

# Maintenance of Softwares in Various Contexts is one of the Main Roles of Clustering In Data Mining Technology

Aziz Khan

**Abstract** - System maintenance in various contexts is an experience that found in tracking a bug and source text code Configuration Management System showing that it is costly and lots of time consuming part of Software-Life-Cycle. Software maintenance has many issues but more pressing in case of legacy Software systems. Clustering in data mining and business intelligence (BI) – Machine learning methodologies permit one to abstract Modals from past experience to make their predictions for future use. Here we discuss the methods employing and solutions regarding encountered issues. We will also present some of the results that we have got.

**Index Terms** – SC: Source-Code, DM: Data mining, JP-Clustering: Jarvis–Patrick Clustering, SVD: singular value decomposition, ISA: Identification of Sub-systems based on Association), PCA: Principal Component Analysis, BI: Business-Intelligence, (M.D.G): Module-Dependency-Graph,

## I. INTRODUCTION

In general computing terminology, legacy system is an old application program or computer system which is in running form to be used due to their cost of replacing or redesigning it and often despite its poor competitiveness and compatibility with new techniques and modern equivalence – the repercussion is, the huge-system, massive and very hard to amend. That's why, Software maintenance is also called sever problems. This is certainly the case for legacy software systems where frequently there are non-trivial relations between different components of the system that are not known.

Data mining with the collaboration of graphs regarding attribute relational is very important approach in this context, as entities considered their siblings (Nod) with the other related & dependent-data as branches/edges. There are different techniques used for data mining and association rules that is an important and useful technique is one of them. In multi-dimensional data association rules play an important role to dig-out an association among their components or items e.g. genes related data, and also the software-component performs association-rule based on data mining in EDPS (Electronic Data Processing System). Different applications are used for association rules mining which decomposes and convert graphs continuing data to Entity-domains, in this case association property is the main

concern i.e. mainly based on. Object oriented/based languages has variety of their properties and abstraction is of one them – clustering is used to support software maintenance and their interaction-with with the usage of Abstraction by using programming languages such as C++ or/and java etc.

The main purpose of clustering is to cluster the data by finding some 'reasonable' group of data items. As clustering is unsupervised learning but more commonly it is just classification that seeks to search rules for classifying objects given a set of pre-classified objects, clustering does not use pre-defined label types related to data items. Clustering algorithm is designed to find structure in the current data, and not for the classification of future data.

Data mining has the capability of producing elevated & sophisticated-overview in the source-code-program with the interdependencies & associations in the elements as well. The main theme and main objective of this paper is to whether maintenance of software systems may supported by data mining engine, plus to propose a methodology to identify and support pattern-recognition and similarity index as well. Data mining has the capability to dig-out and convert source-program into suitable database-components to extract useful information. Cluster analysis is introduced to extract useful and related information and to squeeze high-dimensional data, to find a similar function among program components & entities. As clustering is supervised learning expressing that it doesn't acquire any previous domain understanding – means that maintainers become easier without knowing about the program or with limited knowledge they can assess and analyze a program. Different types of clusters techniques are used here to group and extract knowledge from Source-Program to get program structure and to understand the same program in a sophisticated way. A data-modal is used to group and then organized the entities on their similarity matrix – showing their inter-related relationship among them.

## II. BACKGROUND

There is a need to extract useful information from the data and to interpret the same. Automated data collection tools and mature database technology lead to tremendous amounts of data stored in databases, data warehouses and other information repositories. The main objective is to find the similarity and dissimilarity index in-between the groups in high-dimensional data.

Clustering will be better if the data are showing more homogeneity in the group or more Heteroscedasticity (type of heterogeneity restricted to in-quality of variances). Credit goes to methodology of data mining that it retrieves extraordinary & previously unknown association between attributes in huge databases. It is worth to mention that data mining is also responsible of extracting information from high-dimensional source-data containing complicated systems with its application as well. But, somehow Curse of Dimensionality issue occurs when there is very huge volume and complex structure of data e.g. genes genomic related data. Reveal of priority unfamiliar non-trivial patterns, hidden relationship among software components is one of the tremendous property of the data mining for software maintenance with low knowledge regarding functionality details. Data mining has wider application and usage, somewhere it is used for clustering over Module-Dependency-Graph (M.D.G) and also for ISA methodology, which stand for identification of sub system based on Association. Abstraction is the main theme in the above approaches. M.D.G creates a tree like structure with the relationship of their siblings (components) as sensed by source program. MDG is beneficial in the sense that if someone is interested to extract knowledge in structured view. As ISA concerns, it decomposes the structure (system) into their system components through its relationships among the components of the source program and looks for their shared-files/ programs. For the sake to maintain design quality, relationship between their system-components and system modularity should be checked and measured carefully.

