© 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

A Neighborhood Probability Based Agglomerative Clustering for Test Case Prioritization in Regression Testing

Anju Bala
Maharshi Dayanand University, Rohtak, Haryana, India

ABSTRACT

In this paper, main intention is test case prioritization of test cases such that the testing endeavor reduces significantly while the code coverage remains more or less same. This is accomplished by using clustering approach such that the test cases are selected from each cluster thereby ensuring uniform distribution of code coverage. Our proposal is to expressed a innovative technique for test case prioritization using clustering approach. We used neighborhood probability based agglomerative clustering approach and compared the performance with density based K-means clustering for our investigation.

Keywords : Clustering, Test Case Prioritization, Density based K-means, Regression Testing

I. INTRODUCTION

Clustering is one of the most trendy unsupervised learning techniques (i.e. used for connecting the causative gap between input and output observation). Clustering is "the means of systematize objects into arrays whose members are analogous in some ways". fundamentally, clustering is to locate the internal set of unlabeled information. In clustering, we organize the information in the shape of packets or we can say into clusters. There are various clustering techniques such as Test case prioritization techniques schedule test cases in order to enhance their effectiveness according to some criterion. Test case prioritization concerns with the identification of the perfect test cases. The purpose of this technique is to rally some performance goals like rate of fault detection, increase the effectiveness etc. pace of fault detection is used to assess how rapidly faults are detected within process of testing. This gives feedback to system which is under test. The main purpose of prioritization will be minimizing the test suits [8].

Test case prioritization is used to systematize and implement the test cases in order to save cost and time. Test case prioritization is more efficient and widely used by the testers. Many researchers introduced more schemes for test case prioritization in regression testing.

II. LITERATURE SURVEY

Daniel Di Nardo et al. [8] have recommended a prioritization approach which is based on the finer grained coverage criteria. This technique trusts on further coverage using finer grained coverage criteria (block, basic block and decision) The main purpose of this paper is to assess and distinguish between coverage based prioritization techniques: total coverage versus additional coverage along with and without the use of modification information.

Hettiarachchi Charitha et al. [9] have suggested an approach which employ risk levels of latent defect types to find unsafe necessities. This approach is a new test case prioritization technique. It assign precedence to the test cases based on the link

engaging test cases. In this paper an experiential study is performed using an open source program written in Java with manifold versions and requisite documents.

Mitrabinda Ray et al. [10] have presented a criticality estimation method of component within a system, based upon design documents. According to this method, priority is given to the components used for test. A test case selection approach is genetic algorithm-based technique. A component dependency diagram (CDD) is a directed Graph that used for slicing wrt. Different scenarios are extracted. Component intensity is proportional to its priority that are solving by genetic algorithm-based technique. Muthusamy Thillaikarasi et al. [12] have presented a innovative approach for test case prioritization for earlier fault revealing in the regression testing process. The execution of urged technique has been applied on a banking application project. APFD metric is used to calculate the efficacy and finding are balanced with random ordered execution. Earlier, the recommended algorithm is a lot of improved in prior fault detection than random techniques and also enhance fault revealing rate in testing stage.
R. Beena et al. [13] have planned a innovative technique for test case selection and test case prioritization practice for regression testing. Proposed technique is very effectual in term of cost and time engage in regression testing.

From the literature survey, we have found that very little work has been done on test case prioritization which helps to trim down the human efforts and also diminish the cost. The problem of test case prioritization has acquire significant consideration over the last few years as software testing forms a foremost section of the entire software development process. Rene´e C. Bryce et al. have argued the blueprint of software development is precisely dependent on the testing effort.

Thus we work on cost diminution by prioritizing test cases and successively run the tests for the choosy test cases as per the offered time and manpower. However, number of test cases accessible which can spend a lot of time and effort. A selective number of test cases requires to be selected which would be otherwise used for the same function. The priorities of the test cases require to be preferred on the basis of several parameters. Moreover, from the literature survey, it was found that tree concept is not applied to reduce test case having maximum coverage information. Therefore, we intend to prioritize the test cases to achieve their coverage information. We apply it on density based k-means clustering algorithm.

III. PROPOSED METHODOLOGY

A. Agglomerative Clustering

We procure a quite distinctive approach and prompt a method that concomitantly considers all data points as prospective exemplars.

