

Improve Reusability of Software Using Clustering Techniques

Amit Kumar¹
Teaching Associate,
GJUS&T, Hisar ,Haryana
malikamit7@gmail.com

Balkar Singh²
Teaching Associate,
GJUS&T, Hisar ,Haryana

Sandeep³
Teaching Associate,
GJUS&T, Hisar haryana

Abstract

Software module clustering is a problem of organizing the software system into modules to improve the program structure. Cluster analysis is a system used for cataloging of data in which data elements are screened into groups called clusters that represent collections of data elements that are based on a distance or dissimilarity. The cluster analysis approach is an important tool in decision making and an effective creativity technique in generating reusability of software modules. This research paper covers different types of methods such as Hierarchical Clustering and Non-Hierarchical Clustering techniques for software modules classification.

Keywords: Reusability, Clustering Techniques.

1. Introduction

In a knowledge engineering approach, the knowledge of human experts is described as a set of rules, which are then used in the process of classification. The disadvantage of this approach is that it requires a lot of efforts to make human knowledge explicit and for each new domain, again a separate formulation of the rule need to be done manually. In a machine learning approach, the classifier is built without human intrusion and classification for different domains can be learned using the same algorithm. The accuracy of all automatic cataloging system is highly in need upon the effort and care taken during the rules definition phase.

A cluster analysis acts a big job in software alliance. Cluster analysis is the proposal for sorting out data into clusters or groups in a situation where no prior information about a structure is vacant [17]. It divides data into groups (clusters) that are meaningful, useful or both [16]. The clustering approach is a key gadget in decision making and an effective inspiration method in generating ideas and obtaining solutions. The goal of a cluster analysis is that units within a cluster should be as similar as possible, and clusters should be as different as possible as. Cluster analysis is a tentative data analysis tool for solving classified problems. Its objective is to sort cases (people,

things, events, etc) into groups or clusters, so that the degree of friendship is strong between members of the same cluster and weak between members of different clusters [18]. Each cluster thus describes, in terms of the data composed, the class to which its members well in and this picture may be abstracted through the use of the particular to the general class or type. We have used Hierarchical Clustering Algorithm and Non-Hierarchical Clustering Algorithm in this research to implement our proposed method described in coming topics.

Cluster analysis is a course of identification and categorization of subsets of objects. Partitioning or clustering techniques are used in many areas for a wide spectrum of problems. Among the areas in which cluster analysis is used are graph theory, business area analysis, information architecture, information retrieval, resource allocation, image processing, software testing, chip design, pattern recognition, economics, statistics, biology and many more.

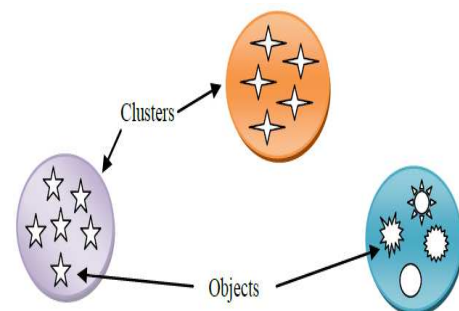


Figure 1. Similar objects in similar clusters and others in different.

2. Proposed Model

In a large component repository, software component retrieval is accomplished through some classification schemes [3]. The need of the software is increasing exponentially but the manpower is not increasing proportionately so we are in the situation of software

failure or crisis. The reusability [4] is the excellence of a portion of software that permits it to be used over again and again with or without a small change. Software professionals have indentified reuse as potent resources to potentially defeat the situation of software crisis [12] [13] [14]. The software module clustering problem consists of automatically finding a good quality clustering of software modules based on the relationships among the modules. These relationships typically take the form of dependencies between modules [15]. So we find relationship between modules with LOC and Code Clones. As discussed earlier, we have used the existing HC algorithm (Agglomerative method) [16] and NHC algorithm (K-Mean method) [17] in order to find out our research objectives.

For this we will follow our **Algorithm** as discussed below:

1. Search program modules from the system.
2. Find out LOCs and Clones from these modules.
3. Calculate Euclidian distance on the behalf of LOCs and Clones.
4. Find the no. of clusters to be made using Agglomerative method on the behalf of Euclidian distance.
5. Use K-Mean method to make clusters.
6. Calculate Efforts and Development time.
7. Compare Efforts and Development time after clustering process.
8. Graphical representation of results.

The figure number 2 will make clear the picture for you, so you would be able to understand this approach more easily.

