Software Modules Clustering: An Effective Approach for Reusability

Jai Bhagwan[1*] Ashish Oberoi[2]

1. Department of Computer Science and Engineering, Maharishi Markandeshwar University, Mullana (Ambala) – 133 207, Haryana, India
2. Department of Computer Science and Engineering, Maharishi Markandeshwar University, Mullana (Ambala) – 133 207, Haryana, India

* E-mail of the corresponding author: jaitweet@gmail.com

Abstract

Software modules reusability may play an unbeatable role to increase the software productivity. Code clones can be used one of the parameter for cluster formation of software modules. Cluster analysis is a scheme 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 dissimilarities or similarities. The clustering approach is an important tool in decision making and an effective creativity technique in generating ideas and obtaining solutions. Software development and maintenance are big challenges in the market for survival of a software industry. This research gives an idea of reducing development time and efforts using clone detection and clustering process. Different types of methods have been applied in this research such as Hierarchical Clustering (HC) and Non-Hierarchical Clustering (NHC) for software modules classification. We have proved how this research is useful in software development and maintenance. The experiments have been done using 13 C++ programs.

Keywords: Lines of Code (LOC), Hierarchical Clustering Algorithm (HCA), Non-Hierarchical Clustering Algorithm (NHCA)

1. Introduction

Code clones are similar program segments of extensive size and logical similarity. Several techniques have been proposed to detect similar code fragments in software, so-called code clones [1]. One of them is string matching technique; on the basis of this string matching idea [1] we have developed a tool named JB Clone Scanner to detect clones in software modules corresponding to each other.

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 rules 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 [2]. 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 plays 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.

It divides data into groups (clusters) that are meaningful, useful or both. 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 [2] [10] [16] as shown in Figure 1. 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. 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 [2] [16] [18].

Moreover, we have used Hierarchical Clustering Algorithm (HCA) [8] [17] and Non-Hierarchical Clustering Algorithm (NHCA) i.e. K-mean [18] in this research to implement our proposed method which will be described in coming topic(s).

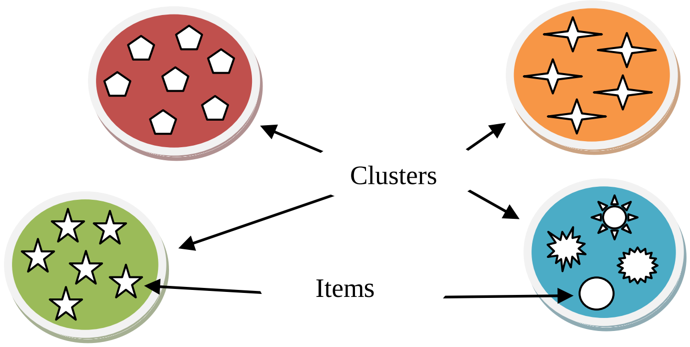

Figure 1. Similar and dissimilar items in clusters

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 [2]. Finally, software systems need to develop as business requirements, technology and environment change [10]. Software modules can be reused to reduce development time, efforts and development cost [11].

2. Related Work

In this section, the study of various papers and opinions of relative authors regarding the cluster analysis is projected which aid in defining the objective. A brief picture of papers analysis is given under this topic. Basit et al. in [1] focused that code clones are the software parts having similar or almost similar properties. Group of similar clones are known as structural clones or higher level clones. E.g. group of functions or classes. The author found higher level clones using clone miner tool in order to manage design components of the software. Authors say you can easily understand the design of a software component. This approach follows formulation of structural clone and the application of data mining techniques to find out the higher level similarity between clones. Clone miner finds simple clones first and then higher level or complex clones. The clone miner is written in C++ language. Authors suggest this research is useful in software maintenance.

Neeraj et al. in [2] proposed that the efforts and development time can be reduced by automatic categorization of software modules. In this research the authors have used Lines of Code and Number of Functions as software metrics in order to find out the number of clusters to be made using Agglomerative method. After that, they used K-mean method to formulate clusters and show how we can reduce efforts and development time. Authors have used SPSS tool for experiments in this research.

