Density Based Clustering Approach for solving the Software Component Restructuring

Shreya Jaiswal[1], Deepak Xaxa[2]

[1]*Computer Science and Engineering, Mats University, Raipur, C.G.*
[2]*Assistant Professor, Computer Science and Engineering Mats University, Raipur, C.G.*

---***---

ABSTRACT- *Software components restructuring is quiet difficult technique for software maintenance and development. Different Clustering techniques have been used to solve this problem. Here in this paper the Density- based spatial clustering of applications with noise (DBSCAN) is used with single linkage method to solve software complexity and to group related software components. DBSCAN forms clusters by grouping points that are closely packed together i.e. points with many nearby neighbors, marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. DBSCAN reduces the time complexity by finding the clusters with the shortest distance and makes feasible for huge data. This paper presents the framework for DBSCAN shown by flow charts which specifies the similarity measures between the two clusters. The experimental results is shown in comparison with fuzzy clustering. DBSCAN achieve a consistent and compatible clustering quality.*

Keywords: Software Component Clustering, DBSCAN, Hierarchical clustering.

1. INTRODUCTION

Analyzing software components is very difficult method and also the maintenance of software and its development too becomes difficult . Every day a evolution of new of software and software development costs are the reason of degrading quality of software. Access new s/w and understanding its structure is becoming uneasy. Due to this free use of resources cannot be accesed. Agents always tries to “steal” resources from remote host .Data stripping or alteration also becomes difficult. Cohesion and coupling always helps the s/w system to maintain the quality of s/w and also make it easy to understand and access.

Cohesion and coupling also enables the program of new s/w to initigate and the main task is to help the component in partitioning. Different algorithms of clustering are likely to group the similar type of components based on similarity function. Clustering of software is boon for software developers as its help for the various purposes such as restructuring of program , recovery of design partitioning of software and most important it makes software very easy to understand numbers if data points are strongly recommended and accepted in software system as easy data carries very unique aspect .There is significant research carried out for designing new similarity measures which can help to find the similar component of two software. The distribution of component has important contribution in evaluation the degree of similar between the software also the different factors which may help coupling along with functional and various nonfunctional requirements and reasons for legacy clustering of s/w also depend on single method such as distance calculations. It is not easier to find the different coupling relation along s/w components basically research on s/w component clustering do not cover other area such as biometrics, pattern recognition etc because here distance calculation and measurements are not possible . DBSCAN technique shows various measurement between two clusters .Also it runs fast large data achieving a consistent and compatible clustering quality. Another challenge in s/w clustering techniques is that few data cannot be classified for high coupled component where as it is having high membership for more than only cluster value. This problem is solved in density based clustering. Therefore the main objective of paper is to first design an efficient similarity measure which essentially considers the distribution of the feature over the complete input .Density based clustering will carry out the analysis for the various situation for s/w components as DB scan algorithm helps in detecting nonlinear shapes structure which are bases on the density .Here this clustering technique is used because it reduces time completely it helps in reusability clustering of code , it completely works on similarity function the main object of the research paper is to focus on forming clustering components based on high cohesion and low coupling DB scan can identify noise data while clustering and can also be able to find arbitrarily size and arbitrarily shaped clusters as it does not require a prior specification of no of clusters.

2. PROBLEM IDENTIFICATION

In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) Algorithm (Bezdek 1981). The FCM algorithm attempts to partition a finite collection of n elements X = {x1,...,xn} into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centres C = {c1,...,cc} and a partition matrix

J= U = u i, j ϵ [0,1], i = 1,….,n, j = 1,…,c

where each element uij tells the degree to which element xi belongs to cluster cj . Like the k-means algorithm, the FCM aims to minimize an objective function. The standard function is: which differs from the k-means objective function by the addition of the membership values uij and the fuzzifier m. The fuzzifier m determines the level of cluster fuzziness. A large m results in smaller memberships uij and hence, fuzzier clusters. In the limit m = 1, the memberships uij converge to 0 or 1, which implies a crisp partitioning. In the absence of experimentation or domain knowledge, m is commonly set to 2. In fuzzy clustering, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. Any point x has a set of coefficients giving the degree of being in the kth cluster wk(x). With fuzzy c-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

Ck =∑x wk (x)x⁄∑x wk (x)

The degree of belonging, wk(x), is related inversely to the distance from x to the cluster center as calculated on the previous pass. It also depends on a parameter m that controls how much weight is given to the closest centre [14].

3. METHEDOLOGY

The main goal of program restructuring is to upgrade the internal structure and strength of software function. The program restructuring is done by the set of tools proposed in this paper. The main criteria is based on clustering applied with the help of cohesion and coupling. The original structure of program with limited measures is shown in fig.1. This approach gives knowledge about the present structure of the function, heuristic guidelines and quantitative measure of structure, clustering analysis with the help of existing code helps to restructure old program into new one. This approach has four phases:

(i)Data Collection And Processing: Data collection is the process of gathering and measuring information on targeted variables in an established systematic fashion, which then enables one to answer relevant questions and evaluate outcomes. Source code is collection of computer instructions, possibly with comments, written using a human-readable programming language, usually as ordinary text. The source code is sent to parsing tool where it automatically parses the data. Parsing is the problem of transforming a linear sequence of characters into a syntax tree. Refining is a process that refines disparate data within a common context to increase the awareness and understanding of the data, remove data variability and redundancy, and develop an integrated data resource. Entities are merged together depending on the attributes which are shared. The more attributes are common in two entities the more similar the two entities are to each other.