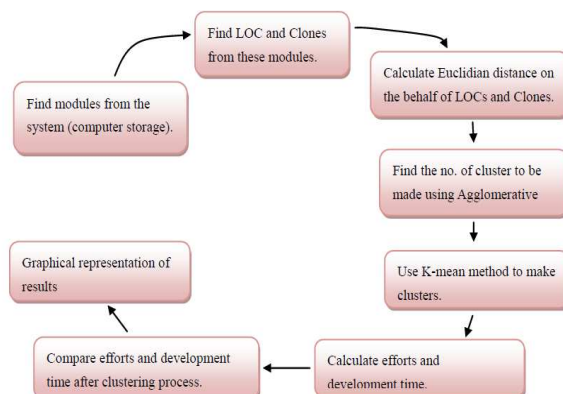


Figure 2. The proposed model given by ourself.

3. Results Discussion

We have experimented on 15 Java modules. After attaching the codes to AM Clone Scanner (developed in Java), we find the results in term of LOC of each module and clones between each module as shown in table 1. Using Agglomerative [3] method we find the schedule on the behalf of which we decide how many clusters we can make. This is shown in the table 2. You see there is a big jump in coefficient values after stage 6. So the number of cluster will be 9 because we deduct 6 from 15 i.e. 9, after applying the equation 1 given below:

$$\text{Number of Clusters} = N - S_i \quad (1)$$

- Where N is the total number of modules.
- Si is the case stage up to where the coefficient values difference is smaller to the next stages.

Table 1. LOC and Clones of each module

Stage	Cluster Combined		Coefficients
	Cluster 1	Cluster 2	
1	14	15	1.000
2	6	7	1.000
3	6	10	4.500
4	8	11	5.000
5	3	5	5.000
6	6	13	8.333
7	8	14	20.000
8	1	4	25.000
9	2	3	37.500
10	6	8	45.354
11	2	12	100.000
12	2	6	163.000
13	1	9	307.500

Table2. Agglomerative Schedule using average linkage met

Module	Line Of Code(LOC)	Clones
1	43	16
2	30	21
3	29	20
4	47	14
5	34	15
6	18	8
7	19	7
8	15	10
9	18	9
10	22	12
11	28	17
12	47	34
13	41	22
14	23	14
15	34	16

After attaching the Lines of Code of each module to COCOMO Basic tool (developed in Java) we find the different results before and after clustering approach. The comparisons of these results are shown with the help of figure 3, 4 and 5.
 Table 3 shows how the LOC and Clones are adjusted in total 8 clusters.

Table 3: Final Cluster Centre

	Cluster							
	1	2	3	4	5	6	7	8
LO C	43	30	29	47	34	18	19	15
Clone	16	21	20	14	15	8	7	10

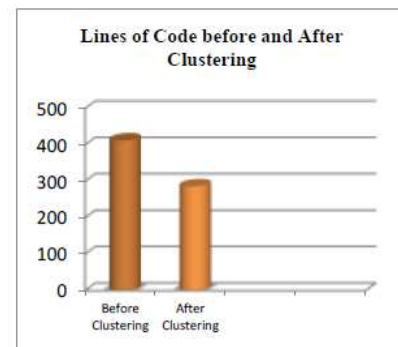


Figure3. Comparison of Lines of Code

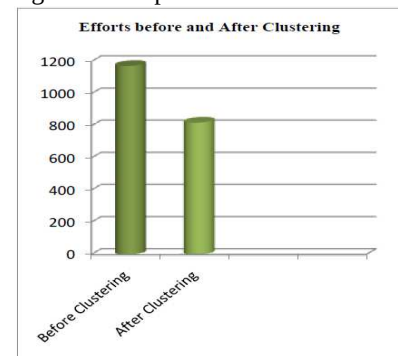


Figure4. Comparison of Development Efforts

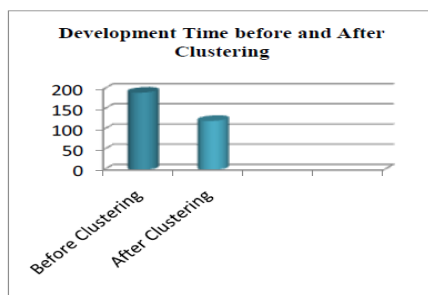


Figure 5. Comparison of Development Time.

