Middle-East Journal of Scientific Research 21 (7): 1097-1102, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.21.07.21566

Comparative Analysis of Clustering Techniques for Requirements Clustering

[1]Ananthi Sheshasayee, [2]P. Sharmila and [2]Quaid-E- Milleth

[1]Government College for Women, Chennai, Tamil Nadu, India
[2]Vivekanandha Institute Of Information and Management Studies,
Tiruchengode, Tamil Nadu, India

Abstract: Collection of requirements from the stakeholders plays a very important role in requirement engineering. As the stake holders vary from small groups to large this task becomes troublesome. Issues arise when the collected requirements are large such that the requirement engineers are not able to focus on the desired requirements. As the stakeholders expertise varies from stakeholder to another, various elicitation methods are needed to be adopted in collecting the requirements. This paper focus in classifying the valid requirements from the set of collected requirements using clustering techniques. In this research two most frequently used algorithms in clustering namely k means and fuzzy c means are used. The output generated is then analyzed for evaluating the performance of the two clustering algorithms. On analysis the fuzzy c means algorithm was found to be more suitable for clustering of library requirements. The results proved to be satisfactory. Since there are large number of requirements collected from stakeholders clustering the requirements reduces the software development and maintenance to large extent.

Key words: Requirements · Stakeholders · Data Mining · Clustering · Fuzzy C Means · K Means

INTRODUCTION

Requirement engineering is the most essential part of software engineering which ensures the overall quality of software product. Errors produced and undetected at this stage will prove costly at the later stage of software development. Requirement engineering activities include elicitation, analysis and validation and documentation of the collected requirements. These tasks individually contribute to the overall quality of the software The requirements elicited from the stakeholders are entered in the software requirement specification (SRS) [1-10]. Hence requirement specification should be correctly collected. The ability of the requirement engineers in eliciting the requirements mainly depends upon the stakeholder's participation. Web based elicitation techniques such as forums, wikis and online survey are used now to elicit the requirements from a large number of stakeholders [11, 12].

During the requirement elicitation, significant effort is needed in discovering and understanding the requirements from large number of stakeholders. The number of stakeholders in modern projects are too large thereby causing problem in identification and managing project requirements [13-15]. As the knowledge of the stakeholders vary to a different extent care should be taken in choosing the elicitation methods while collecting the requirements. One of the biggest problems with requirements engineering is that the stakeholders may be numerous and distributed with varying and conflicting goals [16]. This has resulted in their requirement sets also growing in size, complexity and type [6].

Web-based toolset helps requirements engineers to identify project stakeholders, elicit product requirements and stakeholders' preferences for these requirements by asking stakeholders to recommend other stakeholders, propose new requirements and rate already submitted requirements [14]. Careful evaluation and prediction of valid requirements is necessary for a good SRS. This task when related to very large projects, are in need of automated support. The problem is how to automatically and efficiently coordinate large numbers of stakeholders' requests and to arrange the subsequent requirements into meaningful structures.

Corresponding Author: Ananthi Sheshasayee, Government College for Women, Chennai, Tamil Nadu, India.

Literature Review: The collected requirements from the stakeholders are decomposed based on the common characteristic shared by the requirements using clustering technique. Thus the grouped requirements share certain common properties. In [10] a software automate is constructed for requirement clustering using natural language parser and hierarchical clustering techniques. Certain requirement clustering encapsulates the requirements by clustering the requirements using similarity and association relations and then encapsulate each cluster by defining external interface as stimulus response pair [14]. Problem of document clustering requirements is addressed through surveying standard clustering techniques in [4] and their application to the requirements clustering process is discussed. Based on [15-17] clustering techniques applied for requirement encapsulation reduces the total cost of software over its entire life cycle by 16-30 percent depending on the extent to which the software requirement reuse is deployed. Collected requirements are prioritized based on factors like cost value and weights assigned to stakeholders group. Cost-value based requirements prioritization techniques rely on eliciting the relative costs and value of each requirements for each stakeholders group [8]. Weights are also assigned to all stakeholders groups, by which we can compute the overall value of a requirement as the weighted sum of its value for each stakeholders group and rank the set of requirements accordingly. Different variants of this approach are used in practice [9, 18].

The usage of clustering techniques for requirements engineering is a growing research area now-a-days. Cluster analysis is an important data mining technique used to find data segmentation and pattern information. By clustering the data, people can obtain the data distribution, observe the character of each cluster and make further study on particular clusters. In addition, cluster analysis usually acts as the pre-processing of other data mining operations [7]. Therefore cluster analysis has become a very active research topic in data mining.