### III. MATERIAL AND METHODS/ CONCEPTS

Broadly speaking, Knowledge based software Engineering (KBSE) is in application of AI technology to SE, such systems explicitly encodes the knowledge that they employ [3]. Knowledge based software Engineering devised to help software developers in software maintenance on low scale on daily basis, to diminish/reduce and recognize the interrelationship among different mechanisms of the software systems.

Cluster analysis has been used to extract useful related entities<sup>1</sup> (source code), to find a similar function among them. Java is a powerful OO-Programming language with strong code, definition and approach of entities, on the basis of similarity matrix the same entities are grouped together by using clustering techniques. The main concern is the object of a similar (or related) to each other and different (or unrelated) from other groups object group. Grouping based on information in the description of the data objects and their relationships are found in the analysis of the object/ source-data (source code).



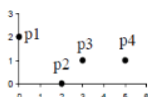
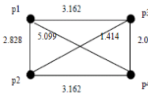
Fig. 1. Clusters

The main theme is to find the homogeneity and heterogeneity in-between the groups in source-code data. Clustering will be better if the data are showing more homogeneity in the group or more Heteroscedasticity (type of heterogeneity restricted to in-quality of variances). Clustering dealt with the points most tightly connected to each other by discarding background and noise-points- aspects. True Cluster may further sub-divided into sub-clusters. The important requirement in above case is sub-clusters that is precise and mostly all the points are inter-related and inter-connected.



Fig. 2. Example of different Clusters

TABLE I. Data (Tetra-Points) with their nearest Graph

 <p>a) points</p>	Points	Xx	Yy				
	P1	0	2				
	P2	2	0				
	P3	3	1				
	P4	5	1				
 <p>d) proximity graph</p>		P1	P2	P3	P4		
	P1	0.00	2.83	3.20	5.10		
	P2	2.83	0.00	1.41	3.20		
	P3	3.13	1.41	0.00	2.00		
	P4	5.10	3.20	2.00	0.00		
Graph		Proximity Matrix					

A cluster is a set of source-data that close to each other (proximity & similarity) than the data outside the cluster. It is also worth to mention that cluster is a set of data/object that close to centroid of the same cluster as compared to the centroid of other clusters not belonging to them. Therefore, many clustering algorithms use the following criteria:

1. Proximity or Nearest Neighbor.
2. Cluster composed of a set of data streams that close to each other (proximity & similarity) on the basis of distance and other similarities with their neighbors than the data outside the cluster.
3. Cluster definition based on Density: cluster comprises of compressed area of dots showing that this is high density area along-with low-density area showing that the cluster contains



Fig. 3.

<sup>1</sup>**Entity:** anything about which an organization is interested to collect information.

lots of irregularities and noise effect in the cluster.

4. Some clusters are similar, on the basis of their objects/streams similarities and vice versa. The dispersion is only due to the high and low density areas and their irregular shapes in the cluster. In this very case to check the proximity, the type of object/data stream. Typical three types of attribute values are in Binary, discrete and continuous. Some SPSS terms regarding data is as follows:

- a. Categorical values: is information gathered from a study i.e. descriptive and not based on numbers. This type data is then not measurable. Qualitative Nominal values are the values of Names, NIC, Address, zip-code etc. that the data entered have no superiority on one another (may be in any order).
- b. Ordinal data: the data that has to be entered has the superiority with one another i.e. the data is in order such as:
  - i. Effected, Least-Effectuated, More-Effectuated,
  - ii. Daily-Basis, Weekly-Basis, Monthly-Basis,
  - iii. Good, Better, Best.

Quantitative: the values in number are meaningful. In quantitative-Interval, the difference between the values is significance. e.g. salaries, temperature in Degree Centigrade (Celsius) or Fahrenheit or Kelvin.