4. Conclusion

A large numbers of software are available in all over the world and the need of software increasing day by day but the manpower is not increasing proportionately. So there is a big need to reuse those software modules in new projects. For this intention it is very crucial to organize those software modules in a meaningful structure, which can be done by cluster analysis. After applying clustering methods, we find the natural groups of software module where the degree of similarity between two software modules is maximal if they belong to the same group and minimal otherwise and we also find the maximum similar clones between software modules within similar cluster. We saw that after implementation of cluster analysis number of LOCs, efforts of development and development time are reduced. So this research is effective to overcome the problem of software crisis or failure. This experiment has done with 15 C++ modules only. This approach can be effective for a large dataset and automation of this approach will be a milestone in software engineering.

References

- [1] Manhas S., Sandhu P. S., Chopra V., Neeru N. (2010), "Identification of Reusable Software Modules in Function Oriented Software System using Neural Network Based Technique", World Academy of Science, Engineering and Technology, Vol. 67.
- [2] Shri A., Sandhu P. S., Gupta V., Anand S. (2010), "Prediction of Reusability of Objected Oriented Software System using Clustering Approach", World Academy of Science, Engineering and Technology, Vol. 67, PP. 853-856.
- [3] Sembiring R. W., Zain J. M., Embong A. (2010), "A Comparative Agglomerative Hierarchical Clustering Method to Cluster Implemented Course", Journal of Computing, Vol. 2, Issue 12, PP. 1-4.
- [4] Sonnum S., Thaithieng S., Ano S., Kusolchu K., Kerdprasop N. (2011), "Approximate Web Database Search Based on Euclidean Distance Measurement", Proceeding of the International MultiConference of Engineers and Computer Scientists, Vol. 1, PP. 16-18.
- [5] Cancino A. E. "Load Profiling of MERALCO Residential Electricity Consumers using Clustering", Manila Electric Company (MERALCO), Pasig City, Philippines.
- [6] Czibula I. G., Serban G. (2007), "Hierarchical Clustering for Software System Restructuring", Babes Bolyai University, Romania.
- [7] Basit H. A., Jarzabek S. (2007), "A Data Mining Approach for Detecting Higher-level Clones in Software", IEEE Transactions on Software Engineering, PP. 1-18.
- [8] Sandhu P. S., Singh H., Saini B. (2007), "A New Categorization Scheme of Reusable Software Components", International Journal of Computer Science and Network Security, Vol. 7, Issue 8, PP. 220-225.
- [9] Maqbool O., Babri H. A., "The Weighted Combined Algorithm: A Linkage Algorithm for Software Clustering", Lahore University of Management Sciences DHA Lahore, Pakistan.
- [10] Nakkrasae S., Sophatsathit P., "A Formal Approach for Specification and Classification of Software Components", University of Louisiana at Lafayette, U.S.A., Chulalongkorn University, Bangkok, Thailand.
- [11] Jalender B., Govardhan A., Premchand P. (2011), "Breaking the Boundaries for Software Components Reuse Technology", International Journal of Computer Applications, Vol. 13, Issue 6.
- [12] Goel H., Singh G. (2010), "Evaluation of Expectation Maximization based Clustering Approach for Reusability Prediction of Function based Software System", International Journal of Computer Applications, Vol. 8, Issue 13.
- [13] Jeng J. J., Cheng B. H. C. (1993), "Using Formal Methods to Construct a Software Component Library", Proc. of 4th Eur. Soft. Eng. Conf., Lect Notes in Comp. Science, PP. 397-417.
- [14] Jeng J. J., Cheng B. H. C. (1995), "Specification Matching for Software Reuse: A Foundation", ACM, PP. 97-105.

- [15] Mahdavi K., Harman M., Hierons R. M. (2003),
“A Multiple Hill Climbing Approach to Software
Module Clustering”, Proceedings of the
International Conference on Software
Maintenance, DISC Brunel University.
- [16] Fokaefs M., Tsantalis N., Chatzigeorgiou A.,
Sander J. (2009), “Decomposing Object-
Oriented Class Modules Using an Agglomerative
Clustering Technique”, IEEE, Proc. ICSM,
Canada.
- [17] Kanungo T., Mount D. M., Netanyahu N. S.,
Piatko C. D., Silverman Wu A. Y. (2002), “An
Efficient k-Means Clustering Algorithm:
Analysis and Implementation”, IEEE
Transactions on Pattern Analysis and Machine
Intelligence, Vol. 24, Issue 7.