I J C T A, 9(11) 2016, pp. 5443-5448

FREFCOSCO: A FUZZY RELATION- BASED FUZZY CLUSTERING OF SOFTWARE COMPONENTS

Jagdeep Kaur[*] and Pradeep Tomar[**]

Abstract: The Component-Based Software Development achieves software reusability by creating software systems from the current software rather than building from scratch. Successful reuse requires high quality components in a repository with suitable description, classification and retrieval mechanism. In software components selection, a number of software components selected from a subset of components or from components repository in such a way that their composition satisfies a set of objectives. The component selection process consists of picking up a component from the given set based on some quality attributes. This paper presents an algorithm based on fuzzy clustering for software component selection. The main advantages are: i) It eliminates the need of choosing a particular similarity function. As choosing the similarity measure was very crucial in fuzzy clustering also the results may vary for different similarity measures even for the same set of data. ii) With the proposed technique multi-dimensional data numerical as well as categorical can be handled. iii) The prior specification of the number of clusters is not required. iv)The components are clustered based on multi-features rather than considering one or two features. The main steps of the algorithm are: normalization of initial dataset, calculation of fuzzy transitive closure, partition set generation using set similar to property and lastly cluster formation based on membership function values. The algorithm is validated on a case study.

Key Words: Fuzzy Clustering, Fuzzy Transitive closure, Component-based software development and Software Component Selection.

1. INTRODUCTION

Software Reusability is the process of creating software from the existing software, rather than building software from scratch. It is an effective way to shorten the development cost and time. There are different approaches to achieve reusability namely program libraries, program generators, design patterns, application frameworks, legacy system wrapping and software components. A software component is a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard. The software components are acquired, adapted and integrated with minimum cost and effort by using Component-Based Software Development (CBSD) paradigm. The major activities of CBSD are component selection, adaptation and composition. The component selection deals with ensuring that the component performs its desired functionality and is according to stated quality characteristics like reliability, performance and usability. The component repository may be very large number of components. Normally the exploration of the components is carried using

[*] CSE & IT Department, The NorthCap University, Gurgaon, Haryana, India. Email: jagdeep_kaur82@rediffmail.com
[**] Department of Computer Science & Engineering, School of ICT, Gautam Buddha University, Greater Noida, Uttar-Pradesh,India. Email: parry.tomar@gmail.com

approximate automated search. The problem deals with selection of components from the repository. It is possible that several types of components are retrieved having similar functionality. In that case how to choose the optimal component based on the given information about the components. The main objectives of this paper to propose an intelligent technique for most suitable software component selection In such situations data analysis is required and for such "things of one kind come together" problem, cluster analysis is the perfect choice. The fuzzy clustering methods are divided into two type: distance-defined-objective functions and fuzzy relation based methods. The first category covers the Fuzzy c-means algorithm and its variants. The second category is fuzzy relation based methods. Here the fuzzy relations matrices are used to present the similarity between the objects. Using this approach of fuzzy relation based clustering, the components are classified into different clusters and the same component may lie in more than one cluster. In the next section, a brief outline of the related work in software component selection is given.

2. RELATED WORK

Many methods/techniques/algorithms have been proposed for software component selection based on clustering. A software component selection using analytic network process based on quality criteria like effectiveness, efficiency, satisfaction, safety and usability is proposed by *Nazir et al.* [1]. Another technique is proposed *Nakkrasae et al.,* [2] where a coarse grain classification based on fuzzy subtractive clustering. Subsequently, its correctness is checked by precision and recall method and if satisfactory results are found the fine grain classification is made. In [3] *Serban* suggested another component selection process based on some quality attributes of the components. Firstly, metrics are made to quantify the attributes of the components. By using these metrics, fuzzy clustering is applied to select the best candidate. While reviewing the fuzzy relation-based cluster analysis we found many researchers have established this process. According to *Quanming and Jianhua*[4], an advanced fuzzy cluster analysis algorithm is proposed based on fuzzy equivalence relation. It was further demonstrated that the fuzzy equivalence matrix can generate the fuzzy clustering. Another attempt by *Yang and Shih* [5], is to use the fuzzy cluster analysis based on fuzzy relations using max-t similarity relations. Another methodology by *Guh et al.*[6] is based on clustering algorithm using fuzzy relation to produce different partition trees with different levels according to different t norms. In [7] *Rahman* presented a fuzzy clustering technique for data mining of records based on their importance.

