Estimating of Software Quality with Clustering Techniques

Deepak Gupta
Research Scholar, S.G.V.U., Jaipur, INDIA
deepakgupta.vce@gmail.com

Vinay Kr. Goyal
Director, HIT, Asodha, Bahadurgarh, INDIA
vinayeq@yahoo.com

Harish Mittal
Director, SPGOI, Rohtak, INDIA
mittalberi@yahoo.co.in

Abstract - **Software faults are one of major criteria to estimate the software quality or the software reliability. There is number of matrices defined that uses the software faults to estimate the software quality. When we have a large software system with thousands of class modules, then it is not easy to apply the software matrices on each module of software system. The present work is the solution of the defined problem. This paper aims at comparing different models based on clustering techniques: k-means (KM), fuzzy c-means (FCM) and hierarchical agglomerative clustering (HAC) for building software quality estimation system. We propose quality measure of partition clustering technique (KM, FCM) in order to evaluate the results and we comparatively analyze the obtained results on two case studies. This paper focuses on clustering with very large datasets and very many attributes of different types.**

Keywords: **Clustering, K-means, Fuzzy c-means, Hierarchical agglomerative.**

I. INTRODUCTION

A. What is Clustering?

Clustering [10], [11] is the assignment of a set of observations into subsets (called clusters). Clustering is a division of data into groups of similar objects. Each group (called cluster) consists of objects that are similar between themselves and dissimilar to objects of other groups. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. The underlying software engineering assumption is that the fault-prone software modules will have similar software measurements, and hence are likely to be grouped together in the same cluster(s). Similarly, the not fault-prone modules will likely be grouped in the same cluster(s) [2].

Fig. 1 shows the 4 clusters into which the data can be divided; the similarity criterion is distance: two or more objects belong to the same cluster if they are "close" according to a given distance. This is called distance-based clustering.

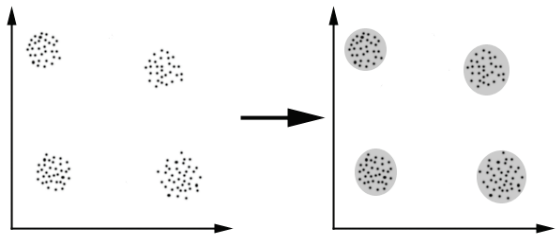


Figure 1. Four clusters

Another kind of clustering is conceptual clustering: two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures [4].

B. What can be Clustered?

Images (astronomical data), patterns (robot vision data), shopping items, feet (anatomical data), words, documents and so on.

C. Why Clustering?

- Improving precision and recall in information retrieval.
- Browsing a collection of documents for purposes of information retrieval (scatter and gatherer strategy).
- Organizing the results provided by the search engine.
- Generation of document taxonomies (cf. YAHOO!).
- Generation of sub categorization frame classes.

D. Possible Applications

Clustering algorithms can be applied in many fields, for instance [5]:

1) Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records.

2) Biology: classification of plants and animals given their features.

3) Libraries: book ordering.

4) *Insurance:* identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds.

978-0-7695-4941-5/13 $26.00 © 2013 IEEE
DOI 10.1109/ACCT.2013.83

5) City-planning: identifying groups of houses according to their house type, value and geographical location.

6) Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones.

7) WWW: document classification; clustering web log data to discover groups of similar access patterns.

E. Requirements of Clustering

The main requirements that a clustering algorithm should satisfy are [1]:

- scalability;
- dealing with different types of attributes;
- discovering clusters with arbitrary shape;
- minimal requirements for domain knowledge to determine input parameters;
- ability to deal with noise and outliers;
- insensitivity to order of input records;
- high dimensionality;
- Interpretability and usability.

F. Problems of Clustering

There are a number of problems with clustering. Among them [1]:

- current clustering techniques do not address all the requirements adequately (and concurrently);
- dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
- the effectiveness of the method depends on the definition of "distance" (for distance-based clustering);
- if an obvious distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;
- the results of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

II. RELATED WORK

Many clustering techniques have been proposed so far for building software quality estimation system.