Manhas et al. in [3] discussed that the cost of software can be reduced by using software components already existing, by their proper reuse. In this research various types of metrics are used to make classification of software modules. Back propagation based neural networks are experimented to obtain results in terms of Accuracy, MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error values).

Shri et al. in [4] propose that hybrid k-mean and decision tree approach which are used in order to find out the reusability values of object oriented software components. The reusability values will be 'Nil' or 'Excellent' in order to make distinguish between the components of software modules. Metrics for recognizing the excellence of software components are predictable. Authors have proposed an algorithm in which the data input is specified to k-mean clustering in the form of tuned values of the Object Oriented software components. A decision tree system is developed in order to achieve similar type of software entities. The target of the metrics is to forecast the quality of software components. A range of attributes which are used in order to calculate the quality are maintainability, defect density, fault proneness, understandable, reusability etc. The proposed five metrics for OO Paradigms are as weight methods per class, depth of inheritance tree, number of children, coupling between object classes and lack of cohesion in methods. The implementation is done in the open source tool named Rapidminer 4.6. On the basis of metrics told above the result is calculated and shown by a decision tree using hybrid k-mean.

Sembiring et al. in [5] analysed that there are various clustering methods in order to mine the data. HCA is useful where clustering of variables is of agglomerative in nature. Hierarchical clustering have been used in this research for mining the of record of an higher level learning institute in order to find out which course is suitable for students according to industries needs. MATLAB and HCE 3.5 software is used to train data and cluster course applied during industrial training. This study shows different methods form different number of clusters. Industrial training also used to acquire data to evaluate the programme of the program. Continuous feedbacks from the industry and from the graduate students enable Higher Learning Institute to develop programme in order to suit industrial job needs. The methods like HCA and divisive hierarchical are used for data mining. The Agglomerative method forms the cluster according to the similar and dissimilar entities. The Dataset is taken from the parameter as questionnaires distributed during industrial training. In implementation 14 universities courses, 17 faculty courses, 9 study program courses and 3 elective courses are taken. The clustering is done on the basis of Euclidean distance and agglomerative methods.

Sonnum et al. in [6] discuss that there are various kinds of methods for data finding and searching which are successful. But these have one major lack that these are not capable to count objects having not sufficient matching properties. To overcome this problem authors use Euclidean distance method. In this paper the Euclidean distance method with the Erlang language is used to implement a Web-based search on mobile phone application. Euclidean distance metric is the familiar distance between two objects and is given by the Pythagorean formula. This research has been done using the web database which is used to sell mobile phones. The sorting or clustering is done on the basis of their parameters like mobile model, resolution, width, height and weight. The data is mined by considering these parameters, so that a customer can easily pick up a mobile by his/her choice of quality. This research proves that objects having similar properties are easily to find out with minimum time spending.

Cancino et al. proposed that [7] clustering has been done here to sort out customers on the basis of their electricity consumed load, so Manila Electric Company will be able to produce the electricity meter according to customers' needs. The information can be obtained on the basis of the use of the daily load profile. A case study is presented on the behalf of the actual load profile data sample residential customers of distribution utilities to determine the customer bracket that is affected by the existing time of use rates of the utility. The customer bracket was based on the energy consumption of all residential customers. Utilities have also been classifying its residential customers according to the KWh consumption use. This research was done to analyze the appropriate clustering methods based on pattern recognition in load profiling for distribution utilities in a context of deregulated power industry. K-mean, Improved K-mean, agglomerative and hybrid Agglomerative methods are compared with.

Czibula et al. [8] focused on the problem of decisive refactoring that can be used in order to improve the design of object oriented software systems. This paper aims at presenting a new hierarchical agglomerative clustering algorithm, *HARS* (Hierarchical agglomerative clustering algorithm for restructuring software systems). Clustering is used in order to overhaul the class structure of the system. *HARS* algorithm is used in order to obtain an improved structure of a software system, by identifying the needed refactoring.

