A Novel Density based K-means Clustering for Test Case Prioritization in Regression Testing Result-II

Dr. Amit Verma, Rohit Bajaj, Ishadeep Kaur Luthra

1,2,3Dept. of CSE, Chandigarh Engineering College, Landran (Mohali), India

Abstract
In this paper, we work on improving the test case prioritization on the basis of clustering approach. A novel density based k-means clustering approach is used to make clusters of different test cases on the basis of statement coverage. Then, prim’s algorithm is used to find out the minimum path between different test cases according to their coverage information. Test cases are select from every cluster; which have maximum coverage information. According to Prim’s algorithm, we will find the tree of test cases; this technique reduces the test cases numbers. Only those test cases are selected which have maximum coverage information. It will reduce the effort, cost and time also.

Keywords
Test Case Prioritization, Density based Kmeans, Regression Testing

I. Introduction
Software development has come into an era of 40ies. Since then testing has become a major part of the software development. In this competitive world providing the best software with latest technology has become the need of software engineers. To achieve this goal, engineers try to develop new software which is more users friendly. Before implementing this software it goes through number of tests. Sometimes the software does not give the desired result; in that case some changes are required into the developing code. Then the repeated test performed on the updated code is the Regression Testing. The various techniques used in Regression Testing are:

- Retest All
- Regression Test Section
- Prioritization of Test Cases

The first case is very time consuming process and cost inefficient as the tester has to perform the test cases repeatedly. In case of selecting the test section, developers has to be careful enough that it pick up only relevant test cases and should not waste time on irrelevant test cases. For example: The test cases in which changes have been made and should not pick the test cases which are already being tested. To solve the problem of test section, Prioritization of test cases need to be done. To prioritize the test cases many algorithms have been produced till date such as Greedy view, Particle Swarm Optimization, Spanning Tree algorithm and many more.

Clustering algorithm has been produced to prioritize the test cases, but we have used the concept of novel density based k-means clustering to make clusters of different test cases on the basis of statement coverage. Then, prim’s algorithm is used to find out the minimum path between different test cases according to their coverage information. Test cases are select from every cluster; which have maximum coverage information. According to Prim’s algorithm, we will find the tree of test cases; this technique reduces the test cases numbers. Only those test cases are selected which have maximum coverage information. It will reduce the effort, cost and time also.

II. Problem Formulation
A The problem of test case prioritization has gained significant attention over the last few years as software testing forms a major section of the whole software development process. The cost of software development is directly dependent on the testing effort. This thesis aims to reduce this cost by prioritizing test cases and running the tests for the selective test cases as per the available time and manpower. There are a number of test cases available which can consume a lot of time and effort. A selective number of test cases need to be selected which would be otherwise used for the same purpose. The priority of the test cases need to be decided on the basis of several parameters. The parameter for the test case prioritization needs to be chosen and a model needs to be developed which would set priority among the test cases. First of all a data set needs to be generated which would be utilized for our proposed algorithm testing. Then the dataset needs to be preprocessed for outlier removal and redundancy removal. Then a technique for clustering of the test cases needs to be developed which would be utilized for the above mentioned problem.

III. Methodologies
Steps of Density Based K- Means Algorithm:

Let X = {x1, x2, x3... xn} be the set of data points in Dataset D, Euclidean “ε” (eps).
K is the number of clusters to be found, minPts is a minimum number of neighbors required in neighborhood to form a cluster and N is a set of points in ε neighborhood.

Step 1: Start with arbitrary starting point (not visited).

Step 2: Find all the neighbor points of starting points using Euclidean distance “eps”.

Step 3: If number of neighbors ‘N’ is greater than or equal to minPts, then starting point and its neighbors are added to cluster. Starting point is marked as visited; otherwise this point is marked as noise.

Step 4: A new unvisited point is recall and processed for further make a part of cluster or noise.

Step 5: Repeat Step 2, until all points are marked as visited.

Step 6: We have ‘m’ cluster for each detected clusters; then find cluster centers, ‘Cm’ by taking the mean find the total number of points in each clusters.

Step 7: If ‘m’ cluster is greater than ‘K’ clusters, then join two or more cluster based on density and no. of points to find the new cluster center. Repeat Step 7, until achieving K clusters with ‘Ck’ centers.

Step 8: Otherwise if ‘m’ is greater than or equal to the number of clusters initially found by density based clustering algorithm ‘l’; select a cluster based on density and number of points split it using k-means clustering algorithm. Repeat Step 8, until achieving k clusters with ‘Ck’ centers.

Step 9: Apply iteration of k-mean clustering with k and new ‘Ck’ centers as the initial parameters and label all the clusters with k labels.

Step 10: End
Flow chart of Density Based K-means Clustering Algorithm:

Start

Calculate ‘e’ of all neighbor points from starting point

If N=minPts

Marked starting point as visited

Yes

Marked starting point as visited and add this into cluster

Marked starting point as visited and add this into cluster

No object more

‘m’ cluster is formed

If m>=k

Join two or more clusters based on density

No

Select a cluster based on density

New cluster is formed by applying iteration for K-means

End

IV. Results

Fig. 1: Flow chart of Density Based K-means Clustering Algorithm

Fig. 2: Clusters Obtained from First Iteration

Fig. 3: Clusters Obtained from Second Iteration

Fig. 4: Clusters Obtained after Number of Iterations

Fig. 5: Representing the Total and Sub-List of Test Cases

Fig. 6: Actual Vs Prioritized Test Case Numbers
VI. Conclusion and Future Scope

A novel Density based K-means clustering has been applied for software testing. Test Case prioritization has been done using a hybrid algorithm of minimum spanning tree and density based k-means clustering. The results have been found to be quite better than their traditional counterparts. In future other algorithms can be implemented for the same and compared for performance.

References