Serban *et al*. [22] used clustering techniques for identifying crosscutting concerns. The comparison was made mostly from the aspect mining point of view, using a set of quality measures (SSE, PAM, ACC).

Berkhin [12] focused on clustering algorithms from a data mining perspective. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems.

Xu *et al.* [22] focused on clustering algorithms and review a wide variety of approaches appearing in the literature. These algorithms evolve from different research communities, aim to solve different problems, and have their own pros and cons.

Gupta et al. [24] proposed a comparative evaluation of predictive accuracy of six commonly used software quality prediction modeling techniques: Classification & Regression Trees, Multiple Linear Regression, Artificial Neural Networks, Case-Based Reasoning, Rule-Based systems, Fuzzy Systems. It was concluded that fuzzy systems yielded better results than other techniques.

Gupta et al. [25] used clustering techniques to build a software quality estimation system. Software metrics thresholds are used to remove the expert necessity. In this paper first applied K-means and Fuzzy C-means clustering method to cluster hundreds of software modules into a small number of coherent groups. After this step, a cluster was predicted as fault-prone if at least one metric of the mean vector is higher than the threshold value of that metric. And then measured the quality of each cluster based on fault prone or not fault prone.

Lu *et al.* [23] proposed a new clustering algorithm called Fast Genetic K-means Algorithm (FGKA). FGKA is inspired by the Genetic K-means Algorithm (GKA) proposed by Krishna and Murty in 1999 but features several improvements over GKA. Experiments indicate that, while K-means algorithm might converge to a local optimum, both FGKA and GKA always converge to the global optimum eventually but FGKA runs much faster than GKA.

III. DISTANCE MEASURE

An important step in most clustering is to select a distance measure, which will determine how the similarity of two elements is calculated [3]. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point (x = 1, y = 0) and the origin (x = 0, y = 0) is always 1 according to the usual norms, but the distance between the point (x = 1, y = 1) and the origin can be 2, or 1 if you take respectively the 1-norm, 2-norm or infinity-norm distance. Common distance functions [1], [5]:

A. Euclidean Distance

Euclidean Distance is the most common use of distance. Euclidean distance or simply 'distance' examines the root of square differences between coordinates of a pair of objects.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk} \right)^2}$$

B. Minkowski Distance

This is the generalized metric distance. When $\lambda = 1$ it becomes city block distance and when $\lambda = 2$, it becomes Euclidean distance. Chebyshev distance is a special case of Minkowski distance with $\lambda = \infty$ (taking a limit). This distance can be used for both ordinal and quantitative variables.

$$d_{ij} = \sqrt[\lambda]{\sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|^{\lambda}}$$

C. Mahalanobis Distance

It is also called quadratic distance. It measures the separation of two groups of objects. Suppose we have two groups with means $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$, Mahalanobis distance is given by the following formula:

$$d_{ij} = \left(\left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j \right)^T \mathbf{S}^{-1} \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j \right) \right)^{\frac{1}{2}}$$

D. City Block Distance

It is also known as Manhattan distance, boxcar distance, absolute value distance. It represents distance between points in a city road grid. It examines the absolute differences between coordinates of a pair of objects.

$$d_{ij} = \sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|$$

E. Chebyshev Distance

Chebyshev distance is also called Maximum value distance. It examines the absolute magnitude of the differences between coordinates of a pair of objects. This distance can be used for both ordinal and quantitative variables.

$$d_{ij} = \max_{k} \left| x_{ik} - x_{jk} \right|$$

F. Hamming Distance

Finite binary 0 and 1 sequence is sometimes called a word in coding theory. If two words have the same length, we can count the number of digits in positions where they have different digit. The length of different digits is called Hamming distance. If q= number of variables with value 1 for the i^{th} objects and 0 for the j^{th} object and r = number of variables with value 0 for the i^{th} objects and 1 for the j^{th} object, we have $d_{ij} = q + r$.