Pseudo-code

```
Given data_point I and data_point k:
        Result = -infinity
        for_eachdata_point z such_that (z is not k) :
                temporary = availability [i,z]+
similarity [i,z]
                if (temporary greater_than result) :
                result = temporary
final_result = similarity [i, k] – result
for each data point pair [i, j] do
        compute a[i, j], r[i, j], and a[i, j] ;
end for
for each data point pair [i, j] do
    if r[i, j]≥0 or a[i, j]+s[i, j]≥maxk_=j{a[i, k]+s[i,
k]}then
        link data point pair [i, j];
    end if
end for
for t = 1 to T do
   for each linked data point pair [i, j] do
      update r[i, j] and a[i, j] ;
  end for
end for
```

The technique's performance is also compared with K-means as given below.

In 1967, MacQueen proposed K- mean clustering algorithm, which is simplest unsupervised learning algorithms. Recommended algorithm is used to solve clustering problem. In this algorithm, data set are classified in a very simple way. For each cluster, k centroids are defined firstly. These centroids must place in a tricky ways with the intent to provide different result because of different location. For better result, clusters may place far away from each other as much as possible. Subsequently phase is to deduce distance between data points in Dataset and the cluster centers and allocate the data point to its marginal cluster. Usually Euclidean distance must be consider to evaluate the distance. After this process preliminary grouping is done. Then new centroids are calculated (formula is given below to calculate the centers). Finally, the intended purpose of this algorithm is to minimizing an objective function,

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \text{ ---------- (1.1)}$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is used to measure distance between a data point $x_i^{(j)}$ and the cluster center c_j

Advantages of K- mean Clustering:

- ✓ When data sets are well separated or distinct from each other this algorithm gives best result.
- ✓ It is easier to understand.
- ✓ It produce tighter clusters than other clustering algorithms, especially in clusters are globular.
- ✓ It is fast, robust than other.
- ✓ In case of large dataset, if we keep K small, then K-means gives faster computation .

Disadvantages of K-means Clustering:

- ✓ It is difficult to predict-value.
- ✓ It is not easy to handle noisy data and outlier.
- ✓ This algorithm fails for non-linear data set.
- ✓ This algorithm does not work well with different size and density of clusters.

A. Steps of implementation:

- ✓ Obtaining data set of test cases from test suites.
- ✓ Remove outlier from test cases.
- ✓ To improve its efficiency by using density information, apply K-means algorithm.
- ✓ Density based K-means clustering algorithm apply on the test cases so that the test cases can be clustered effectively and ready to be prioritized.
- ✓ Form a minimum spanning tree based on Prim's algorithm to select a sub-list of test cases such that the code coverage remains almost same.
- ✓ Compare the code coverage of this sub-list with the entirety number of test cases if taken.
- ✓ And at last, result comparison .

IV. RESULTS

Our whole experiments are performed in MATLAB framework, which is used for prioritizing the test cases.

Our main efforts is to improve the rate of fault detection within less time by prioritizing the test cases. To achieve this, we used innovative Density Based K-means clustering algorithm for test case prioritization. Initially, implement K-means algorithm in MATLAB framework. Then, apply DBSCAN algorithm to improve the density of this algorithm. By using the standard data set, density based K-means algorithm is used to make the clusters on the basis of their density.

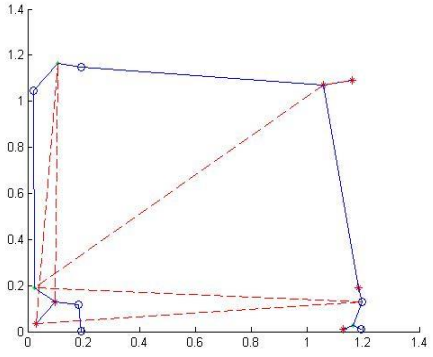


Figure 1. Representing the total and sub-list of test cases

Figure 1 represents the whole number of test cases which are clustered. The blue lines symbolize the

minimum spanning tree linking all the nodes while the red dotted lines symbolize the minimum spanning tree linking the sub list of test cases. As seen, the sub list is selected intelligently by the minimum spanning tree such that, some points are taken from each cluster.

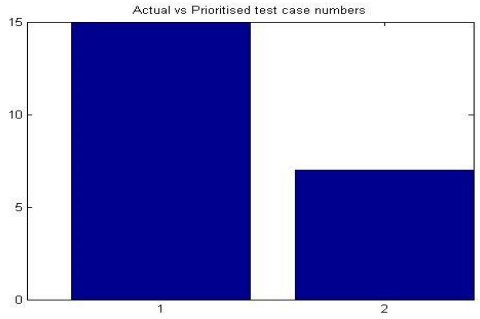


Figure 2. Actual Vs Prioritized Test Case Numbers

In Figure 2 bar plot shows the total number of test cases taken and the sub list of test cases.

