

# Hybrid Approach for Software Component Classification using Computational Intelligence

Pravind Kumar

*Department of Computer Science and Engineering,  
School of Information and Communication Technology,  
Gautam Buddha University,  
Greater Noida, UP, India*

Pradeep Tomar

*Department of Computer Science and Engineering,  
School of Information and Communication Technology,  
Gautam Buddha University,  
Greater Noida, UP, India*

Jagdeep Kaur

*CSE & IT Department, The NorthCap University, Gurgaon, Haryana, India*

**Abstract-**Software engineering provides solutions of the existing problems coming from different-different areas. Fuzzy C-Means and Subtractive Clustering Algorithm play a vital role in the Software Engineering for Software Components Classification in their respective Clusters. Fuzzy C-Means Clustering Algorithm has a major disadvantage that the number of Clusters formed, intimate before classification of Software Components. How can intimate the number of Software Components beforehand, it depends on the nature of data points used. This is the main cause to produce local optimal solutions instead of optimal solution. Fuzzy Subtractive Clustering Algorithm evaluates the number of required Clusters on the base of nature of Software Components. This paper uses a Hybrid Approach for Software Components Classifications by using Computational intelligence. The problems associated with Fuzzy C-Means can rectify to use to Fuzzy Subtractive Clustering Algorithm, but Computation time required is high that's why in this paper use rejection ratio, Recall and Precision to classify Software Components. Selecting a Cluster Center is main task in Clustering process because if selection of Cluster center is good then Computation time to classify Software Components in their respective Clusters is small.

**Keywords-**Fuzzy C-Means Clustering Algorithm, Fuzzy Subtractive Clustering Algorithm, Software Component Classification.

## I. INTRODUCTION

FCM is a method applied to put up in clustering and used in many area domain like image segmentation [1], pattern matching, pattern recognition, Artificial intelligence, Computational intelligence, etc. FCM is very sensitive to its initialization. If the initial value is not proper we can not find optimal solution, people have to set number of cluster initial time again and again for optimal solution. The size of Software Component is very large so the possibility of finding optimal solution is very low.

To find optimal solution will use FSC to initialize the initial value of FCM before use FCM to put up for clustering [2].

Software Component reuse is a popular design methodology common to engineering discipline. Reuse has two primary aspect one is cost reduction and other is time reduction. on the base of three properties Functional, Structural and Behavioral Software Component Classified [3]. Different values of three Properties (Functional, Structural and Behavioral) of Software Component depends on user chosen values. The minimum evaluated values of selected Software Components indicates, most suitable software components for reuse.

FSC combines with FSC Algorithm for optimal solution to Software reuse. Tables created on the base of combining steps of both the algorithms FSC and FCM clustering algorithm. Recall and Precision performance of rejection ratio play a vital role for Software Component Classification.

### 1.1 Software Component Representation

Define software component  $M$  be

$$M=(Str, Fun, Bhr)$$

Where  $Str$  be the structure properties,  $Fun$  be the Functional properties and  $Bhr$  be the Behavioral properties. Each of the three properties denotes of the form

$$Str=\{ Str_1, Str_2, \dots, Str_m \} \quad \text{Equation} \dots \dots \dots (1)$$

$$Fun=\{ Fun_1, Fun_2, \dots, Fun_n \} \text{ and} \quad \text{Equation} \dots \dots \dots (2)$$

$$Bhr=\{ Bhr_1, Bhr_2, \dots, Bhr_p \}. \quad \text{Equation} \dots \dots \dots (3)$$

## II. ALGORITHM FOR SOFTWARE COMPONENT CLASSIFICATION(SCC)

This paper use Hybrid Approach for Software Component Classification. FCM Clustering Algorithm will fall into local optimal solutions due to initialization sensitive of number of Clusters, so here FSC Algorithm overtake the problems associated with FCM. This paper use Subtractive Clustering Algorithm to initialize the initial value of FCM before use Fuzzy C-Means Clustering Algorithm to put up Fuzzy Clustering.

### 2.1 Fuzzy C-Means Clustering Algorithm(FCM)

FCM[4] assigned data point when these belongs more than one cluster. It is very sensitive to its initial value. It produce local optimal solution, if initial value is not good. So use Subtractive clustering algorithm before initialized the initial values of FCM as a result will gain optimal solution in stead of local maxima solution.

This procedure gives a local maxima instead of optimal solution due to outlier data points. FCM Clustering Algorithm may produce optimal solution when try with different – different Clusters values initialization. FCM Clustering Algorithm evaluate membership values of each data points by formula (7) and then stored in the matrix. Number of columns in the matrix is equal to the number of initialized cluster and number of rows is as the number of data points (Software Components). The maximum membership value in column is selected as first cluster center and as same as for second cluster center.