G. Correlation Coefficient

Correlation coefficient is standardized angular separation by centering the coordinates to its mean value. The value is between -1 and +1. It measures similarity rather than distance or dissimilarity

$$s_{ij} = \frac{\sum_{k=1}^{n} \left(x_{ik} - \bar{x}_i \right) \cdot \left(x_{jk} - \bar{x}_j \right)}{\left(\sum_{k=1}^{n} \left(x_{ik} - \bar{x}_i \right)^2 \cdot \sum_{r=1}^{n} \left(x_{jk} - \bar{x}_j \right)^2 \right)^{\frac{1}{2}}},$$

where $\bar{x}_i = \frac{1}{n} \sum_{k=1}^{n} x_{ik}$ and $\bar{x}_j = \frac{1}{n} \sum_{k=1}^{n} x_{jk}$

IV. BASIC CLUSTERING TECHNIQUES

Two types of clustering techniques: Hierarchical and Partitional [2], [6], 10], [12].

A. Hierarchical Clustering

Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure called a dendrogram. The root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations. They are either agglomerative (bottom-up) or divisive (top-down) [6], [7] as shown in Fig. 2:

1) Agglomerative algorithms: start at the leaves with each object being a separate cluster itself, and successively merge groups according to a distance measure [6]. The clustering may stop when all objects are in a single group or at any other point the user wants.

2) Divisive algorithms: They start at the root with one group of all objects and successively split groups into smaller ones, until each object falls in one cluster, or as desired. Divisive approaches divide the data objects in disjoint groups at every step, and follow the same pattern until all objects fall into a separate cluster [13]. This is similar to the approach followed by divide-and-conquer algorithms.

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering

- Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N. (*).

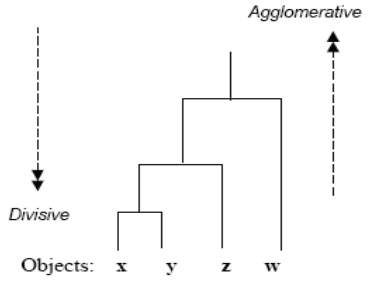


Figure 2. Agglomerative hierarchical clustering

Step 3 can be done in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering.

- *In single-linkage clustering:* (also called the connectedness or minimum method), we consider the

distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster [14].

$$\min\{\, d(x, y) : x \in \mathcal{A},\ y \in \mathcal{B} \,\}.$$

- *In complete-linkage clustering* (also called the diameter or maximum method); we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster [15].

$$\max\{\, d(x, y) : x \in \mathcal{A},\ y \in \mathcal{B} \,\}.$$

- *In average-linkage clustering:* we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y).$$

a) Single-Linkage Clustering: The Algorithm

The N*N proximity matrix is D = [d (i,j)]. The clustering's assigned sequence numbers 0, 1,....., (n-1) and L(k) is the level of the k^{th} clustering. A cluster with sequence number m is denoted (m) and the proximity between clusters (r) and (s) is denoted d [(r), (s)]. The algorithm is composed of the following steps [14]:

- Begin with the disjoint clustering having level L (0) = 0 and sequence number m = 0.
- Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to d [(r), (s)] = min d [(i), (j)] where the minimum is over all pairs of clusters in the current clustering.
- Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L (m) = d [(r), (s)]
- Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way: d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]
- If all objects are in one cluster, stop. Else, go to step 2.

b) Advantages of hierarchical clustering:

- Flexibility regarding the level of granularity.
- Ease of handling any form of similarity or distance.
- Applicability to any attributes types.

c) Disadvantages of hierarchical clustering:

- Vagueness of termination criteria.
- Most hierarchical algorithms do not revisit (intermediate) clusters once constructed.

Fig. 3 shows the hierarchical agglomerative clustering.

