International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

AGGLOMERATIVE HIERARCHICAL CLUSTERING TECHNIQUE FOR SOFTWARE COMPONENT RESTRUCTURING

Shivani Sahu[1], Shipra Rathore[2]

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

ABSTRACT- *Analysis of software components is quiet difficult technique for software maintenance and development. Clustering technique have been used to solve this problem. Here in this paper the fundamental agglomerative hierarchical clustering is used with single linkage method to solve software complexity and to group related software components. This algorithm first connects similar pair of clusters so that the distance between the similar cluster member is shortest and this process goes on until only one cluster is left. The agglomerative clustering algorithm reduces the time complexity by finding the clusters with the shortest distance and makes feasible for huge data. This paper presents the framework for agglomerative hierarchical clustering shown by flow charts which specifies the similarity measures between the two clusters. Also two important methods are obtained from this framework known as hierarchical star algorithm and hierarchical compact algorithm. The experimental results show that it runs fasts for large data achieving a consistant and compatible clustering quality.*

Keywords- Software Component Clustering, Density Based Clustering, Hierarchical Clustering, Fuzzy C-Means Clustering, Agglomerative Hierarchical Clustering.

1.INTRODUCTION

Now a days software is evolving due to the change in technique and need of software user, this is the reason for rising of the software development costs [1] with the evolution of new software , it becomes very complicated for the user to access it and its structure gradually degrades, lacks in quality as before. The high level structure is software architecture of software system, which is very difficult to understand the new software system. Cohesion and coupling helps the software system to maintain its quality and easier to understand.

Cohesion and coupling helps to mitigate the problems regarding the new software evolution with the help of component partitioning [2]. Various clustering algorithms are mostly used to group the similar components on the basis of similarity function.

Software clustering is also used for various purposes, e.g.: design recovery, program restructuring, easier understandability, software partitioning etc. Number if data points are strongly recommended and accepted in software system as each data points carry unique aspect. Along with this, different factors also helps coupling in software components, with functional and non-functional requirements and legacy reasons. Other than this, software clustering depends on single method e.g.: distance calculations [3], which is not easy to detect the difficult coupling relations with software components. But the research on software clustering do not cover other areas such as pattern recognition, where more then one measurement are used with distance calculations.

Another challenge in software clustering technique is that, some data cannot be classified for high coupled components, where instances have high membership value for more than one cluster. This problem has been solved in possibilistic clustering [4], but is not applied in software component analysis. This paper throws light on the work of different aspects affecting software cohesion and coupling.

Therefore the main objective of this paper is to enhance effectiveness of software clustering by improving the membership value calculation. The main goal is to adapt the distance based membership value calculation which is applied Irjet template sample paragraph Irjet template sample paragraph.

Irjet template sample paragraph, Irjet template sample paragraph .Irjet template sample paragraph. Irjet template sample paragraph

in agglomerative hierarchical clustering when any of the distance increases by predefined threshold then the new membership value will be updated in the clustering process which shows faster determination of the clusters.

2.CLUSTERING TECHNIQUE

Clustering is commonly used for analysis of data as it classifies different clusters into several groups on the basis of their similarity function. Along with data analysis, clustering has various use in dividing into two classes: hierarchical clustering and non-hierarchical clustering. Hierarchical clustering is further divided into three classes: agglomerative hierarchical clustering, divisive hierarchical clustering, incremental hierarchical clustering. Various hierarchical clustering have been given for ex: complete link, average link and Bi-secting k-means [5].

The agglomerative hierarchical clustering starts from one point cluster and keeps on adding most similar pair of clusters. The divisive hierarchical clustering starts with one cluster of all points and keeps on dividing most useful clusters. While in the incremental hierarchical clustering it models the hierarchy in an on-line form and it reduces the frequency of data scan.

One of the well known method of agglomerative hierarchical clustering is single linkage method. In this method two similar clusters are pair together and the similar member of cluster should have the shortest distance, in other words it can also be explained as, distance between two nearest point depend upon distance between two clusters. The most important task done by agglomerative cluster is that it search two similar pair of clusters whose distance is nearest and then merge those pair of cluster into new single cluster. Now the distance between new cluster and the old clusters are recorded and updated. Then again the same agglomeration process is repeated until only one cluster is left. This technique generates such type of clusters so that each members of clusters is more closely related to one of the member of same cluster. This method helps to accumulate chain of points into a one single cluster.