Euclidean Distance: The very famous proximity measure is the Euclidean Distance equation to measure scales with an absolute zero (Minkowski Space). The formula is given by:-

$$P_{ij} = \left( \sum_{k=1}^d |x_{ik} - x_{jk}|^r \right)^{1/r}$$

Accordingly, the proximity distance between any two points in xy-coordinates is as given below:

$$\text{dist}((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2}$$

By putting some values such as (2, -1) and (-2, 2), in the above formula, as given below:

$$\begin{aligned} \text{Dist}((2, -1), (-2, 2)) &= \sqrt{(2 - (-2))^2 + ((-1) - 2)^2} \\ &= \sqrt{(2+2)^2 + ((-1) - 2)^2} \\ &= \sqrt{(4)^2 + ((-3))^2} \\ &= \sqrt{16 + 9} \\ &= \sqrt{(4)^2 + ((-3))^2} \\ &= \sqrt{16 + 9} \rightarrow \sqrt{25} \rightarrow 5 \end{aligned}$$

Consider Euclidean Plane for two dimensions: if p equals to (p1, p2) and q equals to (q1, q2) then the distance is as under:-

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

If polar-coordinates of point P, are (r1, θ1) and q are (r2, θ2), then the distance between the points is as follows:

$$\sqrt{r_1^2 + r_2^2 - 2r_1r_2 \cos(\theta_1 - \theta_2)}$$

Euclidean-Space for three plus N-dimensions, the equations are as follows:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}$$

For N dimensions:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_i - q_i)^2 + \dots + (p_n - q_n)^2}$$

Variations in Euclidean Distance<sup>2</sup>:

$$P_{ij} = \left( \sum_{k=1}^d |x_{ik} - x_{jk}|^r \right)^{1/r}$$

d represents dimensionality, parameter is r, X<sub>ik</sub>, and X<sub>jk</sub> are the k<sup>th</sup> components of I<sup>th</sup> and J<sup>th</sup> objects respectively. There are L1-Norm i.e. r = 1, for L2-Norm r = 2, which occurs most commonly (Euclidean distance). L<sub>max</sub> Norm represent when r tends to infinity. Reference Figure-1 the proximity matrix for the above mentioned Norms i.e. L1, L2 and L<sub>∞</sub> is as given below:-

TABLE II. Data matrix L1, L2 & Proximity matrix

Points	Xx	Yy	L2	P1	P2	P3	P4
P1	0	2	P1	0.00	2.83	3.20	5.10
P2	2	0	P2	2.83	0.00	1.41	3.20
P3	3	1	P3	3.13	1.41	0.00	2.00
P4	5	1	P4	5.10	3.20	2.00	0.00

L1	P1	P2	P3	P4	L <sub>∞</sub>	P1	P2	P3	P4
P1	0.00	4.00	4.00	6.00	P1	0.00	2.0	3.0	5.0
P2	4.00	0.00	2.00	4.00	P2	2.0	0.00	1.0	3.0
P3	4.00	2.00	0.00	2.00	P3	3.0	2.0	0.00	2.0
P4	6.00	4.00	2.00	0.00	P4	5.0	3.0	2.0	0.00

So, the calculated Minkowski-Distance is exactly the metric-distance as it satisfies the Symmetrical Mathematical-Functions-Reflex given as under:-

Dist(Xx, Xx)=0,

(dist(Xx, Yy)=dist(Yy, Xx)) and the data in triangle

inequality:(dist(Xx, Zz) ≤ dist(Xx, Yy)+dist(Yy, Xx).

<sup>2</sup>Euclidean Distance mainly used for proximity measure, while Minkowski Metric is used for ratio scale with an absolute zero, which is the generalization of distance between Euclidean Space Points.

The approach consists of mainly two measures, the clustering algorithm and the input data modal. As concerns the input data modal, it takes some limited components of source-code-Program i.e. Class, functions, procedures (parameters), Files and Packages. Entities extracted from the above mentioned elements which are placed in various components of the database for further knowledge extraction, with a number of different fields/attributes with their interconnected association as given below:-

TABLE III

Table Name	Attributes	Description
<b>Files</b>	File-ID	Primary Key (Uniqueness)
	fileName	Full (Folder) path
<b>Packages</b>	packageID	PackageUniqueID
	packageName	Name
	ImportedPackage	ImportAPI
<b>Class</b>	ID	Full Family
	ID	Primary Key(Uniqueness)
	Name	Name
	Inherit	Boolean
	Inherits-From	baseClass
	Implement	interfaceName
<b>Methods</b>	ImplementsTo	Class that follows word implement
	fileID	Full Family
	ID	Primary Key(Uniqueness)
	Name	Name of Method
	Arguments	Boolean
	Argument-Num	Argumentpassed byfunction
	Returned-Value	Returnedby the function
	Other	Any other property
	Modifiers	That method declare a modifier
	ID	FileID of the method
<b>Parameters</b>	parameterID	Primary Key(Uniqueness)
	Parameter-Name	Name
	Parameter-Type	type
	Parameter-Use	Parameter Ref.
	ID	of respected paramter

