A Survey Paper On Program Restructuring And Component Reuse With Data Mining Techniques

Matrika Sinha

P.G. Scholar

Department of Computer Science & Engineering

SSGI, SSTC, Bhilai (C.G)

matrika.sinha.22@gmail.com

Prof. Shreya Jain

Assistant Professor

Department of Computer Science & Engineering

SSGI, SSTC, Bhilai (C.G)

shreyajain.0312@gmail.com

Abstract—Software engineering is the study and an application of engineering to the design, development and maintenance of software. It is a disputation for the technical person, as it is difficult for them to produce software which is very cheap, fast functions better. The benefit of Software resembles in the growth of the computer hardware. Component based software system is reuse based method. By the reusability the component can have better qualified, cheaper cost, improved performance. The reusable software component works better than the existing software as they are created with overcoming of the existing software module. Provide more standardized system. The objective is to estimate high reusability of component because it decrease the cost and implementation time as compared to develop new system. These papers present a brief study about a process or method through which we will classify the reusable component in proper way to get the full benefits of reusability. The output is a set of highly cohesive component.
Keywords— software component , hierarchical clustering, fuzzy logic, nueral network, xnor similarity function, fuzzy clustering .

INTRODUCTION

Software reuse is the process of repeated use of the software designed for reuse. Due to the benefits of reusability like reduction in time, cost, maintenance or work as compared to develop a new system. If we apply reusability into software component then reusable component are maintaining easily and usually have a higher quality value. There is a need to organize and classify collection of component so that we can identify the suitable component for potential reuse. Classify the reusable components according to their cluster. Clustering is mainly the process of making the group of similar type of component. The benefit of grouping over categorization is that, it is flexible for modification as well as assist distinct feature that illustrate dissimilar group. Software reuse consists of two approaches that are development of code from abrasion and extraction of reusable code from the existing code. Code reuse is extracting the partial or complete written programming code from the application software system and reuses it to write a new code in order to adopt in the existing application software system. Reusing limits the development time and reduces the energy to develop the code by reusing the programming code.

Researchers have proposed various methods for reuse the component. [1] Design and define an algorithm for clustering the document. Authors have discussed a clustering of components on the basis of XNOR similarity function to find degree of similarity between two document sets or software component. [2] Proposed a method through which classify the reusable components in proper way to get full benefits of reusability. [3] Adaptive fuzzy clustering technique proposed based on possibility clustering algorithms to address the issue of single metric. [4] Compares the sensitivity analysis of the two models depending upon different parameters: Modularity, Interface Complexity, Maintainability, Flexibility and Adaptability for accessing Software Reusability level using Soft computing techniques. [5] Reusable component technology is used in order to improve the efficiency and quality of management information system. [6] Proposed program restructuring at the functional level based on the clustering technique with cohesion. [8] Proposed various algorithms and techniques for efficiently retrieval of components from the component repository. [10] Define how to collect useful information on software component reusability and the factors on which reusability of the component is highly dependent. [15] Quantitatively compare two clustering techniques for clustering a repository of classes from a Java API for building mobile system. [17] Cluster the software components and form a subset of libraries from the available repository.

The rest of the paper is organized as follows: Section II the problem of software reusability is described. Section III presents a brief overview of clustering technique. Section IV presents the results analysis. Section V presents conclusions and future scope.

PROBLEM DEFINITION

Software development self addressed challenges of increasing complexity and dependence on external software by that specialize in one system at a time and on delivery deadlines and budgets, whereas ignoring the evolutionary needs of the system. This has led to a number of problems: the failure of the majority of projects to fulfill their deadline, budget, and quality requirements and the continued increase within the costs related to software maintenance. If new software products are each time to be developed from scratch, these goals can't be achieved. The key to the solution to this problem is Reusability. From this perspective Component-Based Software Development (CBSD) appears to be the correct approach. In CBSD software systems are built by grouping components already developed and prepared for integration. [13] CBSD are the provision of support for the development of systems as assemblies of components, the development of components as reusable entities, and also the maintenance and upgrading of systems by customizing and replacing their components. [10] The main objective of CBSD is to write once and reuse any number of times with no or minor modification .CBSD has several benefits. These include more effective management of complexity, reduced time to market, increased productivity, improved quality, a greater degree of consistency, and a wider range of usability.