### 2.2 Fuzzy Subtractive Clustering Algorithm

FCM[4] assigned data point when these belongs more than one cluster. It is very sensitive to its initial value. It produce local optimal solution, if initial value is not good. So use Subtractive clustering algorithm before initialized the initial values of FCM as a result will gain optimal solution in stead of local maxima solution.

This procedure gives a local maxima instead of optimal solution due to outlier data points. FCM Clustering Algorithm may produce optimal solution when try with different – different Clusters values initialization. FCM Clustering Algorithm evaluate membership values of each data points by formula (7) and then stored in the matrix. Number of columns in the matrix is equal to the number of initialized cluster and number of rows is as the number of data points (Software Components). The maximum membership value in column is selected as first cluster center and as same as for second cluster center.

## III. HYBRID APPROACH FOR SOFTWARE COMPONENT CLASSIFICATION

This paper encompasses the Software Components which does not encompassed by Fuzzy C-Means Clustering Algorithm on the base of assigned  $r_a$  and  $r_b$  and allowable error before calculations.

$$r_a=14, r_b=1.5*r_a$$

Step1: First of all calculate density of all software components by formula

$$P_i = \sum_{j=1}^n e^{-\alpha} \|x_i - x_j\|^2 \quad \text{Equation.....(4)}$$

Where  $\alpha = 4/r_a^2$

$\| \cdot \|$  is the Euclidean distance and the selected software component with highest potential value as first cluster center  $X_{c1}$ .

Step2: After calculating first cluster center, calculate the density of remaining (n-1) software components by formula

$$P_i = P_i - P_k e^{-\beta \|x_i - x_k\|^2} \quad \text{Equation.....(5)}$$

Where  $x_k$  is point of the  $k^{th}$  cluster center,  $p_k$  is its potential value and  $\beta=4/1.25$  and then select second highest potential as second cluster center  $X_{c2}$ .

Step3: Give initial matrix  $U(0)$ .

Step4: Calculate the cluster center using formula

$$C_j = \frac{\sum_{i=1}^n u_{ij} x_i}{\sum_{i=1}^n u_{ij}} \quad \text{Equation.....(6)}$$

Step5: Update  $U(r)$  by using formula

$$u_{ij} = \frac{1}{\sum_{k=1}^m \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad \text{Equation.....(7)}$$

Step6: If the condition

$$U^{(r+1)} \leq (\epsilon) * (P_{max})$$

Where  $\epsilon$  is allowable error and  $P_{max}$  be the Component having maximum potential value.

#### IV. EVALUATION & RESULTS

Recall = Retrieved target software component /Total no. of target software components.

$$\frac{\sum_{i \in C_i} \#C_i}{\#C_i} \quad \text{Equation.....(8)}$$

Precision= Retrieved target software component /Total number of software components retrieved

$$\frac{\sum_{i \in C_i} \#C_i}{\#C_i} \quad \text{Equation.....(9)}$$

Where  $\#C_i$  denotes the number of elements on cluster  $C_i$  and  $0 \leq \text{recall, precision} \leq 1$ .

In this paper conduct trial with different rejection ratio groups (like 0.10-0.21,0.34,0.40-0.50) and used the derived cluster centers from each trail to calculate its recall and precision performance.

Let  $U$  denotes a software component requirement matrix which matches cluster center  $C3$ , this cluster center encompasses  $C1, C2, C3, C4$  and  $C5$ .

Table 1. Recall and Precision Performance on the base of Rejection Ratio 0.10-0.21

Selected cluster center	Software component relevant	Software component retrieved	Recall	Precision
SCC5	C1,C2,C3,C4,C5	C1	0.20	1.00
SCC3	C1,C2,C3,C4,C5	C2,C3,C4,C5	0.80	1.00
SCC5	C6,C7,C8,C9,C10	C6,C7	0.40	1.00
SCC8	C6,C7,C8,C9,C10	C8,C9,C10	0.60	1.00
SCC11	C11,C12,C13,C14,C15	C11	0.20	1.00
SCC13	C11,C12,C13,C14,C15	C13	0.20	1.00

SCC15	C11,C12,C13,C14,C15	C13,C14,C15	0.60	1.00
SCC19	C16,C17,C18,C19,C20	C16,C17,C18,C19,C20	1.00	1.00
SCC23	C21,C22,C23,C24,C25	C21,C22,C23,C24,C25	1.00	1.00
SCC26	C26,C27,C28,C29,C30	C26	0.20	1.00
SCC28	C26,C27,C28,C29,C30	C27,C28,C29,C30	0.80	1.00
SCC31	C31,C32,C33,C34,C35	C31	0.20	1.00
SCC33	C31,C32,C33,C34,C35	C32,C33,C34,C35	0.80	1.00
SCC37	C36,C37,C38,C39,C40	C36,C37	0.40	1.00
SCC38	C36,C37,C38,C39,C40	C38,C39,C40	0.60	1.00
SCC43	C41,C42,C43,C44,C45	C41,C42,C43,C44,C45	1.00	1.00
SCC48	C46,C47,C48,C49,C50	C48	0.20	1.00
SCC50	C46,C47,C48,C49,C50	C46,C47,C48,C49,C50	1.00	1.00
	Average		0.56	1.00