<sup>3</sup>Hierarchical-Agglomerative-clustering (AHC) Algorithm is extensively used to provide better results by measuring similarity index of the source-code data. Such algorithm take their start with each data point as individual and single clusters and by combining all these points form a cluster. By decreasing similarity-index preprocessed

data is required in such a case by this algorithm. All the entities that has their numerical as well as in nominal values are converted to numerical values (distance b/t two records in the database) for the sake that the proximity measure may easily be done which will be stored in similarity-matrix. Numerical value ranges from 0 to 1 i.e. 0 indicate the most similar while 1 indicates the most dissimilar value (farthest value), equation is given by:-

$$d(i + j) = \frac{\sum_{f=1}^n X_{i,j} Y_{i,j}}{\sum_{f=1}^n x_{i,j}}$$

Distance Calculation Equation.

Where i & j are tuples number of attributes is represented by n of the concerned records.

Product of Sum of X & Y functions is directly proportional to the distance function and inversely proportional to the Sum of X function, where  $Y_{i,j}$  is given by,

$$Y_{i,j} = \frac{|q_{i,n} - q_{j,n}|}{(\max(q_{m,n}) - \min(q_{m,n}))}$$

And  $X_{i,j}$  possess just 1 & 0 values.  $X_{i,j} = 1$  means when an attribute i.e. ( $q_{i,f}$  or  $q_{j,f}$ ) of one of the two records is not missing otherwise  $X_{i,j} = 0$ . In a similar fission  $Y_{i,j}$  is dependent on the attribute-type of the concerned record.

- If (attribute-type==binary || attribute-type == nominal)
  - { if ( $q_{i,n} == q_{j,n}$ )
  - {  $Y_{i,j} == 0$ ; }
  - else
  - {  $Y_{i,j} = 1$ ; }
  - }
- If (attribute-type== numerical)

$$Y_{i,j} = \frac{|q_{i,n} - q_{j,n}|}{(\max(q_{m,n}) - \min(q_{m,n}))}$$

|  $q_{i,n} - q_{j,n}$  | Mod value,  $\max(q_{i,n})$ ,  $\min(q_{j,n})$  is the maximum & min value of the fields related to specific tuple.

For the Software maintenance, in Software Engineering source data is to be converted to some suitable format that maintained easily and sophisticatedly, the same formula is applied discussed earlier to convert source data of concerned application appropriate format as required.

#### IV. CLUSTER MERGING

Single-link or single-linkage clustering terminologies are used which refers to the proximity measure of two clusters between their most similar members –generally called local-merge-criterion. By this way, neighboring

<sup>3</sup>Hierarchical Agglomerative Clustering One of the clustering approaches, which is a bottom-up clustering method. The concept of sub-clusters found here. e.g. species taxonomy in animals and plants in

biology. A specie is building-blocks of biological categorization and taxonomic ranking i.e. kingdom, phylum, genus, species etc. agglomerative Hierarchical Clustering.

clustering technique occurs besides these more distant parts of the cluster and the clusters' overall structure are not taken into account. Whilst, on the other hand, in complete-link or complete-linkage clustering, the proximity measure of two clusters is the comparison of their most heterogeneous member – this means that the diameter falls at the smallest point to merge a cluster-pair (Non-local criterion) which causes sensitivity to outliers. Single-linkage technique is used to merge clusters recursively. If the members of the two clusters are close-to-each-other, that clusters will be merge together.

1. Jarvis – Patrick (JP) clustering encounters Nearest-Neighbor-Approach to clusters points. Distance between two points/ objects are measured. Let's J be the size of the closest neighbors and P be the number of closest neighbors. The following steps must be encountered.
2. Determine the size of closest neighbor in terms of J for each object in the cluster.
3. The objects 'A' & 'B' will be considered in the same cluster if 'B' contains in 'A' neighbors list and 'A' contains in 'B' neighbors list.
4. Both 'A' & 'B' at least P nearest neighbors in common.

By applying the Jarvis – Patrick (JP) clustering approach a singleton clusters formed here is different than that of formed in any other approach.