METHODOLOGY

Program restructuring or code reusability can be used to transform such programs or poorly-designed ones to another form that is better organized and easier to understand, without changing the behaviour of the programs. For that, the input data set is a component-attribute data matrix. Components are the entities that we want to group based on their similarities. The components could be software modules; the attributes, a set of features.The Program restructuring and component reuse approach is based on clustering analysis for the entities and their attributes extracted from source code. It also keeps software maintenance cost under control. Following different clustering techniques for component reuse and program restructuring are-

A. Hierarchical Agglomerative Clustering (HAC)

To assign each component to a cluster we use HAC algorithm. This algorithm merges the two closest similar clusters. It will be repeated until; there is only one cluster left. Weighted Pair-Group Method using Arithmetic Averages (WPGMA), Complete Linkage Algorithm (CLINK), and Single Linkage Algorithm (SLINK) are the three commonly used HAC algorithms. The distance between two clusters in WPGMA is finding by taking the average of distance between all pairs of components in the two clusters. CLINK takes the distance between the most distant pair of components, one from each cluster SLINK find by measuring the distances between the closest pair of components, taking one component from each cluster. There are many problem with HAC algorithm that are when large amount of data is provided then its perform it very slowly and it is highly sensitive with halting criterion that is by mistakes it can merge valuable clusters into one cluster and also it not scale well. Previously done things can never be undo by using HAC algorithm. It has to recalculate over and again at each repetition in order to find similarity between clusters.

B. Fuzzy logic

Fuzzy logic is a methodology that deals with uncertain and imprecise data. Fuzzy logic system can be modeled with little data or without any data. Fuzzy logic technique has many advantages over other soft computing techniques. One of them is that it is less reliant on previous values. It may also work without data. So it will result in better understanding the reuse task. It considers factors like customizability, interaction complexity, understand ability, commonality and portability as input while reusability of component is considered as output. This model can be used to predict the reusability of CBSD, which will help in estimating development efforts and quality for the application.

C. Nueral Network

Neural Network (NN) can be used to represent the complex non-linear associations (input-output relationship) and functions. NN is used to predict the reusability levels of software by establishing the relationship between the inputs and outputs based on its training. It produces the correct output from its training data or incomplete data. NN model is more stable then the fuzzy model for building most of the software and application. It is also used to represent the software reusability functionality.

D. XNOR Similarity Function

To design a clustering algorithm firstly we will have to make the similarity function which is the core part of any clustering algorithm. Generally we use Hybrid XNOR function as a similarity function, which is used to calculate the similar features between any pair of entities which may be software components or documents. The documents may be a software product document or a code fragments or it may be text files to be classified. The input for components clustering algorithms are software components with properties predefines and the output for components clustering algorithms is a set of highly cohesive components with low coupling feature. According to the XNOR similarity function we find out the similarity between the two documents and it will produce value Z (worst case) when the feature is absence in one of the documents and it will produce value zero when the feature is absence in both

documents and it will produce value one if the feature is present in both the documents. It normalize the documents highly cohesive software component is produce by the use this algorithm.

E. Fuzzy Clustering

One challenge in clustering is to identify the cluster for some components that do not reveal strong relationship with a particular cluster or any clusters, or have high coupling with multiple clusters. It has been proposed to identify those "fuzzy" components by showing explicit membership values that a component is associated within all prospective clusters. The membership values are an indication of which cluster a component belongs to. The component belongs to the cluster with which it has the highest membership value. Fuzzy clustering has three requirements that is Membership value or degree, Total Membership, No empty clusters are allowed and no cluster may contain all the data. Compared with different clustering technique, Fuzzy clustering is more flexible and practical for real world complicated uneven distributed data, because during the clustering iteration, data instances can belong to more than one cluster with different membership values. Fuzzy clustering obtains the optimal result.

RESULT

After studying and analyzing different clustering techniques in order to address the issue of software component reuse and program reusability we conclude that fuzzy clustering is more robust to data distribution and more flexible to number of clusters as compared to other techniques. This technique can efficiently group data into cohesive clustering. The overall effort is reduced here. Effort includes number of lines of codes, effort of development and development time. The graphical representation of effort of development and development time before and after clustering technique is shown below
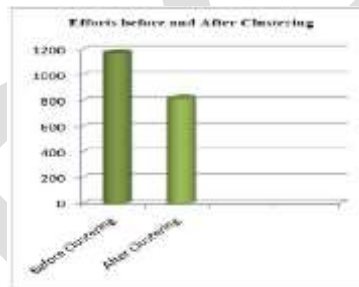

Fig.1. Comparison of Development Effort before and after clustering.
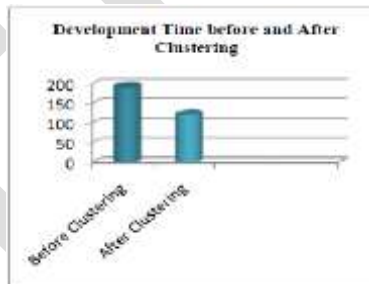

Fig.2. Comparison of Development Time before and after clustering.