Sandhu et al. [9] discussed that software industries pay their attention to software reusability. The systematic use of the software reuse approach improves the productivity and quality of the software. However there are issues which have been restraining the wide broaden use of software reuse. This is all about to the software components representation, its storage and rescue. Accepting and codifying the properties of software components is vital to the useful management and development of component-based software systems. This paper gives a new classification idea for reusable software component based on information retrieval theory. Different organizations of the extracted keywords that represent the semantic feature of the software component are evaluated. This approach allows using unwieldy terms and automatic cataloging of software components that are stored as a reusable component library. It could be beneficial for improving the productivity of reuse repository manager by easy identification and retrieval of desired software components. The experiment has been done in this research on 43 reusable software components which are taken from 'C' based open access repositories. A program is developed in MATLAB 7.2 to test the validity of the proposed characterization scheme and model. The new proposed model overcomes some of the limitations of the earlier models. The doublet and triplet word schemes make a better indexing unit as it also provides some contextual information as well. This is shown by the similarity function based evaluation of the new model for grouping of the Software Components. Also it shows that the use of the model will provide an effective characterization and indexing technique for storing out or grouping of

Software Components in Software Reuse Libraries. So, this approach can be used for automatic classification of software components and can be helpful in improving the productivity of the reuse repository managers.

3. Proposed Work

In a large component repository, software component retrieval is accomplished through some classification schemes [4]. 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 [14] [15]. The reusability 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] [15]. The software module clustering problem consists of automatically finding a good quality clustering of software modules based on the relationships among the modules [10]. 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) [17] and NHC algorithm (K-Mean method) [18] in order to find out our research objectives. To achieve objectives of our research, we propose the method as illustrated in Figure 2.

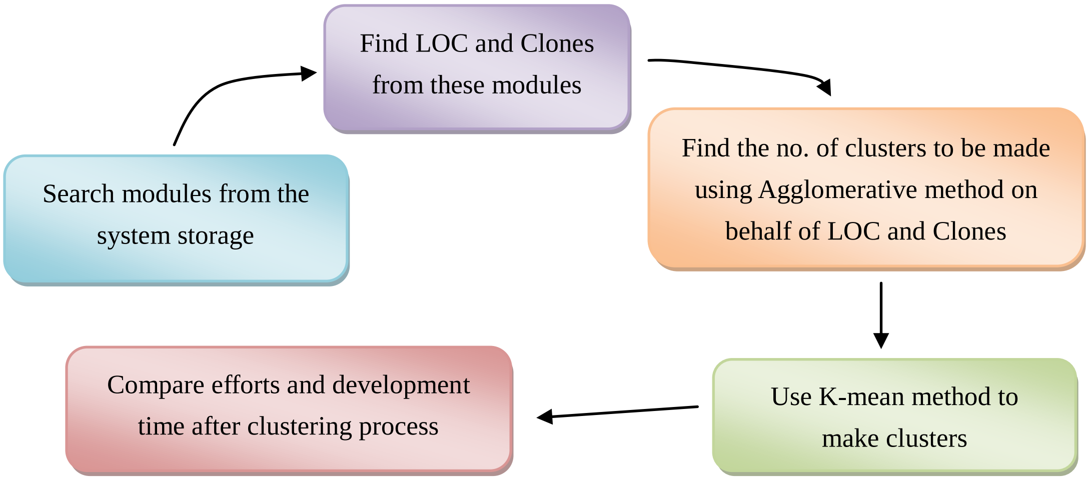
Find LOC and Clones from these modules
Search modules from the system storage
Find the no. of clusters to be made using Agglomerative method on behalf of LOC and Clones
Compare efforts and development time after clustering process
Use K-mean method to make clusters

Figure 2. The proposed approach given by us.

4. Implementation

As discussed earlier, in this research we have performed experiments on 13 C++ modules. According to previous work [2] the efforts and development time can be reduced using cluster analysis. So we have calculated LOC and Functions of these modules using existing approach which are shown in table 1. It is not valid that the number of functions of a module can make it consequently similar to other modules, so we observed that the code clones can make a better relation between software modules. On the behalf of LOC and Clones we can validate it better that how much a module is similar to other one. So we had a need to develop a tool named JB Clone Scanner which is based on a string matching inspiration as many researchers suggested a clone can be identified on the basis of a number of ways like string matching algorithms, using tokens, different kind of metrics etc [1]. After attaching these 13 programs to JB Clone Scanner (developed in C# by us), we find the results in term of LOC of each module and Clones between each module as shown in table 1 as well. Using Agglomerative [3] [17] method we find the schedule on the behalf of which we decide how many clusters are to be made. This is shown in the table 2. You can see, there are big jumps in coefficient values after stage 3rd in previous research and after 4th stage in our proposed work. So the number of cluster will be 10 and 9 in case of both previous one and our own work respectively by using the equation (i) given below:

$$\textit{Number of Clusters} = N - S_i \qquad \ldots\text{(i)}$$

- Where N is the total number of modules.
- Si is the case stage after which the coefficient values increase in large figure.

Table 1. LOC, Functions and Clones present in each module

Programs (Modules)	**LOC (Lines of Code)**	**No. of Functions (By Previous Research)**	**Clones (By our tool JB Clone Scanner)**
1	42	18	19
2	28	11	21
3	46	21	23
4	26	8	7
5	20	7	11
6	17	8	9
7	57	20	22
8	22	10	12
9	16	5	11
10	19	7	19
11	21	7	9
12	13	6	6
13	14	8	7

Table 3 shows how the LOC and Functions are adjusted using Previous Research and LOC and Clones are adjusted using our Proposed Research in total 10 and 9 clusters respectively using K-mean method. You see the Lines of Code are condensed in actual as redundancy removed. After clustering process, LOC present in table 3 attached to COCOMO Basic tool (developed in C# by us) then we find the better results in term of development efforts and development time which are shown in figure 3 with respect to previous and our research.

Table 2. Agglomeration Schedule

	Previous Work			**Our Work**		
Stage	**Cluster Combined**		**Coefficients**	**Cluster Combined**		**Coefficients**
	Cluster 1	**Cluster 2**		**Cluster 1**	**Cluster 2**	
1	5	11	1.000	12	13	2.000
2	5	10	2.500	5	11	5.000
3	12	13	5.000	6	9	5.000
4	6	9	10.000	5	8	7.500
5	6	12	13.000	6	12	23.000
6	2	4	13.000	1	3	32.000
7	5	8	13.667	4	5	40.667
8	1	3	25.000	4	6	69.875
9	5	6	38.375	2	10	85.000
10	2	5	105.500	1	7	178.000
11	1	7	175.500	2	4	187.250
12	1	2	1034.833	1	2	1015.800

Table 3. Final Cluster Centre

Clusters (Previous Work)										**Clusters (Our Work)**								
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
LOC										**LOC**								
42	28	46	26	20	15	57	22	16	14	42	28	46	26	14	17	57	21	19
Functions										**Clones**								
18	11	21	8	7	8	20	10	5	7	19	21	23	7	7	10	22	11	19