Clustering is a group of physical or abstract objects, divided into several groups according to the degree of similarity between them, and makes the same data objects within a group of high similarity and different groups of data objects are not similar [3, 21]. Applications of clustering techniques are used in identifying homogenous groups of stakeholders that can be used as input to existing requirements selection and prioritization techniques [19]. The choice of application of a particular method generally depends on the type of output desired,

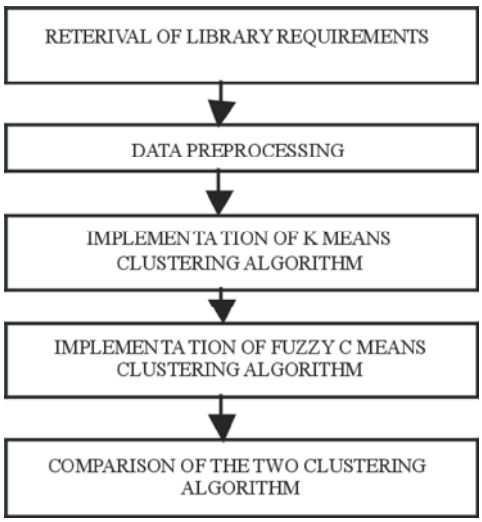

Fig. 1: Proposed Methodology For Requirement Clustering On Applying Clustering Techniques.

the known performance of the method with particular type of data, available hardware and software facilities and size of the dataset [17].

Proposed Work: Proposed method uses fuzzy c means and k means clustering algorithm to group the library requirements. The requirements are grouped on the similarity measures calculated. We use cosine and vector computation as similarity measures for the clustering techniques. Since there are large number of requirements collected from stakeholders clustering the requirements reduces the software development and maintenance to large extent. The work has been carried out in WEKA. The Waikato Environment for Knowledge Analysis (WEKA) 3.60 serves as an intelligent tool for data analysis and predictive modelling. WEKA was chosen for its wide collection of free analytical tools and data mining algorithms. The following figure describes about the proposed methodology for the research work in this paper.

MATERIAL AND METHODS

Requirements clustering techniques address the relationship between requirements. Requirements clusters contribute to requirements reuse, but not sufficient for design and code reuse.

Data Preparation: The well known UCI Machine Learning Repository is used and it is actually a collection of databases which is widely used by the researchers of

Machine Learning, especially for the empirical algorithms analysis of this discipline [1]. The requirements specified by the stakeholders of the library are taken as the data set. The data set contains five attributes namely stake-id, Requirement, specific requirements and role.

Fuzzy C Means Algorithm: Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1973 [5] and improved by Bezdek in 1981[2]) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \| x_i - c_j \|^2, 1 \le m < \infty \quad (1)$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster and ||*|| is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k-1}^{C}\left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|}\right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i-1}^{N} u_{ij}^{m} . x_i}{\sum_{i-1}^{N} u_{ij}^{m}} \quad (2)$$

This iteration will stop when $\max_{ij}\left\{|u_{ij}^{(k+1)} - u_{ij}^{k}|\right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas k is the iteration steps. This procedure converges to a local minimum or a saddle point of J_m.

The algorithm is composed of the following steps:

- Initialize $U=[u_{ij}]$ matrix, $U^{(0)}$
- At k-step: calculate the centers vectors $C^{(k)}=[c_j]$ with $U^{(k)}$

$$c_j = \frac{\sum_{i-1}^{N} u_{ij}^{m} . x_i}{\sum_{i-1}^{N} u_{ij}^{m}} \quad (3)$$

- Update $U^{(k)}$, $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k-1}^{C}\left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|}\right)^{\frac{2}{m-1}}} \quad (4)$$

- If $\|U^{(k+1)} - U^{(k)}\| < \varepsilon$ then STOP; otherwise return to step 2.

K Means Algorithm: K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy 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 centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place 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 centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenters 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 centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more [19-21].

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2 \quad (4)$$

where $|x^{(j)}_i - c_j|^2$ is a chosen distance measure between a data point $x^{(j)}_i$ and the cluster centre $_j$,
c_j is an indicator of the distance of the n data points from their respective cluster centres.

The algorithm is composed of the following steps:

- Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
- Assign each object to the group that has the closest centroid.