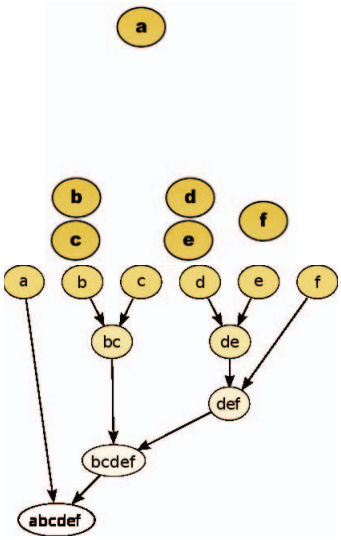


Figure 3. Agglomerative hierarchical clustering

B) Partitional Clustering

Given a database of n objects, a partitional clustering algorithm constructs k partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster. Types of Partitional clustering:

1) K-means clustering:

One of the simplest clustering algorithms is K-means clustering method. The k-means algorithm assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster — that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster [8], [9].

a) The algorithm steps are [16], [19]:

- Choose the number of clusters, k.
- Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers.
- Assign each point to the nearest cluster center, where "nearest" is defined with respect to one of the distance measures discussed above.
- Recompute the new cluster centers.
- Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed).

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

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

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers.

b) Advantages of k-means:

- It is easy to implement and works with any of the standard norms.
- It allows straightforward parallelization.
- It is insensitive with respect to data ordering.
- High speed which allows it to run on large datasets.

c) Disadvantages:

- The results strongly depend on the initial guess of centroids.
- A local optimum (computed for a cluster) does not need to be a global optimum (overall clustering of a data set).
- It is not obvious what a good number k is in each case.
- Resulting clusters can be unbalanced (even empty).

d) Algorithm to initialize the centoids:

To overcome the problem of initial guess of centroids we are using an algorithm that is to find the dimension with maximum variance, sorting it, dividing it into a set of groups of data points then finding the median for each group, using the corresponding data points (vectors) to initialize the kmeans. The method works as follows.

- For a data set with dimensionality, *d*, compute the variance of data in each dimension (column).
- Find the column with maximum variance; call it *cvmax* and sort it in any order.
- Divide the data points of *cvmax* into *K* subsets, where *K* is the desired number of clusters.
- Find the median of each subset.
- Use the corresponding data points (vectors) for each median to initialize the cluster centers.

Fig. 4 shows the K-means algorithm. Training examples are shown as dots, and cluster centroids are shown as crosses. (a) Original dataset. (b) Initial cluster centroids. (c-f) Illustration of running two iterations of k-means.

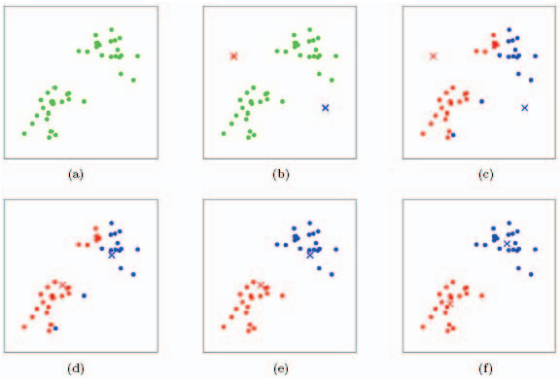


Figure 4. K-means algorithm.

2) Fuzzy C-means clustering:

Fuzzy c-means clustering method was developed by Bezdek [18], [20]. Each instance can belong to every cluster with a different membership grades between 0 and 1 for this algorithm [17]. A dissimilarity function, as in (2), is minimized and centroids which minimize this function are identified.

$$J(U,V|Z) = \sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^m \left\| x_j - c_i \right\|_A^2 \qquad (2)$$

In this equation, $V = \{y_1,.....y_c\}$ is a vector of cluster prototypes (centers) and m is a constant. Also, we have

$$D_{ijA}^2 = \left\| x_j - c_i \right\|_A^2 = (x_j - c_i)^T A (x_j - c_i).$$

If A = I, then we are using a Euclidian norm. The shape of the clusters will be circular. If we are using the Mahalanobis norm then A is defined, as in (3). The shape of the clusters will be ellipsoidal [21].

$$A = \left(\frac{1}{N}\sum_{j=1}^{N}(x_j - \bar{x})(x_j - \bar{x})^T \right)^{-1} \qquad (3)$$

a) The algorithm steps are:

- Initialize the membership function randomly, as in (4).
- Calculate centroids, as in (5).
- Calculate dissimilarity value, as in (2). Stop, if the improvement compared to previous iteration is below a threshold level.
- Calculate a new u, as in (6). Go to step 2.