In the diagnosis module we help the patient in disease diagnosis; this will help to reduce the treatment time till the appointment with the specialist is fixed. This will assist to do all require formalities and test before hand which can reduce the wastage of time and money by reducing the consultation time and consultation fee atleast for one appointment. Following will be the scenario multi-agent system for diagnosis.

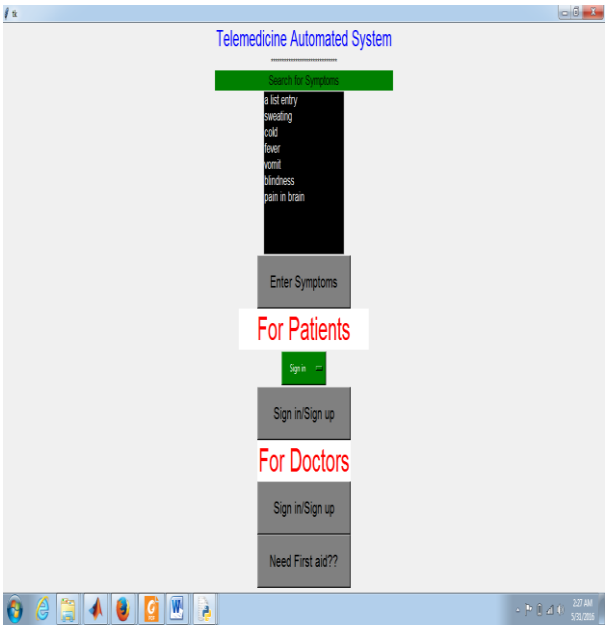
Procedural steps for appointment module are as follows:-

1. The patient using the patient agent provides the request for appointment, which invokes the Appointment agent.
2. The appointment agent interface with the patient agent and main agent, with its main function being to give confirmed appointment information to the appointment requesters.
3. The main agent is playing very important role that interface with appointment agent doctor agent with the database.
4. The purpose of core in main agent to cross examines the request of appointment against doctor schedule to provide accessible appointment slots.
5. The purpose of core in the doctor agent is to obtain doctor‘s schedule distantly and interface with the doctor‘s appointment database and the schedule agent.
6. The schedule agent interfaces with the main agent and gives proof of doctor‘s schedule and confirmed appointments

Database Structure

Below is mentioned the database tables which are used in the proposed scheme. Along with the tables, the attributes are also mentioned.

1. Doctors Data
2. Patient’s data
3. Doctor-Expertise
4. Disease-Expertise
5. Disease-Test
6. Disease-Symptoms
7. Doctor-password
8. Doctor-Patient
9. Patient-Test
10. Doctor Schedule
11. Patient Prescription



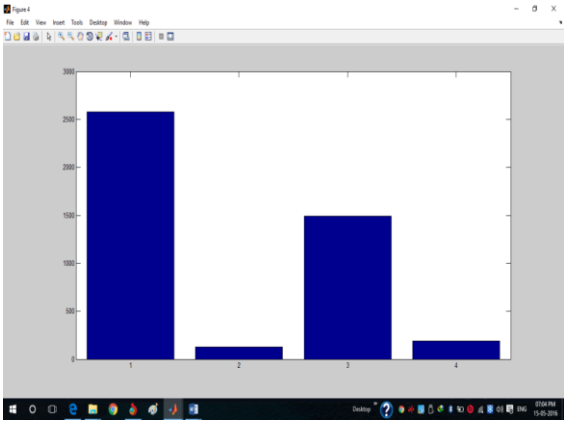


Figure 3. Showing Confusion Matrix

Figure 3 explains the values of the TP, FP, TN and FN obtained from our prediction algorithm i.e. hybrid Density based K-means aided SVM. As observed the TP and TN values are quite more as compared to the False values. This establishes our efficacy of the algorithm.