**Cluster Center Selection:**

Relevant Software Component is C1,C2,C3,C4,C5, then calculate the potential of all the software component by formula (4) and then stored in the matrix as below:

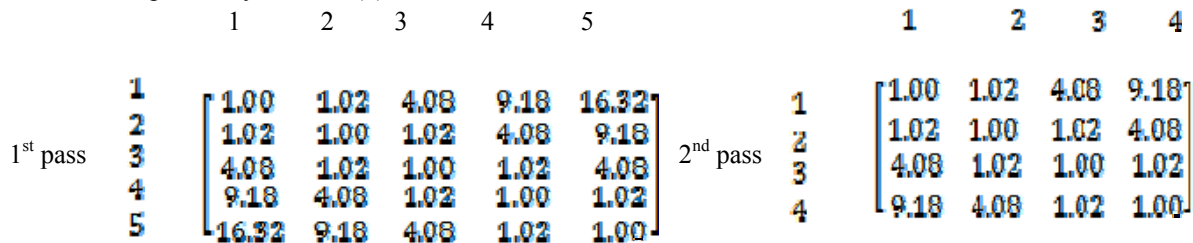


Fig.(a)

Fig. (b)

$P_{\max}(1.00,1.02,4.08,16.32) = 16.32$  means SCC5 as the first Cluster center as  $X_{c1}$ . Except SCC5 potential values of remaining Software Components (C1,C2,C3,C4) SCC3 is the second cluster center as  $X_{c2}$ . C2 and C4 does not becomes next cluster center.

In Similar way the remaining relevant software components produce cluster center as (C5 C8),(C11 C13 C15), C19, C23, (C26,C28),(C31 C33), ( C37 C38), C43),(C48 C50). Precision and Recall evaluated by formula (8) & formula (9).

Table 2. Recall and Precision Performance on the base of Rejection Ratio 0.34

Selected Cluster Center	Relevant Software Component	Retrieved Software Component	Recall	Precision
SCC3	C1,C2,C3,C4,C5	C1,C2,C3,C4,C5	1.00	1.00
SCC5	C6,C7,C8,C9,C10	C6,C7	0.40	1.00
SCC8	C6,C7,C8,C9,C10	C8,C9,C10	0.60	1.00
SCC11	C11,C12,C13,C14,C15	C11,C13	0.40	1.00
SCC15	S11,C12,C13,C14,C15	C13,C14,C15	0.60	1.00
SCC19	C16,C17,C18,C19,C20	C16,C17,C18,C19,C20	1.00	1.00
SCC23	C21,C22,C23,C24,C25	C23	0.20	1.00
SCC28	C26,C27,C28,C29,C30	C26,C27,C28,C29,C30	1.00	1.00
SCC33	C31,C32,C33,C34,C35	C31,C32,C33,C34,C35	1.00	1.00
SCC37	C36,C37,C38,C39,C40	C36,C37	0.40	1.00
SCC38	C36,C37,C38,C39,C40	C38,C39,C40	0.60	1.00

SCC43	C41,C42,C43,C44,C45	C41,C42,C43,C44,C45	1.00	1.00
SCC50	S46,C47,C48,C49,C50	C46,C47,C48,C49,C50	1.00	1.00
	Average		0.63	1.00

Table 3. 3 Recall and Precision Performance on the base of Rejection Ratio 0.40-0.50

Selected cluster center	Relevant Software component	Retrieved Software component	Recall	Precision
SCC3	C1,C2,C3,C4,C5	C1,C2,C3,C4,C5	1.00	1.00
SCC8	C6,C7,C8,C9,C10	C6,C7,C8,C9,C10	1.00	1.00
SCC15	C11,C12,C13,C14,C15	C11,C12,C13,C14,C15	1.00	1.00
SCC19	C16,C17,C18,C19,C20	C16,C17,C18,C19,C20	1.00	1.00
SCC23	C21,C22,C23,C24,C25	C23	0.20	1.00
SCC28	C26,C27,C28,C29,C30	C6,C26,C27,C28,C29,C30	1.00	0.83
SCC33	C31,C32,C33,C34,C35	C31,C32,C33,C34,C35	1.00	1.00
SCC38	C36,C37,C38,C39,C40	C36,C37,C38,C39,C40	1.00	1.00
SCC43	C41,C42,C43,SC44,C45	C41,C42,C43,C44,C45	1.00	1.00
SCC50	C46,C47,C48,C49,C50	C46,C47,C48,C49,C50	1.00	1.00
	Average		0.92	0.98

Table 4. Recall and Precision Performance Comparison

Rejection Ratio	Number of cluster selected	Recall	Precision
0.10-0.21	18	0.56	1.00
0.34	13	0.63	1.00
0.40-0.50	10	0.92	0.98

Table 5 shows the result of suitable software component (SC) selection on the bases of equation with different degree of significance on structural, functional and behavioral properties.

