization Algorithm

Input: Max%, Min%, Data set (O), δ
Output: Number of centers $|C|$ and the centers C

```
1.  initialize the data space as a gray bucket;
2.  while there are gray buckets
3.          {
4.          select a bucket;
5.          divide it into 2^n sub buckets; // n is the dimension
6.          label the sub buckets as white leaf bucket, black leaf
            bucket or gray bucket;
7.          for every black leaf buckets calculate center (c_{i,(1≤i≤m)});
            // m is the number of black leaf buckets;
8.          }
9.  C = Φ;
10. label all centers c_{i,(1≤i≤m)} as unmarked;
11. for i = 1 to m do ℜ_i = c_i;
12. for each neighborhood ℜ_{i(1≤i≤m)}
13.         {
14.         if there exist an unmarked center in ℜ_i then
15.                 {
16.                 while there is an unmarked center c_k in ℜ_i
17.                         {
18.                         select c_k and label it as marked;
19.                         find δ-nearest unmarked neighbors
                            of c_k and include them in ℜ_i;
20.                         }

21.                 for all c_k ∈ ℜ_i calculate the mean m_i and call it
                    the cluster center;
22.                 C = C ∪ {m_i};
23.                 }
24.         }
25. return C and |C|;
```

IV. THE FUZZY C-MEANS CALCULATION

Fluffy c-implies (FCM) is a technique for bunching which permits one bit of information to have a place with at least two bunches. Each case can have a place with each bunch with a diverse participation reviews somewhere around 0 and 1 for this calculation. A difference work is minimized and centroids which minimize this capacity are recognized. The general ventures of this calculation are,

I. Introduce the participation work haphazardly as per this condition.

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1, \ldots, n$$

II. Compute centroids as per this condition.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}}$$

III. Compute disparity esteem concurring to this condition.

$$J(U, c_1, c_2, \ldots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2}$$

Stop, if the change contrasted with past emphasis is underneath an edge level.
IV. Compute another u as indicated by this condition.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}}$$

Go to step 2.

The execution can be contrasted and different grouping calculations in light of pick up esteem. The ideal number of groups is said to happen when the intercluster separation is augmented (or intercluster similitude is minimized) and the intracluster separation is minimized (or intracluster similitude is augmented). The grouping pick up achieves a greatest esteem at the ideal number of groups. The rearranged recipe for figuring of pick up is as per the following:

Gain = ∑k=1k(vk-1) || z0- z0k||2,

It introduces the forecast blunder investigation for the QDC (Quad-tree based Fuzzy CMeans) approach when contrasted with other approaches, specifically, two phase approaches with basic K-Means with six traits (KM), Catal et al. Two phase approach (CT), Catal et al. Single stage approach (CS), Naı¨ve Bayes (NB) and Linear Discriminant Investigation (DA) (with ten times cross approval setting). QDC, KM, CT, and CS approach, and additionally NB and DA have considered six traits from the said information set.

To look at the execution of QDC for instatement of Fuzzy C-Means with GM (Worldwide K-Means calculation) [6] and KMeans have been executed. The separation parameter d for the DD calculation has been obtained by different races to get the craved number of groups. These separation values have been said in Table 5. The parameters for assessment are number of emphasess (NOI) which checks the quantity of emphasess of Fuzzy C-Means to land at the meeting criteria, Sum of squares blunder (SSE), Gain and rate of Error.

V. CONCLUSION AND FUTURE WORK

In this paper, we have assessed the viability of Quad Tree based Fuzzy C-Means bunching calculation in foreseeing broken programming modules as contrasted with the first C-Means calculation. Quad Trees are connected for finding the underlying bunch places for Fuzzy C-Means calculation. On the off chance that the client expects to shape a sought number of bunches for K-Means calculation, the Quad Tree-based calculation can give K introductory bunch focuses to be utilized as contribution to the basic Fuzzy C-Means calculation. This is encouraged by shifting the estimation of the edge parameter which is contribution to the Quad Tree calculation. The general mistake rates of programming flaw expectation approach by QDC calculation are discovered similar to other existing calculations. Actually, in the instance of AR4 and AR5 information sets, the general blunder rates of QDC are tantamount with the managed learning approaches NB and DA. The QDC calculation works as a viable introduction calculation. The quantity of emphases of Fuzzy CMeans calculation is less on account of QDC and present Error give reasonably satisfactory qualities.

REFERENCES

[1] S. Zhong, T.M. Khoshgoftaar, and N. Seliya, "Unsupervised Learning for Expert-Based Software Quality Estimation," Proc. IEEE Eighth Int'l Symp. High Assurance Systems Eng., pp. 149-155, 2004.

[2] C. Catal, U. Sevim, and B. Diri, "Clustering and Metrics Threshold Based Software Fault Prediction of Unlabeled Program Modules," Proc. Sixth Int'l Conf. Information Technology: New Generations, pp. 199-204, 2009.

[3] S. Albayrak, F. Amasyalı, "Fuzzy c- means clustering on medical diagnostic

[4] N. Seliya and T.M. Khoshgoftaar, "Software Quality Classification Modeling Using the PRINT Decision Algorithm," Proc. IEEE 14th Int'l Conf. Tools with Artificial Intelligence, pp. 365-374, 2002.

[5] V. Bhattacherjee and P.S. Bishnu, "Unsupervised Learning Approach to Fault Prediction in Software Module," Proc. Nat'l Conf. Computing and Systems, pp. 101-108, 2010.
Likas, N. Vlassis, and J. Verbeek, "The Global K-means Clustering Algorithm," Pattern Recognition, Vol. 36, pp. 451-461, 2003.

[6] R. Ammon, C. Emmersberger, T. Greiner, A. Paschke, F. Springer, and C. Wolff, "Event-Driven Business Process Management," Proc. Second Int'l Conf. Distributed Event-Based Systems (DEBS '08), July 2008

[7] Principles and Applications of Distributed Event-Based Systems, A. Hinze and A. Buchmann, eds. IGI Global, 2010.

[8] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin, "Complex Event Processing over Uncertain Data," Proc. Second Int'l Conf. Distributed Event-Based Systems (DEBS '08), pp. 253-264, 2008

[9] K. Gomadam, A. Ranabahu, L. Ramaswamy, A. Sheth, and K. Verma, "A Semantic Framework for Identifying Events in a Service Oriented Architecture," Proc. IEEE Int'l Conf. Web Services (ICWS '07), pp.545-552, July 2007.

[10] P.R. Pietzuch, B. Shand, and J. Bacon, "Composite Event Detection as a Generic Middleware Extension," IEEE Network, Vol. 18, No. 1, pp. 44-55, Jan./Feb. 2004.

[11] R. O'Callahan and J.-D. Choi, "Hybrid dynamic data race detection," in PPoPP '03, 2003, pp. 167–178.