(ii) Phase 2 is clustering. The clustering technique used in our work is DBSCAN which forms cluster by grouping points that are closely packed together i.e. points with many nearby neighbors, marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN works on two parameters: ε (eps) and the minimum number of points required to form a dense region (minPts). If a point is found to be a dense part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, as is their own ε-neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise. The clustering tool constructs cluster automatically. Clustering Tool used is MATLAB.

(iii) The next Phase is visualization and analysis. After the most important clustering phase, the result is shown and produced by a graph. The graph clearly represent the data containing sparsity is clustered in one cluster and data which are noise free is clustered in one cluster. Two clusters are formed by the clustering tool.

(iv) Phase 4 is final restructuring of program. Software restructuring is a form of perfective maintenance that modifies the structure of a program's source code. Changes to the structure are introduced through the application of transformations. Manually transforming the source code may introduce undesirable as well as undetectable changes in the system's behavior.

4. RESULT

This section describes the experiments we have performed and the result obtained using Fuzzy clustering and DBSCAN. The result comparison between these two clustering techniques is been presented.

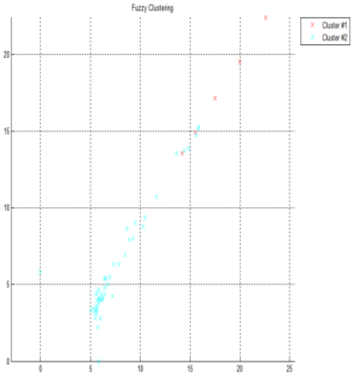

Fig.1 Fuzzy Clustering

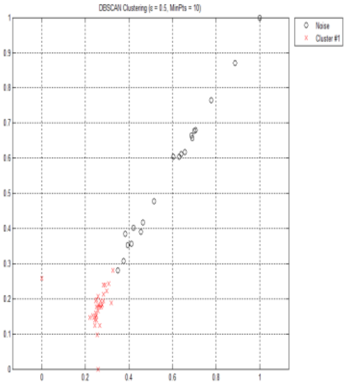

Fig.2 DBSCAN

Both the graphs shows the distribution of input on the clusters formed by the fuzzy clustering and DBSCAN. The proportion of input is far better by our clustering technique then the existing clustering technique.

5. CONCLUSION AND FUTURE SCOPE

Fuzzy Clustering forms two clusters in which proportion of data shared in the two clusters are not balanced. Cluster-1 has only 6 programs which is only 10.1695% of the total input where as Cluster-2 has 53 programs which is 89.8905% of the total input. The distribution is not proportionate. DBSCAN clustering also forms two clusters. The two clusters have evenly distribution of data. Cluster-1 so formed has 21 programs which is 35.5832% of the input. Cluster-2 contains 38 programs which is 64.4068% of input.

Another drawback of fuzzy clustering is that we have to pass the explicit cluster size where as in our proposed technique it is not required.

Hence, we conclude that program restructuring can be done by DBSCAN clustering technique as it is best suited in software development, maintenance and reliability causing evenly distributed data on both clusters formed by DBSCAN which will result in a better clustering technique and will improve the performance by reducing cost, low time and reduced effort

REFERENCES

[1] Sommerville. “Software Engineering”. 5th ed. Addison – Wesely. England. 1996.
[2] Chikofsky. E.J.. Cross. “Reverse Engineering and Design Recovery: A taxanomy”. IEEE Software..1990.
[3] Fowler.M.. “Refactoring: Improving the design of existing code”. Addison-Wesely.1999.
[4] Briand. L..Morasca. S.. Basili. “Property Based software engineering measurement. IEEE Trans. Software Engineering. 1996.
[5] Munson. C.J.. “ Software engineering measurements”. Aurebach Publications. ACRC Press Company. 2003.
[6] Pressman. R.S.. “Software Engineering : A Practitioner’s Approach”. 4th edition McGraww-Hill.Inc. 1997.
[7] Wiggerets. T.A.. “Using Clustering Algorithms In Legacy Systems Modularization” Fourth Working Conference On Reverse Engineering. 1997.
[8] Arnold. R.S. “ Software Restructuring”. Proc. IEEE. 1989.
[9] Everitt. B. “Cluster Analysis”. Heinemann Educational Books. London.
[10] Romesburg. H.C. “Cluster Analysis for Researchers”. Krieger Publishing Company. Malbar.FL. 1990.
[11] Sneath. P.H.A. Sokal.R.R. “Numerical Taxanomy: The Principles and practice of Numerical Classification”. W.H. Freeman and Company, San Francisco. 1973.

[12] Duo Liu. Chung-Horng Lung. Samuel A. Ajila, "Adaptive Clustering Techniques For Software Components and Architecture". IEEE, 39th Annual International Computers, Software & Applications Conference.2015.

[13] Chung-Horng Lung. Xia Xu. Marzia Zaman and Anand Srinivasan. "Program Restructuring Using Clustering Techniques". Journal of systems and software. 2006.

[14] Mrs. Bharati R.Jipkate, Dr. Mrs.V.V.Gohokar /International Journal Of Computational Engineering Research / ISSN: 2250–3005.