CONCLUSION

In this paper a wide survey has been done on different clustering technique for reusability of software component and program restructuring. The results provided by Fuzzy Clustering are valuable and objective information. These results obtained are effective in handling the two common challenges for clustering technique the first one being the determination of number of clusters and the other one is determination of a software module for some highly coupled components.

In future, fuzzy clustering is used to estimate high reusability of component and program restructuring.

REFERENCES:

[1] Chintakindi Srinivas, Vangipuram Radhakrishna and C.V. Guru Rao, "Clustering software components for program restructuring and component reuse using hybrid XNOR similarity function," Procedia Technology 12 , pp. 246–254, 2014.

[2] Muhammad Husnain Zafar, Rabia Aslam and Muhammad Ilyas, "Classification of reusable components based on clustering," I.J. Intelligent Systems and Applications, vol. 10, pp. 55-62, 2015.

[3] Duo Liu, Chung-Horng Lung and Samuel A. Ajila, "Adaptive clustering techniques for software components and architecture," IEEE 39th Annual International Computers, Software & Applications Conference, pp. 460–465, 2015.

[4] Charu Singh, Amrendra Pratap and Abhishek Singhal, "Estimation of software reusability for component based system using soft computing techniques," 5th International Conference- Confluence The Next Generation Information Technology Summit (Confluence), pp. 788–794, 2014.

[5] Meng Shang, Haitao Wang and Longqiang Jiang, "The development process of component-based application software," International Conference of Information Technology, Computer Engineering and Management Sciences, pp. 11–14, 2011.

[6] Chung-Horng Lung, Xia Xu, Marzia Zaman and Anand Srinivasan, "Program restructuring through clustering technique," Journal of Systems and Software, vol. 79, pp. 1261-1279, September 2006.

[7] Abdulaziz Alkhalid, Chung-Horng Lung, Duo Liu and Samuel Ajila, "Software architecture decomposition using clustering techniques," IEEE 37th Annual Computer Software and Applications Conference, pp. 806–811, 2013.

[8] Amarjeet Kaur and Iqbaldeep Kaur, "Design and development of algorithm for software components retrieval using clustering and support vector machine," International Journal of Innovation in Engineering and Technology (IJIET), vol. 5, pp. 28–39, April 2015.

[9] Amit Kumar, Balkar Singh and Sandeep, "Improve reusability of software using clustering techniques," IJCSMS International Journal of Computer Science & Management Studies, vol. 14, issue 02, pp. 17-21 February 2014.

[10] Swati Thakral, Shraddha Sagar and Vinay, "Reusability in component based software development – a review," World Applied Sciences Journal 31 (12), pp. 2068-2072, 2014.

[11] Vishal Gupta and Gurpreet S. Lehal, "A survey of text mining techniques and applications," Journal Of Emergency Technologies In Web Intelligence, vol. 1, no. 1, pp. 60-76, August 2009.

[12] Arvinder Kaur and Kulvinder Singh, "Component selection for component based software engineering," International Journal of Computer Applications(0975-8887), vol. 2, pp. 109-114, May 2010.

[13] Ivica Crnkovic, "Component-based software engineering - new challenges in software development," 25th Int. Conf. information Technology lnterfaces ITI, pp. 9–18, June 16-19 2003.

[14] Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee, "A fuzzy self-constructing feature clustering algorithm for text classification," IEEE sTransactions On Knowledge And Data Engineering, vol. 23, no. 3, pp. 335–349, March 2011.

[15] Ronaldo C. Veras and Silvio R. L. Meira, Adriano L. I. Oliveira and Bruno J. M. Melo, "Comparative Study of Clustering Techniques for the Organization of Software Repositories," 19th IEEE International Conference on Tools with Artificial Intelligence, pp. 210–214, 2007.

[16] Rachana Kamble and Mehajabi Sayeeda, "Clustering Software Methods, Comparison," Int.J.Computer Technology and Applications, vol. 5, no. 6, pp. 1878 –1885, Nov-Dec 2014.

[17] Chintakindi Srinivas, Vangipuram Radhakrishna and Dr.C.V.Guru Rao, "Clustering and classification of software component for efficient component retrieval and building component reuse libraries," 2nd International Conference on Information Technology and Quantitative Management, ITQM, Procedia Computer Science 31, pp. 1044–1050, 2014