After going through the literature survey the authors refer to their previous work[8]. in the four tier architecture clustering using Hybrid XOR where similarity function was applied which is now replaced by new fuzzy clustering technique based on fuzzy relations. This technique would be discussed in the next section. It is realized that most of the previous work done for software component selection based on fuzzy clustering is either using the default values of acceptance and rejection ratios of MATLAB built-in fuzzy clustering methods or the number of clusters generated are less as compared to actual number of clusters that can be generated. Over the past few years attempts have been made to help the application developer for finding a suitable component from the repository for reuse.

3. THE PROPOSED ALGORITHM FOR SOFTWARE COMPONENT SELECTION: FREFCOSCO (FUZZY RELATION-BASED FUZZY CLUSTERING OF SOFTWARE COMPONENTS)

We present the proposed algorithm, FREFCOSCO (Fuzzy RElation based Fuzzy Clustering Of Software COmponents). As it is clear from the literature survey, earlier many efforts were made for software component selection. The main idea of FREFCOSCO (Fuzzy RElation based Fuzzy

Clustering Of Software COmponents), is to make a fuzzy similarity relation matrix based on number of features. The main contributions of the algorithm are:

The reduction in search space for component retrieval and clustering based on multi-attributes rather than considering one or two attributes, as compared to built-in functions of Fuzzy c-means and subtractive clustering MATLAB.

The use of fuzzy clustering based on fuzzy relations has eased the task of selection as the tedious task of minimization or maximization of objective functions is eliminated. Even the distance functions cannot be applied for subjective type dataset.

There is a dependency of clustering results on the distance metric for similarity or dissimilarity. It is removed using the proposed algorithm.

As demonstrated by [9], the simple fuzzy clustering technique requires the need of mentioning the number of cluster centres beforehand, in case of Fuzzy c-means and the radii of the cluster in case of subtractive clustering. The new algorithm eliminates this need.

The algorithm takes input in the form of matrix of components; the features can be mix of numerical as well as categorical types. The first step consists of normalizing the data, in terms of converting the categorical data to numerical one and assigning appropriate labels to the features. Further, the fuzzy relation matrix is formed by comparing the attributes of two components. The second step involves finding the fuzzy transitive closure so that it can be used as a similarity measure. The third step is to generate the partitions from the matrix obtained in step 2. The entries in the partition matrix can be obtained by using set similar to property. The fourth step, generates the final fuzzy clusters as the components are repeated in many clusters.

The main steps of the algorithm are:

Input: A Dataset Δ, number of components n

Output: A set of fuzzy clusters C

1. Normalize the dataset , Δ.

   $\Delta' \leftarrow$Normalize (Δ) /*Call Normalize () */\

2. Fuzzy Transitive closure Computation.

   $R^{(n)} \leftarrow$ Fuzzy_trans_clos(Δ,n) /*call Fuzzy_trans_clos() function */

3. Generate partition set of the matrix $R^{(n)}$

   $P^{\lambda} \leftarrow$GenPart($R^{(n)}$, Δ' , $\acute{\eta}$) /* call GenPart() function */

   for (λ = 1 to $\acute{\eta}$) do /* λ is the number of partition */

4. Cluster Formation

   $\varsigma \leftarrow$ Clust_form(P^i, $C^i{}_l$)

5. Termination conditions for FTransCoSel

6. if $P^{\lambda} \leftarrow \Phi$ do

   | Break : /* Terminate clustering as it meets the termination condition */

   end if

7. End for

8. Generate the final clusters with membership degree.

   C← Final_cluster (Δ')

9. Return C

4. EXPERIMENTAL RESULTS

The main aim of this section is to describe the actual execution of the algorithm and how it generates clusters for the user. The algorithm is executed on a dataset of thirty components taken from www.componentsource.com. The Barcode applications are used in logistics and supply chain management. It has many advantages like keeping track of the articles using barcodes, maintaining the database of these articles and transmitting the information over various communication links. Consider a case where software components are required for performing following

functions: It can read and detect industrial barcodes, it can generate GUI's for charting, email, spell check application, it must have printing capabilities and it must be capable of business and data analytics.

A set of thirty are retrieved from the online repository. The repository maintains the following information about the components: Functionality offered by the component, the cost of the component, the bestseller rating, the review based rating, the download rating for the trial version and the asset value that denotes the time and experience if the component is developed in house. The application developer is concerned about the cost which should be medium, the bestseller rating should be high, the compatibility should be high and download rating he/she is least concerned. In the present case, the features selected are: price range, compatibility, best seller rating and download rating.

