Component Clustering Based on the Reusability through Relational K-Means Technique

Matrika Sinha
M.E Scholar, Department of Computer Science & Engineering, SSGI, SSTC, Bhilai (C.G), India.

Shreya Jain
Assistant Professor, Department of Computer Science & Engineering, SSGI, SSTC, Bhilai (C.G), India.

Abstract – Software engineering is the application of engineering to the design, development, implementation and maintenance of software in a systematic method. Today most of the applications developed using some existing libraries, codes, open sources etc. As a code is accessed in a program, it is is represented as the software component. These components are able to use programming code or controls that excel the code development. A component based software system defines the concept of software reusability. Software reuse is that the method of making software systems from existing computer code instead of building them from scratch. Hard and soft clustering method used in this paper. The objective is to cluster the component in a judicial way. The result is found that hard partitioning allow with relational matrix performing better than the object with fuzzy partition.

Index Terms – Component, reusability, k-mean clustering, fuzzy clustering, relational matrix.

1. INTRODUCTION

As software systems become more and more complex, software programmers must know a variety of information and knowledge in varied areas. Throughout the software development process, the management and maintenance of knowledge creation is necessary thing. Then only that knowledge is integrated to develop the innovative concept from the older one. So the company should store and manage it for reuse. The basic idea behind building Reusable software components is to design interchangeable components from other industries to the software field of construction. Components are nothing but the smaller module that consists of classes and services which is defined in an application software system. Restructuring a program can make it easier to understand to design of a program and can assist in finding reusable components. Component assessment consists of various steps as find the component, verify the component, and finally store the component within the repository. We 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. A programmer cannot be expected to reuse an existing part unless its functionality is crystal-clear. A component will only be reused if its behavior is completely and unambiguously specified in a form understandable by potential programmers. 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.

The rest of the paper is organized as follows. Section II describes related work. Section III, the problem of component reusability is described. Sections IV describe the overview of component reuse through clustering framework. Section V, results is discussed. Section VI presents conclusions and future scope.

2. RELATED WORK

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. [7] Proposed a set of software matrix that will check the interconnection between the software component and the application. [8] Proposed various algorithms and techniques for efficiently retrieval of components from the component repository. [9] Describes how to build the code level reusable components and how to design code level components. [10] Define how to collect useful information on software component reusability and the factors on which reusability of the component is highly dependent.

3. PROBLEM DEFINITION

If new software products are each time to be developed from scratch, 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. The key to the solution to this problem is Reusability. Sparsity or sparse matrix problem and cluster imbalance problem occur while component reuse.

Sparsity: A sparse matrix is a matrix in which most of the elements are zero. The fraction of nonzero elements over the total number of elements in a matrix is called the sparsity or density. Operations using standard dense matrix structures and algorithms are slow and inefficient once applied to large sparse matrices as processing and memory are wasted on the zeroes. MATLAB stores every matrix element internally. Zero valued elements require the same amount of space for storing as any other matrix element. One disadvantage of the sparse formats is that assignment is much slower than for standard matrices.

Cluster Imbalance: Class or cluster imbalance problem become greatest issue in data mining. Imbalance problem occur where one of the two classes is having more sample than other classes. The minority samples are those that rarely occur but very important. In an imbalance data set the majority class has a large of all the samples. In this case, classifier usually tends to predict that samples have the majority class and completely ignore the minority class.

4. METHODOLOGY

Component reuse and clustering through relational behavior of software expressed by following methods:

A. Data Acquisition

Data acquisition is defined as the process of collection and organizing information. Component or data acquisition is that the method of acquiring components for reuse or development into a reusable component. It may involve accessing locally-developed components or services or finding these components from an external source. The programmers need to search for right components matching their needs in a very database of components. A number of software components selected from a set of components or from components repository in such a way that their composition satisfies a collection of objectives. Here selection of components and deployment of component into applications is used to minimize the total cost, considering compatibility among components.

B. Term Frequency

Term Frequency (TF) that measures how frequently a term occurs in a document. Since every document is totally different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is usually divided by the document length (the total number of terms in the document) as a way of normalization:

TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document).

TF can be successfully used for stop words filtering in varied subject fields as well as text summarization and classification. If a word appears frequently in a document, it's important. Give the word a high score. But if a word appears in several documents, it's not a unique identifier. Give the word a low score.

C. Document Frequency

The document frequency, defined to be the number of documents in the collection that contain a term. Term frequency is how many times a term appears in a particular document in your corpus. Document frequency is how many of the documents in your corpus a term appears in (and inverse document frequency is the multiplicative inverse of this number).

D. Term Matrix

Document term matrix or term document matrix (TDM) could be a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document term matrix, rows correspond to documents within the collection and columns correspond to word or terms (in the collection vocabulary).We generate TDM by arranging our list of all content words on the vertical axis, and a similar lists of all documents on the horizontal axis. These needs not be in any explicit order, as long as we keep track of which column and row corresponds to that keyword and document. The keywords are an alphabetized list. We fill in the TDM by going through each document and marking the grid square for all the content words that seem in it because any one document will contain only a tiny subset of our content word vocabulary, our matrix is very sparse.

E. Fuzzy Clustering

In non-fuzzy or hard clustering, data is split into crisp clusters, where every data point belongs to exactly one cluster. In fuzzy clustering also mentioned as soft clustering, the data points will belong to more than one cluster, and associated with each of the points are membership grades which indicate the degree to which the data points belong to the various clusters. One of the most widely used fuzzy clustering algorithms is the Fuzzy C Means (FCM) algorithm. It starts with an initial guess for the cluster centers, which are meant to mark the mean location of each cluster. The initial guess for these cluster centers is possibly incorrect. Next, it assigns every data point a membership grade for each cluster. By iteratively updating the cluster centers and also the membership grades for each data point, FCM iteratively moves the cluster centers to the correct

location within a data set. This iteration is based on minimizing an objective function that represents the distance from any given data point to a cluster center weighted by that data point's membership grade.