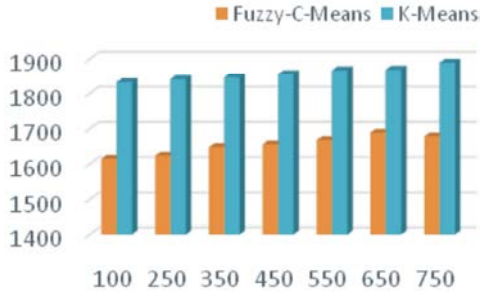

Fig. 2: Time comparison of fuzzy c means and k means algorithm

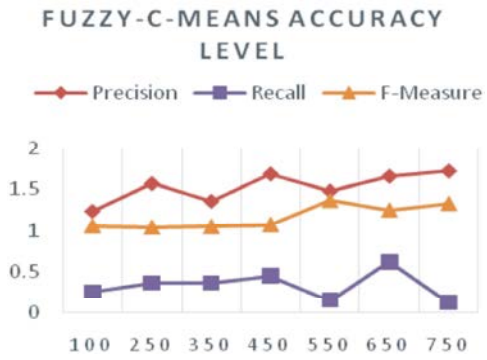

Fig. 3: Precision, Recall And F Measure Values Of Fuzzy C Means Algorithm

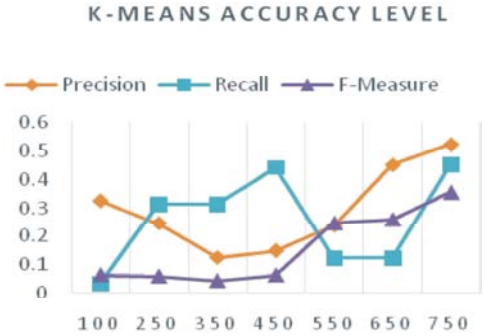

Fig. 4: Precision, Recall And F Measure Values Of K Means Algorithm

- When all objects have been assigned, recalculate the positions of the K centroids.
- Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Experimental Results and Analysis: The number of clusters is given by the user during the execution of the program. The output generates four clusters: cluster1, cluster2, cluster3 and cluster4.The cluster quality is evaluated using the following factors.

Table 1: Execution time in milli seconds

Number of Records	Fuzzy-C-Means	K-Means
100	1614	1832
250	1623	1841
350	1648	1846
450	1656	1856
550	1668	1867
650	1689	1869
750	1678	1889

Table 2: Precision, Recall, F values for Fuzzy C Means Algorithm

Number of Records	Precision	Recall	F-Measure
100	1.23546	0.25896	1.05493
250	1.56983	0.36581	1.03658
350	1.35647	0.36592	1.04568
450	1.68523	0.45236	1.069852
550	1.47892	0.15632	1.36258
650	1.65823	0.62315	1.245698
750	1.72153	0.12365	1.32658

Table 3: Precision, Recall, F values for k Means Algorithm

Number of Records	Precision	Recall	F-Measure
100	0.325441	0.032589	0.06312
250	0.245634	0.312444	0.05863
350	0.123485	0.312454	0.04256
450	0.147895	0.442323	0.06245
550	0.236985	0.123123	0.24563
650	0.452173	0.123123	0.25858
750	0.523698	0.452392	0.35435

- Execution time,
- f measure
- Precision
- Recall.
- Distribution of the requirements in the clusters

Table 2 shows the execution time taken by fuzzy c means and k means algorithm with increase in the number of records. Though the execution time increases with the increase in number of records the time consumption for fuzzy c means is less compared to the K-Means.

The following figure shows the execution time comparison of fuzzy c means and k means algorithm. Here the x-axis represents the number of records and the y-axis represents the time in milliseconds.

Table 2 shows the precision values, recall values and f measure values which evaluate the cluster accuracy using fuzzy c means algorithm.

In the following graph, the x-axis represents the number of records and the y-axis represents the accuracy values.

The following table shows the precision values, recall values and f measure values which evaluate the cluster accuracy using k means algorithm.

Table 4: Distribution of requirements in clusters

Clustering algorithm	Cluster1	Cluster2	Cluster3	Cluster4
Fuzzy c means	201	145	225	223
K means	79	50	291	362

Table 5: Distribution Of Requirements In Each Cluster Using Fuzzy C Means

Requirements that	No of Requirements
Occurred in Both 1&2	65
Occurred in Both 1&3	85
Occurred in Both 1&4	60
Occurred in Both 2&1	59
Occurred in Both 2&3	58
Occurred in Both 2&4	70
Occurred in Both 3&1	89
Occurred in Both 3&2	53
Occurred in Both 3&4	55
Occurred in Both 4&1	72
Occurred in Both 4&2	63
Occurred in both 4&3	65
Occurred in 1,2,3 &4	15

In the following graph, the x-axis represents the number of records and the y-axis represents the accuracy values

Also since k means algorithm is hard where the data points falls exactly in only in one cluster, fuzzy c means allows the data points to group in more than one clusters. Thus on finding the similarity measures the library requirements group in more than one clusters.