SC=Xreuse Equation.....(10)

values of reuse can be calculated from

$$\text{Reuse}=\arg \min_{1 \leq i \leq N_r} \left( \sum_{p=S,F,B} \Phi_p \|X_{pr} - X_{pi}\| \right) \text{ Equation.....(11)}$$

where pr and pi is retrieved potential and potential of i-th software component.

$\Phi_s, \Phi_f$  and  $\Phi_h$  be degree of structural, functional and behavioral properties.

Table 5. Software Component Selection with different Degree of Significance

Degree of Significance			Value of Each Software Component				Suitable Software Component for Reuse
Structural	Functional	Behavioral	C17	C18	C19	C20	
0.8	0.1	0.1	3.250	3.163	3.163	3.250	C18
0.1	0.8	0.1	3.250	3.163	3.163	3.250	C18
0.1	0.1	0.8	3.250	3.163	3.163	3.250	C19
0.3	0.3	0.3	2.925	2.846	2.846	2.925	C19

V. CONCLUSSION & DISCUSSION

Hybrid Approach (FSC & FCM) use to classify software components for reuse. This paper will use computational intelligence approach to classify software components repository into their respective cluster groups with the help of FCM & FSC algorithm. FCM is a common method used in many aspects. For example, image segmentation, preprocessing for data and pattern recognition. Before using FCM number of Cluster Centers required, calculated by FSC for optimal solution.

Fuzzy clustering algorithm could not assigned some of the software components into their respective cluster, but Fuzzy subtractive clustering algorithm assigned the outliers Software Component into their respective Cluster. In this paper, Hybrid approach classify the Software Component on the base of Precision and Recall performance by formula (8 & 9) and then evaluate the value of each Software Components by formula (10 & 11). The minimum value of Software Component select as suitable Software Component for reuse.

## REFERENCES

- [1] Samina Naz, Hammad Majeed, and Humayun Irshad, "Image Segmentation using Fuzzy Clustering : A Survey", [ieeexplore.ieee.org/iel5/5621662/5638364/05638492.pdf?arnumber...](http://ieeexplore.ieee.org/iel5/5621662/5638364/05638492.pdf?arnumber...)
- [2] Wen-Yuan Liu, Chun-Jing Xiao, Bao-Wen Wang, Yan Shi, and Shu-Fen Fang, "Study on Combining Subtractive Clustering with Fuzzy C-Means Clustering", in Proceedings of the Second International Conference on Machin Learning and Cybernetics, Xian, 2003.
- [3] Sathit Nakkrasae, Peraphon Sophatsathit, and William R. Edwards, "Fuzzy Subtractive Clustering Based Indexing Approach For Software Components Classification", International Journal of Computer & Information Science, Vol. 5, No. 1, 2004.
- [4] R.Suganya, and R. Santhi, " Fuzzy C-Means Algorithm-A Review", International Journal of Scientific and Research Publications, Volume 2, 2012.
- [5] Agus Priyono, Md. Ridwan, Ahmad Jais Alias, Riza Atiq O. K. Rahmat, Azmi Hassan and Mohd Alauddin Mohd. Ali, " Generation of Fuzzy Rules With Subtractive Clustering", in Journal Teknologi, 43 2005: .
- [6] M. S. Yang, "A Survey of Fuzzy Clustering", in Mthl. Compt. Modeling Vol.18,No.11,pp. 1- 16,1993 printed in great Britain.
- [7] Jianxiong Yang, and Junzo Watada, " Fuzzy Clustering ananalysis of data minig: Application to an Accident Mining Syatem", International Journal of Innovative Computing, Information and control Volume 8, Number 8, 2012.
- [8] Witold Pedrycz, Senior Member, IEEE, and James Waletzky, " Fuzzy Clustering With Partial Supervision", In IEEE Transactions on Systems, Man, and Cybernetics, Vol.27.
- [9] Soumi Ghosh, and Sanjay Kumar Dubey, " Comprative Analysis of K\_Means and Fuzzy C-Means Algorithm", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 4, 2013.
- [10] Boldeanu Silviu, "Fuzzy Clustering", in Babes-Bolayi University, Faculty of Compute Science. [www.cs.ubbcluj.ro/.../Clustering/FUZZY%20CLUSTERING%20Article%..](http://www.cs.ubbcluj.ro/.../Clustering/FUZZY%20CLUSTERING%20Article%..)