This algorithm have an additional approach for various application domain as they provide view of data at various abstraction level which helps users to visualize and interact with large data. In this technique, pair of cluster are merged when new cluster are added at each level producing large hierarchies [7] gave an algorithm for hierarchical cluster [6] based on multilayered clustering for producing hierarchy. With the help of multi-layered clustering, we propose a framework of agglomerative hierarchical clustering algorithm which can be shown by graph, it not only form cluster on similarity measure but can also handle mixed numerical and attributes and can also get disjoint or overlapped hierarchies.

3.METHODOLOGY

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

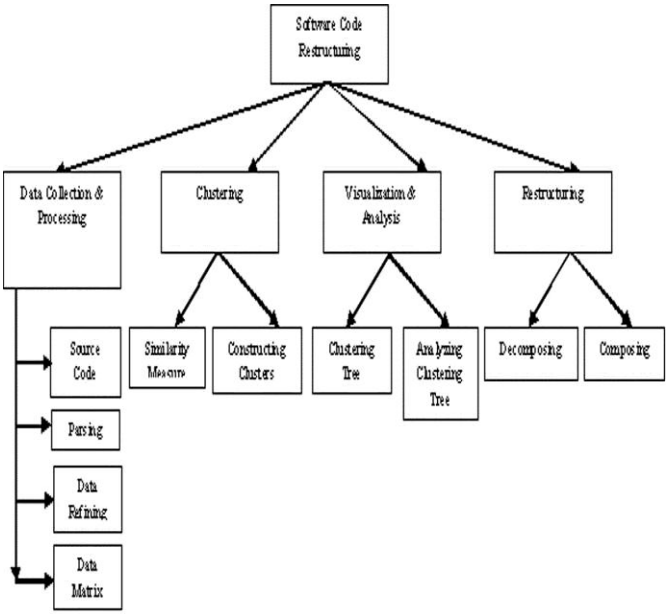

Fig 1: Approach to program restructuring

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 [13].

Fig.1 shows that using clustering techniques how program is restructured. This approach resolves various fundamental issues which are challenging for the user and software developer. Some issues like attributes inside of function, use of algorithm for calculating similar coefficients and also selecting of suitable clustering technique. This approach has four phases. Phase one is data processing and data collection. In this phase at the first phase the source code is sent to parses tool for parsing automatically. After this the parses generates the raw data of entity. But some of the raw data contains noises i.e. unwanted data, which can be eliminated during refining of data. Data refining is the most fundamental step as all the unwanted data which degrades the quality of program and reduced and eliminated. 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. Before applying clustering technique entities and attributes are defined then matrix is generated. After data refining is done, it is ready and prepared for the second phase i.e. Clustering.

Phase 2 is clustering. The most essential and fundamental step in clustering analysis is to find out similarity between two entities based on similarity measures. A matrix known as similar coefficient is calculated to find out similarity between two entities. This calculation

can only be done after defining entities and attributes. This calculation can only be done on the basis of resemblance coefficient of two entities. Many clustering algorithms have been given to coefficients for various function[]. There are many artificial factors which software consists for which algorithms are not suitable. After defining resemblance coefficient clusters are formed using agglomerative hierarchical clustering algorithm. The clustering tool performs this phase automatically.

Phase 3 is visualization and analysis. After the most important clustering phase, the result is shown and produced and shown in a tree which also shows existing structure of program function. Similar entities are grouped together to forms cluster. This similarity between clusters is represented by resemblance coefficient. By deeply studying the tree ill structured code can be detected which becomes the eligible candidates for restructuring. The clustering tree helps to restructure a function. But the main work of restructuring objective is based on the decision of software designer's experience, insights, although the tool can automatically generate clusters if sufficient clusters are provided. Phase4 is final restructuring of program. Those functions which will be identified as low cohesive will be decomposed into various code fragments and few of them will be composed into new functions. This is not executed automatically, as it needs manual processing.