The algorithm is implemented in MATLAB R2014b. These components are given as an input to the program in form of comma separated row, with four columns. The step one will process the inputs. The script can process upto n dimensions of data. For the current dataset, only four features are considered. These are price range, best seller rating, compatibility and download ratings. The price range is expressed as high, medium and low. The best seller rating varies from one to thirty. The compatibility of the components is expressed as 1 for yes and 2 for no. Based on the number of downloads of a particular components, the components are rated in download rating. It ranges from one to thirty. Hence, the input data is tabulated by MATLAB, as shown in Figure 1. So, the components are taken in form of a matrix of dimensions 30x30 as thirty components are initially retrieved. For instance, the first component, Aspose.Bar Code is in high price range, compatibility is high, bestseller rating is one and its rated tenth on download ratings. So, any number of components with any number of features can be taken as an input here as shown in fig. 2. The input values are processed: To normalize the attributes the price range of high, medium and low is represented as 1, 2 and 3, respectively. The download rating and best seller rating is categorized as the rating in 1-10 is assigned 1, the rating between 11-20 is assigned 2 and the rating between 21-30 is assigned 3. This matrix after processing the input is as shown in fig.4. The next step consists of normalization, where the pairwise comparison is made and the membership function is calculated for both the components as discussed in the algorithm. The input matrix is normalized as per the algorithm and results are as shown in fig.3. From the normalized matrix, the next step is to calculate the fuzzy transitive closure according to the n-step procedure given in the algorithm. There are different types of compositions of fuzzy relations that can be used. Namely, drastic product, bounded difference, Einstein product, algebraic product etc. As evident from (Guh et al., 2008), a higher level of fuzzy intersection will produce higher degree of similarity among the features. Hence, the max-min composition is used. After finding out the fuzzy transitive closure,

the next step is to generate partitions and finally the clusters are generated. Corresponding to thirty components there are thirty three clusters generated as tabulated in Fig. 4.

	Price Range	Best Seller Rating	Compatibility	Download Rating
1	High	1	1	10
2	High	2	1	30
3	Medium	3	1	11
4	High	4	1	12
5	Low	5	2	6
6	High	30	2	30
7	Low	10	2	27
8	Medium	11	2	25
9	Medium	8	2	24
10	Low	12	1	12
11	High	1	1	10
12	High	2	1	30
13	Medium	3	1	11
14	High	4	1	12
15	Low	5	2	6
16	High	30	2	30
17	Low	10	2	27
18	Medium	11	2	25
19	Medium	8	2	24
20	Low	12	1	12
21	High	1	1	10
22	High	2	1	30
23	Medium	3	1	11
24	High	4	1	12
25	Low	5	2	6
26	High	30	2	30

Figure 1: Step 1, The Input Matrix

	Price Range	Best Seller Rating	Compatibility	Download Rating
1	1	1	1	1
2	1	1	1	3
3	2	1	1	2
4	1	1	1	2
5	3	1	2	1
6	1	2	2	3
7	3	1	2	3
8	2	2	2	3
9	2	1	2	3
10	3	2	1	2
11	1	1	1	1
12	1	1	1	3
13	2	1	1	2
14	1	1	1	2
15	3	1	2	1
16	1	2	2	3
17	3	1	2	3
18	2	2	2	3
19	2	1	2	3
20	3	2	1	2
21	1	1	1	1
22	1	1	1	3
23	2	1	1	2
24	1	1	1	2
25	3	1	2	1
26	1	2	2	3

Figure 2: Step 1 after processing user Input

Cluster Name	Component 1	Component 2	Component 3
1	1	2	3
2	1	6	7
3	1	11	21
4	1	11	30
5	1	20	30
6	2	3	4
7	4	5	12
8	4	11	12
9	5	6	7
10	5	11	12
11	5	12	13
12	6	7	8
13	8	9	10
14	8	9	15
15	9	10	15
16	9	10	16
17	10	20	30
18	11	20	30
19	11	21	30
20	13	14	15
21	13	14	21
22	14	15	21
23	14	15	22
24	16	17	10
25	17	18	19
26	19	20	26
27	19	25	26
28	20	25	26
29	20	26	27
30	22	20	24
31	23	24	25