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,.....n \qquad (4)$$

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{\ m} x_j}{\sum_{j=1}^{n} u_{ij}^{\ m}} \quad (5)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{D_{ij}}{D_{kj}} \right)^{2/(m-1)}} \quad (6)$$

c_i is i^{th} cluster's centroid, u is between 0 and 1, D_{ij} is the Euclidean distance between centroid and the data point, m is a weighting exponent which is between 1 and ∞.

Fig. 5 shows the fuzzy clustering. Training examples are shown as dots, and cluster centroids are shown as *, +, ◯

b) Advantages of fuzzy c-means

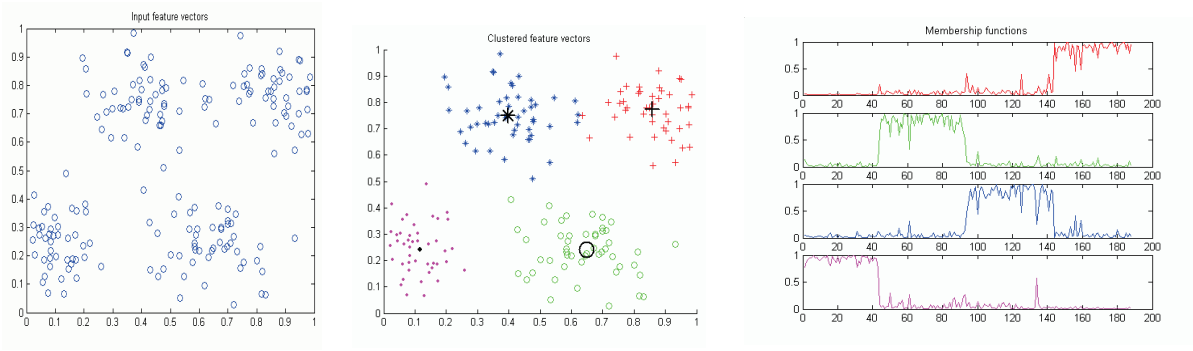
- Unsupervised
- Always converges

c) Disadvantages of fuzzy c-means

- Long computational time.
- Sensitivity to the initial guess (speed, local minima).
- Sensitivity to noise: One expects low (or even no) membership degree for outliers (noisy points).

IV. COMPARISION OF CLUSTERING TECHNIQUES

The algorithms discussed so far are given in Table 1. together with some of their characteristics. This table indicates the input parameters required by each algorithm (2nd column), the type of data sets it is optimized for (3rd column), the capability of tackling high dimensional data or not (4^{th} column), the cluster structure (5th column), whether it handles noise or not (6th column) and its computational complexity (7th column).



(a) Original dataset x =188, d =2.

(b) Cluster centroids and c =4

(c) Membership Functions

Figure 5. Fuzzy c-means algorithm

TABLE I. COMPARISION of CLUSTERING TECHNIQUES

Algorithm	Input Parameters	Optimized for	Capability of tackling high dimensional data	Cluster Structure	Outlier Handling	Computational Complexity
Hierarchical	Number of clusters, Branching Factor, Diameter Threshold	Large data sets	No	Spherical	Yes	$O(N^2)$ (time) $O(N^2)$ (space)
K-means	Number of clusters	Separated clusters	No	Spherical	No	O(NKd) (time) O(N + K) (space)
Fuzzy c-means	Number of clusters	Clusters with different membership function	No	Spherical, Ellipsoidal	No	Near O(N)

V. EXPERIMENTAL STUDY

A. System Description

The empirical case study presented is that of two NASA software project written in the C++ programming language, labeled JM1, KC3. The software measurements and quality data was obtained through the NASA Metrics Data Program

(http://mdp.ivv.nasa.gov). The software metrics and fault data for the project was collected at the program's function or subroutine levels. In JM1, KC3, some modules with the same attribute values had different defect labels. When we removed these inconsistent software modules and those with missing values from JM1, 8916 modules remained. We labeled the reduced JM1 data set JM1-8916. The KC3 data set has 314 software modules, and is labeled KC3-314. The 15 metrics used in our experiments as input feature vectors as shown in Table 2.