4.RESULT

For the performance comparison we also stimulated fuzzy c-means algorithm in the entity attribute and distance matrix where as in agglomerative hierarchical clustering we converted entity attributes into relational matrix which is of n x n i.e. (10 x 10 in our case) and this would be feed as input for agglomerative hierarchical clustering and further the clustering are formed with maximum size of 3 clusters.

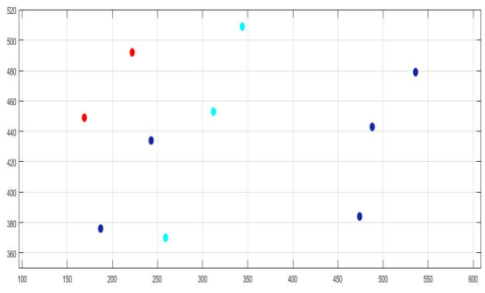

Fig-2: Graph showing clustering by agglomerative hierarchical

In the above graph there are ten plots of different colours in each graph. Each plot represent different programs which we have used in our clustering i.e. we have used ten different programs for applying clustering algorithms. In fig 2, three programs are forming cluster1, five programs are forming cluster 2 and two programs are forming cluster 3. After applying the fuzzy c-means clustering technique on the existing code we get only two clusters although the requirement was of three cluster groups. On applying hierarchical agglomerative clustering all the three clusters are formed giving equal weights to all data and maintaining the structure of program.

5. CONCLUSION AND FUTURE SCOPE

Fuzzy C-means forms only two clusters due to noise (unwanted data) while agglomerative hierarchical clustering forms three clusters. Hence, we conclude that program restructuring can be done by agglomerative hierarchical clustering technique as it is best suited in software development, maintenance and reliability causing low cost, low time and reduced effort.

Another research direction is to use different clustering algorithms like density based clustering, divisive approach of hierarchical. Instead of using term frequency one can also use inverse document frequency (IDF).

REFERENCES

[1] Sommerville, I., Software Engineering, fifth ed. Addison-Wesley, 1996, England.

[2] Lung, C., et al., "Program restructuring using clustering techniques," Journal of Systems and Software, 79 (2006), pp. 1261-279.

[3] Chong, C. Y., Lee, S. P. Ling, T. C. "Efficient Software Clustering Technique Using an Adaptive and Preventive Dendrogram Cutting Approach," Information and Software Tech., 55 (2013), pp. 1994-2012.

[4] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly Detection: A Survey," ACM Computing Surveys, 2009, pp. 1-72.

[5] Braden, R., et Al., "Resource ReSerVation Protocol (RSVP)," RFC 2205, 1997.

[6] Shtern, M., and Tzerpos, V., "Clustering Methodologies for Software Engineering," Advances in Software Engineering, Vol. 2012, Article ID 792024, pp. 1-18.

[7] Shtern, M., and Tzerpos, V., "Methods for Selecting and Improving Software Clustering Algorithms," J. of SPE, 2014, pp. 33-46.

[8] Maqbool, O., Babri, H.A., "The weighted combined algorithm: a linkage algorithm for software clustering," Proc. of the 8th Euromicro Working Conf. on Software Maintenance and Reengineering, 2004, pp. 15-24.

[9] Yassin, W., et al., "Anomaly-based Intrusion Detection Through Kmeans Clustering and Naives Bayes Classification," Proc. of ICOCI2013, pp. 298-303.

[10] Ensafi, R., Dehghanzadeh, S., and Akbarzadeh, M., "Optimizing Fuzzy K-means for network anomaly detection using PSO," Proc. of Int'l Conf. on Computer Systems and Applications, 2008, pp. 686-693

[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] Matrika Sinha. Shreya Jain. "Program Restructuring And Component Reuse With Data Mining Techniques". International Journal of Engineering Research and General Science Volume-4, Issue-3. 2016.

[15] Ronaldo C. Veras and Silvio R.L. Meria. Adriano L.I. Oliveiria and Bruno J.M. Melo. " Comparitive Study of Clustering Techniques for the Organisation of Software Repositories". 19th IEEE International Conference on Tools with Artifical