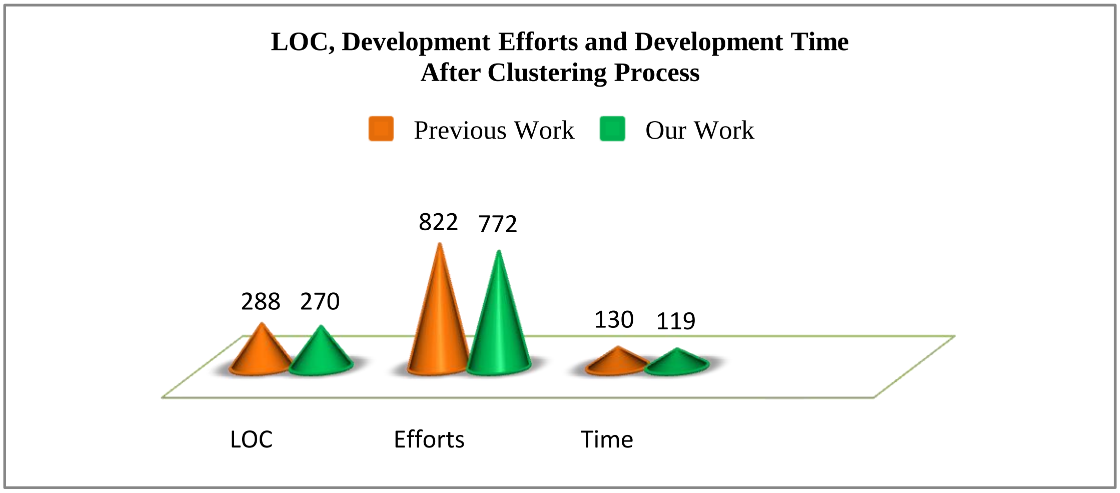

Figure 3. Comparison of Results

5. Conclusion

A large numbers of software are available in the market and the need of software increasing day by day but the manpower is not increasing proportionately. So there is big need to reuse software modules in new projects as well as in old one for effective development and maintenance. For this intention it is very crucial to organize those software modules in a meaningful pattern, which can be ended by cluster analysis. After applying clustering methods, we find the predictable 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, after implementation of cluster analysis the number of LOC, development efforts and development time are reduced. So this research is effective to overcome the problem of software crisis or failure at the large extent. Although the previous research is also a good idea for reusability but it is not valid to justify similarity between modules using the number of functions, so we applied string matching method to find out clones between modules using JB Clone Scanner that is developed in C# by us. The experiments have been done with 13 C++ modules only. Our approach can be effective for a large dataset and automation of this approach will be a milestone in software engineering for maintenance purpose. In future a more refined approach can be introduced using existing clone detection algorithms or after enhancement in these techniques.

References

[1] 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.

[2] Neeraj Verma, "Automatic Categorization of Modules Through Cluster Analysis", Master Thesis, Guru Jambheshwar University of Science & Technology, India.

[3] Manhas S., Sandhu P. S., Chopra V., Neeru N. (2010), "Identification of Reusable Software Modules in Funcion Oriented Software System using Neural Network Based Technique", World Academy of Science, Engineering and Technology, Vol. 67.

[4] 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.

[5] 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.

[6] 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.

[7] Cancino A. E. "Load Profiling of MERALCO Residential Electricity Consumers using Clustering", Manila Electric Company (MERALCO), Pasig City, Philippines.

[8] Czibula I. G., Serban G. (2007), "Hierarchical Clustering for Software System Restructuring", Babes Bolyai University, Romania.

[9] 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.

[10] Maqbool O., Babri H. A., "The Weighted Combined Algorith: A Linkage Algorithm for Software Clustering", Lahore University of Management Sciences DHA Lahore, Pakistan.

[11] Nakkrasae S., Sophatsathit P., "A Formal Appraoch for Specification and Classification of Software Components", University of Louisiana at Lafayette, U.S.A., Chulalongkorn University, Bangkok, Thailand.

[12] Jalender B., Govardhan A., Premchand P. (2011), "Breaking the Boundaries for Software Components Reuse Technology", International Journal of Computer Applications, Vol. 13, Issue 6.

[13] 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.

[14] 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.

[15] Jeng J. J., Cheng B. H. C. (1995), "Specification Matching for Software Reuse: A Foundation", ACM, PP. 97-105.

[16] 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.

[17] Fokaefs M., Tsantalis N., Chatzigeorgiou A., Sander J. (2009), "Decomposing Object-Oriented Class Modules Using an Agglomerative Clustering Technique", IEEE, Proc. ICSM, Canada.

[18] 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.

First Author, Jai Bhagwan has been a bonafide student of Master of Technology (Computer Science and Engineering) degree in Maharishi Markandeshwar University, Mullana, India during the period 2009-2011. He had been awarded for Best Explained Model of a satellite launching through PSLV vehicle in Kurukshetra University, Kurukshetra during his schooling. Currently he is working as Assistant Professor in Information Technology Department of Maharishi Markandeshwar University, Mullana, India.

He has also worked as a software developer for 7 months before his M.TECH degree. His current research interests are Software Engineering, Data Mining.

Second Author, Ashish Oberoi is working as Assistant Professor in Computer Science and Engineering department of Maharishi Markandeshwar University, Mullana, India.

He is Master of Technology in CSE discipline. He has 6 plus years of teaching experience and he is pursuing his PhD in Image Processing field. His research areas are Image Processing, Software Engineering. He has guided many Master students in the area of software engineering.