B. Experimental Setting

We are using both K-means and Fuzzy c-means clustering implementation locating in MATLAB. We have developed some MATLAB programs to evaluate the overall performance of these algorithms. For our data sets, the k-means and fuzzy c means algorithm always converged within 100 iterations. We set the number of clusters K (a parameter needed by both the k-means and Fuzzy c-means algorithms) to 30 and 15 for JM1-8916 and KC3-314 respectively.

TABLE II. SOFTWARE METRICS

Branch count metric		Branch_Count
Error Count Metric		Error Density
Line count metrics		Total_Lines_of_Code Executable_LOC Comments_LOC Blank_LOC Code_&_Comments_LOC
Halstead metrics		Halstead Error Estimate Total_Operators Total_Operands Unique_Operators Unique_Operands
McCabe metrics		Cyclomatic_Complexity Essential_Complexity Design_Complexity

C. Clustering Quality

For clustering quality, we use the objective function of both k-means & fuzzy c-means, as in (1) and (2) respectively and average purity (ave-pur). It can be seen as an optimization process that aims to minimize the objective function of any one technique then the quality of clusters of that technique is better as compared to other one. The purity of a cluster is defined as the percentage of the most dominated category (fault-prone or not fault-prone) in the cluster, and average purity is the mean over all clusters. It has a range of between 0 and 1; the higher the number, the better is the average purity.

D. Clustering and Quality Estimation Results

The K-means and Fuzzy c-means based approach divides data points into k clusters. The clustering quality results are shown in Table 3. The Fuzzy c-mean algorithm performed significantly better in terms of objective function then K-means for both data sets. The fuzzy c-mean algorithms performed slightly worse for JM1-8916 and slightly better for KC3-314 in terms of average purity.

TABLE III. CLUSTERING RESULTS

Data set	Technique	Objective Function	Average Purity
JM1-8916	K-means	3615.40	0.7747
	Fuzzy c -means	776.92	0.7523
KC3-314	K-means	831.16	0.8613
	Fuzzy c -means	297.55	0.8726

VI. CONCLUSION AND FUTURE WORK

In this paper we have analyzed three clustering techniques and we have comparatively presented the results of applying two clustering algorithms (k-means & Fuzzy c-means) and effective results can be produced by using Fuzzy c-means clustering. We conclude the survey with listing some important issues and research trends for cluster algorithms.

- There is no clustering algorithm that can be universally used to solve all problems.
- New technology has generated more complex and challenging tasks, requiring more powerful clustering algorithms.
- At the preprocessing and post-processing phase, feature Selection/extraction is as important as the clustering algorithms.

Further work can be done as to compare the results obtained by the clustering algorithms proposed in this paper with other clustering approach that was proposed in the literature (such as hierarchical clustering).

REFERENCES

[1] J. Han and M. Kamber, *"Data Mining: Concepts and Techniques"*. Morgan Kaufmann Publishers, 2001.

[2] A. Jain, M. N. Murty, and P. Flynn, *"Data clustering: A review"*. ACM Computing Surveys, vol. 31, no. 3, pp: 264-323, 1999.

[3] G. S. Moldovan and G. Serban, "*Quality Measures for Evaluating the Results of Clustering Based Aspect Mining Techniques*". In Proceedings of *Towards Evaluation of Aspect Mining(TEAM), ECOOP*, 2006, to be published.

[4] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim, "*On the surprising behavior of distance metrics in high dimensional space*". In Proceedings of the 8th International Conference on *Database Theory (ICDT)*, pp: 420–434, London, UK, 2001.

[5] Michael R. Anderberg "*Cluster Analysis and Applications*". Academic Press, New York, NY, USA, 1973.