Figure 3: Components with different fuzzy clusters

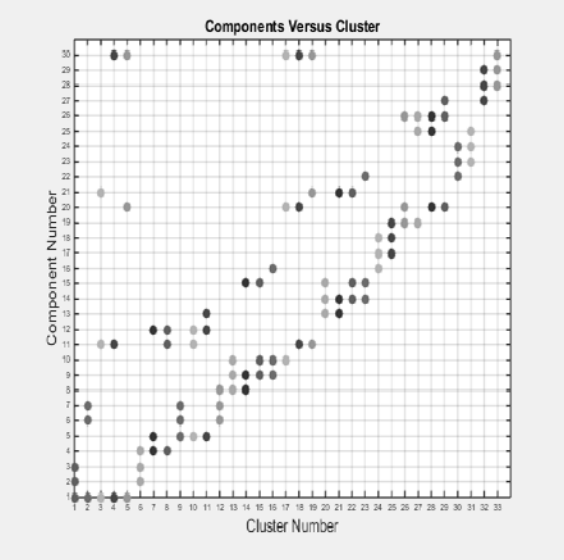

Figure 4: Plot for component vs. clusters

5. RESULTS & DISCUSSIONS

The final result of the fuzzy clustering is presented in fig. 4. This distribution of components over the cluster can be better visualized in the plot for components vs. clusters as shown in fig. 8. From the graph it is clear that each cluster contains at most three components and each cluster is represented by a different color. However, it can be seen that there are various fuzzy clusters present in the plot. Some components tend to lie in more than one cluster. For instance, the component 1 lies in cluster 1, cluster 2, cluster 3, cluster 4 and cluster 5. The next challenge is to validate the cluster. It is done by using silhouette coefficient value. The silhouette coefficient whose value represents how similar that component is to components in its cluster when it is

compared to components in other clusters. The average of the silhouette coefficients is calculated. Hence the application developer can select those clusters where the values closer to one. This algorithm can be compared with other algorithms for software selection on the basis of computation time shown in table 4. The main advantage of this current algorithm is its fast computation time. Its value is 100ms for thirty components.

6. CONCLUSION AND FUTURE SCOPE

To enhance the software reusability Component based Software Development (CBSD) is an important paradigm. The software component repositories play a vital role in component selection. The more organized, filtering mechanisms and access means are provided in the repository the lesser the efforts and time is required to search for the component. This current paper shows the usage of the fuzzy clustering for component selection. The fuzzy clustering used here is based on fuzzy relations. The main advantages of this approach are high computational time, ability to handle multi-features and mix type of features (both categorical and numerical)of components effectively and generation of small clusters. However, this work can be extended in future in two different ways. Firstly, for computing the fuzzy closure, this present algorithm has only taken into consideration the max-min composition. However, by application of different composition functions like bounded difference, Einstein product, algebraic product etc. different result may yield. Secondly, it works on the fact that the values of all the features under consideration are available. No missing values are in the feature set. In future work, the algorithm will be run for different component sets and their comparison in terms of computation time will be made.

References

[1] S. Nazir, S. Anwar, S. A. Khan, S. Shahzad, M. Ali, R. Amin, M. Nawaz, P. Lazaridis and J. Cosmas,"Software Component Selection Based on Quality Criteria using the Analytic Network Process" , Abstract and Applied Analysis, vol. 2014, pp. 1-12, 2014.

[2] S. Nakkrasae, P. Sophatsathit and W. R. Edwards. "Fuzzy Subtractive Clustering based Indexing Approach for Software Components Classification", International Journal of Computer and Information Science ,vol.5, no. 1, pp. 63-72, 2004.

[3] C. Serban, A. Vescan and H. F. Pop, " A New Component Selection Algorithm Based on Metrics and Fuzzy Clustering", Creative Mathematics and Informatics, vol. 1, no. 3, pp. 505-510, 2009.

[4] Z. Quanming and H. Jianhua. " An advanced fuzzy cluster analysis algorithm and the applications based on fuzzy equivalence relation" in Proc. of 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), pp. 616-619, 2013.

[5] M.S Yang and H.M Sih, "Cluster Analysis based on Fuzzy Relations", Fuzzy Sets and Systems, vol. 120, pp. 197-212, 2001.

[6] Y. Y. Guh, M. S. Yang, R. W. Po and E. S. Lee, "Establishing Performance Evaluation Structures by Fuzzy Relation-Based Cluster Analysis", Computers & Mathematics with Applications, vol. 56, pp. 572-582, 2008.

[7] M.A Rahman and M. Z. Islam, "CRUDAW: A Novel Fuzzy Technique for Clustering Records Following User Defined Attribute Weights," Aus. Data Mining Conference, pp. 27- 40, 2012.

[8] J.Kaur and P. Tomar, "Four Tier architecture for software component selection process using clustering", International journal of Software engineering & Application Technology, Vol. 2,(in press).

[9] K. M. Bataineh, M. Naji and M. Saqer, A Comparison Study between various fuzzy clustering Algorithms ,Jordan Journal of Mechanical and Industrial Engineering, Vol 5, No. 4, pp. 335-343,2011.