## V. K-MEANS CLUSTERING ALGORITHM:

K-Means clustering algorithm Looks for k (non-overlapping) different clusters, and break data into K-1 different partitions by placing some points as centroids. If  $k = 3$ , the data will be break into 3 clusters as shown in Fig 4(A). Three seeds are randomly placed by K-Means algorithm to decide which cluster is associated with its nearest centroid. Centroids are only for the sake to minimize the ERROR term. For the same we draw straight line between two seeds as shown in graph:-



Fig. 4(A)

Calculate the Mid-Point of the line and draw perpendicular line over it as as in Fig 4(b):-



So according to the High School Geometry any point to the left of this point will be near to seed No.1 and any point to the right of this line is near to seed No.2. Similarly if we draw line between seed No.1 and seed No.3 and then draw a perpendicular bisector of this line as shown in Fig 4(c).



Fig. 4(b)(Mid-Point)

Now, any observation below this perpendicular line belongs to seed No.1, and any observation above the line belongs to seed No.3. The combination of both perpendicular bisectors, any point below line belongs to seed No.1, as given in Fig. 4(d).

Similarly, for seed No.2 and seed No.3 are as given in Fig. 4(e) respectively. Here we assigned all the records into three different groups; assigned them to one of the three seeds, the below shown graph is the first set of cluster that are formed. K-Means Algorithm finds Random Clusters as:

- a. Selection of random clusters with the identification of their centroids.
- b. Differentiate Points b/t with respect to its nearest points.

On the basis of above point 2 centroids are again to be taken. Above Points 2 & 3 will be executed again and again until centroids remain the same.

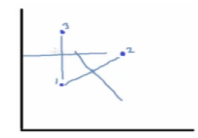


Fig. 4(c)



Fig.4(d)

## VI. CONCEPT TOOL & PROCESS MODAL

Consider the following process-modal for the evaluation of the methodology used here:

- The Preprocessing-Engine mainly used to analyze and store the extracted source-code (program) data according to the input modal in a database. After processing the source-code data different clustering techniques of data mining is to be performed for transformation and further implementation.
- Hierarchical Agglomerative clustering technique that is a bottom-up clustering method make able database records and attribute-selection to form clusters of source-program elements with the help of Hierarchical-Agglomerative-clustering algorithm. After analyzing the clusters-data output displayed through GUI, as shown below:-

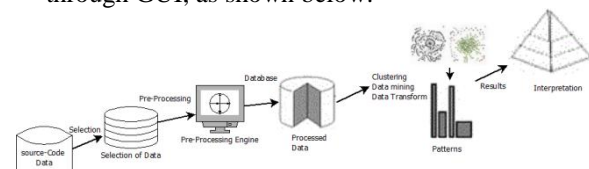


Fig. 5. Process Model

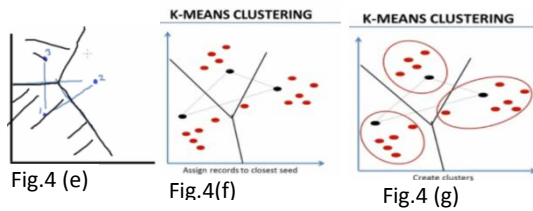
## VII. OTHER APPROACHES

This paper concentrates on grouping (discretizes) source-code data/elements based on their proximity measure (Euclidean Distance).

By analyzing the data from different sources in high dimensional data on the basis of source-code (source program), is a complex task, but clustering make a sense to do the same on the basis of proximity measure. In such a situation the gap between the selected data with relation and with respect to their performance is to be measured, and the difference of their adjacent points (may be near or farthest) will reach to zero in high dimensional data. Such facts occur if the points within the cluster are identically and independently distributed.



$$\lim_{d \rightarrow \infty} \frac{\text{MaxDist} - \text{MinDist}}{\text{MinDist}} = 0$$



Observation from real-world problems are often high dimensional vectors i.e. containing many variables. Relatively in Neural Networks a small number of high dimensional data is very difficult to handle. In high-dimensionality data we are interested to separate meaningful information (voices) but at the same time the result reflects the problem of “Curse of dimensionality”. In this case absolute difference instead of relative difference is preferred as: -

$$\text{MaxDistance} - \text{MinDistance}$$

For  $L = 1$  metric  $\text{MaxDistance} - \text{MinDistance}$  is directly proportional to dimensionality, while for  $L=2$  metric  $\text{MaxDistance} - \text{MinDistance}$  remains relatively constant but for  $L \geq 3$  metric  $\text{MaxDistance} - \text{MinDistance}$  tends to zero as dimensionality increases (i.e. meaningless), this problem may overcome if we decrease data dimensionality without losing required information (apriority) i.e. only most frequent items has to be extracted that are of interest and rest are discarded (i.e. with little variations or with high correlation). As a result dimensionality reduces in this way.