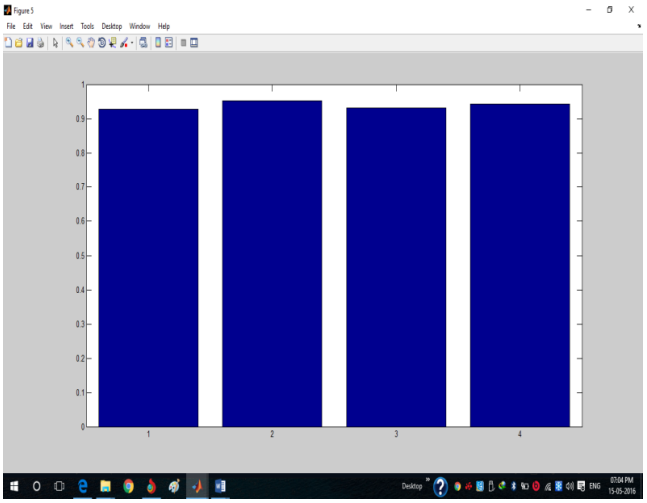


Figure 4. Showing Accuracy, Precision, Recall and F-Score

Figure 4 shows the various performance parameters obtained from our hybrid algorithm to test its efficacy. The values are calculated using Confusion matrix. As shown the accuracy and precision are quite high.

V. CONCLUSION

Our aim is test case prioritization of test cases such that the testing effort reduces significantly by maintaining code coverage remains more or less same. This is attained by using clustering approach such that the test cases are selected from each cluster thereby ensuring uniform distribution of code coverage. We proposed technique for test case prioritization using clustering approach. We used innovative density based K-means clustering for our analysis.

We apply K-means clustering algorithm and improve it on the bases of density i.e. Density Based K-means clustering algorithm. Then, apply the dataset on DB K-means algorithm to make the clusters which is generated on the bases of code coverage information.

VI. REFERENCES

[1]. Rothermel, Gregg, Roland H. Untch, Chengyun Chu, and Mary Jean Harrold. "Test case prioritization: An empirical study." In Software Maintenance, 1999. (ICSM'99) Proceedings. IEEE International Conference on, pp. 179-188. IEEE, 1999.

[2]. Rothermel, Gregg, Roland H. Untch, Chengyun Chu, and Mary Jean Harrold. "Prioritizing test cases for regression testing." Software Engineering, IEEE Transactions on 27, no. 10 (2001): 929-948.

[3]. Elbaum, Sebastian, Alexey G. Malishevsky, and Gregg Rothermel. "Test case prioritization: A family of empirical studies." Software Engineering, IEEE Transactions on 28, no. 2 (2002): 159-182.

[4]. Carlson, Ryan, Hyunsook Do, and Anne Denton. "A clustering approach to improving test case prioritization: An industrial case study." In Software Maintenance (ICSM), 2011 27th IEEE International Conference on, pp. 382-391. IEEE, 2011.

[5]. Arafeen, MdJunaid, and Hyunsook Do. "Test case prioritization using requirements-based clustering." In Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on, pp. 312-321. IEEE, 2013.

[6]. Kayes, MdImrul. "Test case prioritization for regression testing based on fault dependency." In Electronics Computer Technology (ICECT),

2011 3rd International Conference on, vol. 5, pp. 48-52. IEEE, 2011.

[7]. Jacob, ThangavelPrem, and ThavasiAnandam Ravi. "A NOVEL APPROACH FOR TEST SUITE PRIORITIZATION." Journal of Computer Science 10, no. 1 (2013): 138.

[8]. Di Nardo, Daniel, Nadia Alshahwan, Lionel Briand, and YvanLabiche. "Coverage-based test case prioritisation: An industrial case study."In Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on, pp. 302-311. IEEE, 2013.

[9]. Ray, Mitrabinda, and Durga Prasad Mohapatra. "Multi-objective test prioritization via a genetic algorithm." Innovations in Systems and Software Engineering 10, no. 4 (2014): 261-270.

[10]. Kaur, Arvinder, and ShubhraGoyal. "A genetic algorithm for regression test case prioritization using code coverage." International journal on computer science and engineering 3, no. 5 (2011): 1839-1847.

[11]. Muthusamy, Thillaikarasi, and K. Seetharaman. "EFFECTIVENESS OF TEST CASE PRIORITIZATION TECHNIQUES BASED ON REGRESSION TESTING." International Journal of Software Engineering & Applications 5, no. 6 (2014): 113.

[12]. Beena, R., and S. Sarala. "Code Coverage Based Test Case Selection and Prioritization." arXiv preprint arXiv:1312.2083 (2013).

[13]. Bhatia, M. P. S., and DeepikaKhurana. "Experimental study of Data clustering using k-Means and modified algorithms." International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol 3 (2013).

[14]. D.R MedhunHashini. “Clustering Approach to Test Case Prioritization Using Code Coverage Metric.” International Journal Of Engineering And Computer Science, no. 4 (2014):5304-5306.