F. Relational Matrix

A distance matrix or relational matrix is a matrix containing the distances, taken pair wise, between the elements of a set. A relation matrix could be matrixes which calculate the relation or distance between documents to each other document.

G. K-mean Clustering

K-mean is one of the simplest unsupervised learning algorithms that solve the documented clustering problem. The procedure follows an easy and straightforward way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to put them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is completed. At this point we need to recalculate k new centroids of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no additional changes are done or in other words centers do not move any more.

5. EXPERIMENTAL SETUP AND RESULT

An Intel Pentium Dual core with 3GB RAM computer is used for conducting the experiments. The main software tool used is MATLAB. The database consists of programs or documents.

Consider there are ten documents or programs as a training set. From each document remove stop words and stemming words. Find unique words in each document and count of the same. Form a word set consisting of each word in frequent item set of each document. Calculate the term frequency of the word set. It consist the most frequently used words in each document, frequency of the word that is the number of times that word appeared in the document. Calculate the relative frequency of the word. Now construct a matrix with row indicate each document and column indicate to each unique frequent item set of all document. It shows that a particular item or component appears how many times in the particular document. This is shown in the table I.

TABLE I. Represent a Matrix

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
D1	0	2	0	3	0	4	1	1	4	0
D2	3	2	5	3	2	2	2	0	0	0
D3	0	0	0	0	0	0	0	2	2	2
D4	3	2	0	3	2	2	0	0	0	0
D5	0	0	0	0	0	0	0	4	0	1
D6	3	2	5	3	2	2	2	0	1	0
D7	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	5	0	0	3	0	2
D9	0	0	5	0	0	0	0	0	5	0
D10	3	2	0	3	0	0	2	0	0	0

Once obtain the above table then the component is divided into 3 clusters. For that Fuzzy clustering is apply and find which document belong to which cluster.

TABLE II. Represent the documents divided into cluster

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
cluster	1	2	1	1	1	2	1	1	3	1

In the table II row indicates the cluster and columns indicate the document. The total of document belongs to cluster 1 is 7, cluster 2 is 2 and cluster 3 is 1.

In fuzzy clustering, the documents are unevenly distributed. So to get evenly distributed document apply relational matrix in table I. For the relational matrix, here Euclidean distance matrix is used, which calculate the distance of document with each other.

TABLE III. Relational Matrix

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
D1	0	7.7	6.2	5.9	7.4	7.2	6.8	8.8	7.5	6.5
D2	7.7	0	8.4	5.3	8.7	1	7.6	8.7	7.6	5.7
D3	6.2	8.4	0	6.4	3	8.2	3.4	5.4	6.4	6.1
D4	5.9	5.3	6.4	0	6.8	5.4	5.4	6.9	8.9	3.4
D5	7.4	8.7	3	6.8	0	8.7	4.1	5.1	8.1	6.5
D6	7.2	1	8.2	5.4	8.7	0	7.7	8.8	7.0	5.8
D7	6.8	7.6	3.4	5.4	4.1	7.7	0	6.1	7.0	5.0
D8	8.8	8.7	5.4	6.9	5.1	8.8	6.1	0	9.3	8
D9	7.5	7.6	6.4	8.9	8.1	7.0	7.0	9.3	0	8.7
D10	6.5	5.7	6.1	3.4	6.5	5.8	5.0	8	8.7	0

To decrease the numbers of zero from the table I matrix, relational matrix is created. This matrix solves the problem of sparsity. Through relational matrix k-mean clustering is created. In k-mean clustering again document is divided into cluster. Row defines the cluster and column defines the document.

TABLE IV. Documents divided into cluster

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
cluster	3	1	2	3	2	1	2	2	1	3

In k-mean clustering documents are distributed evenly as compare to fuzzy clustering. Cluster 1 contain 3 document, cluster 2 contain 4 document, cluster 3 contain 3 document. If Software developers had a collection of reusable software components, and then software industry could build applications by simply plugging existing components together.

The set of cluster finally formed for Fuzzy clustering is

Cluster-1: {1, 3, 4, 5, 7, 8, 10}
Cluster-2: {2, 6}
Cluster-3: {9}
For K-mean clustering is
Cluster-1: {2, 6, 9}
Cluster-2: {3, 5, 7, 8}
Cluster-3: {1, 6, 10}

6. CONCLUSION

In this paper present comparison between fuzzy cluster and k-mean cluster is done for the reusability of software component. The results provided by relational with K-mean Clustering are valuable and objective information. Fuzzy clustering form cluster which contain unevenly distributed document. Compared with Fuzzy clustering, relational K-mean clustering is more flexible and practical for real world.

In future, relational behavior of software work with Fuzzy clustering or some other clustering technique to estimate high reusability of component can be used.

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] Prakriti Trivedi and Rajeev Kumar, "Software metrics to estimate software quality using software component reusability," International Journal of Computer Science Issues, vol. 9, issue 2, no 2, pp. 144-149, March 2012.

[8] 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.

[9] B.Jalender, Dr A.Govardhan and Dr P.Premchand, "Designing Code Level Reusable Software Components," International Journal of Software Engineering & Applications (IJSEA), vol.3, no.1, pp. 219-229, January 2012.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.