[6] A. K. Jain and R. C. Dubes, *"Algorithms for Clustering Data"*. Prentice-Hall, Englewood Cliffs, NJ, 1988.

[7] L. Kaufman and P.J. Rousseeuw, *"Finding Groups in Data: An Introduction to Cluster Analysis"*. Wiley, New York, 1990.

[8] J. Hartigan and M. Wong, *"Algorithm as136: A k-means clustering algorithm"*. Applied Statistics, vol. 28, pp: 100–108, 1979.

[9] J. A. Hartigan, *"Clustering Algorithms"*. Wiley, 1975.

[10] B. Everitt, S. Landau, and M. Leese, "*Cluster Analysis"*. London: Arnold, 2001.

[11] E. Backer and A. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 1, pp. 66–75, Jan. 1981.

[12] P. Berkhin. (2001) *Survey of clustering data mining techniques*. [Online]. Available: http://www.accrue.com/products/rp_cluster_review.pdf http://citeseer.nj.nec.com/berkhin02survey.html.
[13] J. Cherng and M. Lo, "A hypergraph based clustering algorithm for spatial data sets," in *Proc. IEEE Int. Conf. Data Mining (ICDM'01)*, 2001, pp. 83–90.
[14] P. Sneath, "The application of computers to taxonomy," *J. Gen. Microbiol.*, vol. 17, pp. 201–226, 1957.
[15] T. Sorensen, *"A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyzes of the vegetation on Danish commons," Biologiske Skrifter*, vol. 5, pp. 1–34, 1948.
[16] J. MacQueen, *"Some methods for classification and analysis of multivariate observations,"* in *Proc. 5th Berkeley Symp.*, vol. 1, 1967, pp. 281–297.
[17] L. Zadeh, *"Fuzzy sets," Inf. Control*, vol. 8, pp. 338–353, 1965.
[18] J. Bezdek, "*Pattern Recognition with Fuzzy Objective Function Algorithms"*. New York: Plenum, 1981.
[19] G. Gan, C. Ma and J. Wu*, "Data clustering: theory, algorithms, and applications"*. Society for Industrial & Applied Mathematics, Philadelphia, 2007.
[20 M-S.Yang, Yu-Jen Hua, Karen Chia-Ren Linb and Charles Chia-Lee Linc, *'Segmentation techniques for tissue differentiation in MRI of Ophthalmology using fuzzy clustering algos"*. Magnetic Resonance Imaging, vol 20, no 1, 2002, pp: 173-179.
[21] X. Yuan, T. M. Khoshgoftaar, E. Allen and K. Ganesan, *"An application of fuzzy clustering to software quality prediction"*. Proceedings International Conference on *Software Engineering Technology,* Richardson, TX, March 2000, pp: 85-90.
[22] R. Xu and D. Wunsch, *'Survey of clustering algorithms"*. Neural Networks, vol 16, no 3, 2005, pp: 645-678.
[22] Gabriela Serban and Grigoreta Sofia Moldovan, *"A Comparison of Clustering Techniques in Aspect Mining"*. Studia univ. Babes Bolyai, Informatica, vol. LI, no. 1, 2006.
[23] Yi Lu, Shiyong Lu, Farshad Fotouhi, Youping Deng, Susan J. Brown, *"FGKA: A Fast Genetic K-means Clustering Algorithm"*. Poster Abstract, 2003.
[24] Deepak Gupta, Vinay Goyal and Harish Mittal, " Comparative Study of Soft Computing Techniques for S/W Quality Models". International Journal of S/W Engineering Research & Practice, Vol-I,Issue I,Jan 2011.
[25] Deepak Gupta ,Vinay Goyal and Harish Mittal, "Software fault Prediction using Fuzzy Clustering". 1st International Peer Reviewed Journal of Engineering & Technology,in SGVU, Jaipur , 2012.
[26] Deepak Gupta ,Vinay Goyal and Harish Mittal, "Software Quality Analysis of Unlabeled Program Moduls with fuzzy-C means Clustering Techniques.". published in IMRS's international Journal of Engineering Sciences, Vol-I , Issue 02, june,2012.