The following table shows the number of requirements that appear in more than one clusters On analyzing the distribution of requirements it is found that the requirements that appear in one cluster also appears in other clusters also depending upon the similarity measures.. These requirements can be used in identifying the valid requirements and the stakeholders who specify it. Further it can also be used for prioritizing the requirements which greatly reduces the software development time and cost.

CONCLUSION

This study was proposed to study the similarities between the requirements and group them into clusters using fuzzy c means and k means and also compare the performance of both the algorithms. The library data set specified by various stakeholders is used. After analysing fuzzy c means and k means algorithm we conclude the following results.

The execution time speed of the fuzzy c means algorithm is better than k means algorithm thus the performance of fuzzy c means is higher compared to the k means. As the number of records increases, the time execution of both the technique gets increased but the fuzzy c means performance is found to be better than the k means algorithm. The precision, recall and f measure values measure more accuracy on applying fuzzy c means compared to k means algorithm. From the experimental results it is concluded that fuzzy c means algorithm is efficient for larger data set and is well suited for requirement clustering. Stakeholders grouping can be done by analysing the common requirements that appears in all the clusters using fuzzy c means algorithm Future work may consider the stakeholders priorities of requirements during requirement elicitation and the requirement clustering performance using various other clustering techniques can be compared. Further the requirements thus grouped using clustering may be used for software modernization, requirements re-use and software requirement improvement.

REFERENCES

1. Asuncion, A. and D.J. Newman, 2013. UCI Machine Learning Repository Irvine, CA: University of California, School of Information and computer science.
2. Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA.
3. Daniel, B.A., 2003. Ping Chen Using Self-Similarity to Cluster Large Data Sets [J].Data Mining and knowledge Discovery, 7(2): 123-152.
4. Duan, C., 2008. Clustering and its Application in Requirements Engineering, Technical Report #08-001, School of Computing, (DePaul University,).
5. Dunn, J.C., 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Cybernetics and systems, 3: 32-57.
6. Firesmith, D., 2003. Modern requirement specification. Journal of Object Technology, 2(2): 53-64.
7. GuedaliaI, D.C.M. London and M. Werman, 1999. An on-line agglomerative clustering method for non-stationary data, NeuralComputation, 11: 521 540.
8. Karlsson, J. and K. Ryan, 1997. A cost-value approach for prioritizing requirements, IEEE software, 14: 67-74.
9. Karl Wiegers, E., 1999. First things first: Prioritizing requirements. Software Development, 7(10): 24-30. King, Margaret, 1996. Evaluating natural language processing systems. Communications of the ACM, 39(1): 73-79.

10. King Margaret, 1996. Evaluating natural language processing systems. Communications of the ACM, 39(1): 73-79.
11. Laurent, P. and J. Cleland-Huang, 2009. Lessons Learned from Open Source Projects for Facilitating Online Requirements Processes, In: Glinz, M., Heymans, P. (eds.) REFSQ 2009. LNCS, 5512: 240-255. Springer, Heidelberg.
12. Lim, S.L., D. Quercia and A. Finkelstein, Stake Net: using social networks to analyse the stakeholders of large-scale software projects, in Proc. ICSE 2010, 1: 295-304.
13. Lim, S.L., 2010. Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation. PhD Thesis. School of Computer Science and Engineering, University of New South Wales.
14. Zude, Li, Quazi Abidur Rahman and Nazim H. Madhavji, 2007. An Approach to Requirements Encapsulation with Clustering. In WER, pp: 92-96.
15. Lutowski, R., 2005. Software Requirements: Encapsulation, Quality and Reuse, Aurbach Publisher,
16. Parsons-Hann, H. and K. Liu, 2005. Measuring requirements complexity to increase the probability of project success. ICEIS, 3: 434-438.
17. Rao, V.S. and Dr. S. Vidyavathi, 2010. Comparative Investigations and Performance Analysis of FCM and MFPCM Algorithms on Iris Data, Indian Journal of Computer Science and Engineering, 1(2): 145-151.
18. Robertson, S. and J.C. Robertson, 2006. Mastering the Requirements Process, Addison Wesley.
19. Veerappa, V. and E. Letier, 2011. Clustering stakeholders for requirements decision making. In: Berry, D. and Franch, X. (Eds.) Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science, Berlin, Springer Heidelberg, 6606: 202-208.
20. Villegas, O.L., M. Angel Laguna and F.J. Garcia, 2002. Reuse based analysis and clustering of requirements diagrams. In Proc. of Int'l Workshop Conf. on Requirements Engineering: Foundation for Software Quality (REFSQ'02).
21. Wei Chi-Ping, Lee Yen-Hsien□C Hsu Che Ming, 2003. Empirical comparison of fast Partitioning-based clustering algorithms for Large data sets, Expert Systems with Applications, 24(4): 351-363.