It is also worth to mention that high dimensional data is also a cause of noise in data, which is another problem. And such problem may be resolved by keeping relatively small number of dimensions from bulk of dimensions. In such case some “true” information may lose and will enhance data analysis but is an efficient way to remove noise from data. Statistics and linear algebra techniques such as Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) has a wide contribution to make high dimensional data more concrete and brief.

Different types of applications are in used to day. K.I.T (Keep-in-Touché) is one of them, which provides 15 methods. The programmer grouped these methods according to their proximity measure (DB Control, Setters-Getters, Display-modes) shown in Table IV.

By matching the above table-data, 86% correct placements were made, only activity-Support was scattered. On the other hand due to some arguments prospect-Activity kept separated.

## VIII. CONCLUSIONS

Discussion regarding clustering analysis Maintenance of softwares in various contexts is one of the main roles of data mining technology. Initial experimental results from above discussion were

TABLE IV.

#	Group 1 (DB Control)	Group 2 (Setters-Getters)	Group3 (display-mode)
1.	Prospect-Activity	Activity-Support	Activity-Form
2.	Get-Practivity-Row	Clear-All-Fields	Show-Dialog
3.	getPPractivityPK	updateAllFields	setState
4.	getActivity	create	process
5.	getDescription	commit	fire

#	Group-1 (D.B Control)	Group-2 (Setters-Getters)	Group-3 (display-mode)
1	Get-Practivity-Row	Clear-AllFields	activityForm
2	Get-PPractivity-PK	Update-AllFields	showDialog
3	Get-Activity	generate	setState
4	Get-Description	comit	manage
			Fired
			Activity-Support
<b>Group 4 (Misc)</b>			
<b>prospect</b>			
<b>Activity</b>			

encouraging which successfully reveals similarities among source-code elements (java in practice). Efficiency increase with the increase of entity size with the addition of more attributes.

For example a package-ID, a Class-ID, and a Method-ID could be included at the Class, Method and Parameter entities. Other possibilities worth of exploration involve extending the data modal with more grammatical Elements like objects and array& structures, IF-Then-ELSE, DO-Until, DO-WHILE, SWITCH and Exceptions.

K-Mean’s algorithm looks for k different clusters and divide the source-data into k-1 different clusters with the issue of “Curse of Dimensionality”. K-Means algorithm (vector property) itself select some random points for the separation of different clusters as centroids. This is the point where process of similarity and proximity matrices are to be calculated.

The solution of the issue regarding “Curse of Dimensionality”. Cluster analysis has been used to extract related streams, to find a similar function of source-code/programs.

## IX. ACKNOWLEDGMENT:

Al-Mighty Allah, the Most Merciful and Most Beneficent has made me able to publish this paper in

Journal of Basic & Applied Scientific Research (JBASR). I would like to pay my regards to all my family members, friends and teachers for their help and support. But frankly speaking especially my parents have the great contribution in it to praying for me from their core of their hearts.

It is worth to mention to please allow me to dedicate my acknowledgment especially to Dr. Abdul Salam (Head of the Department Computer Science) who provided me the valuable advices and guidance. My Research Paper would not possible without missing any of the above personalities. God bless all of them.

## REFERENCES

- [1] Alberto Fernandez, Sergio Gomez, "Solving non-uniqueness in Agglomerative Hierarchical Clustering Using Multidendrograms".
- [2] Charu Aggarwal, Cecilia Procopiuc, Joel Wolf, Phillip Yu & Jong Park "Fast Algorithm in Projected Clustering" (Conference 1998).
- [3] R. McCartney. Knowledge Based Software Engineering: Where we are and where we are going. Automated Software Design. Edited by M.R Lowery and R.D. McCartney, .AAAI Press, 1991.
- [4] N. Anquetil and T. C. Lethbridge, "Experiments with Clustering as a Software Remodularization method", Proc. 6th Working Conf. Reverse Engineering (WCRE 99), IEEE Comp. Soc. Press, Oct. 1999, pp. 235-255.
- [5] RA Jarvis, EA Patrick, IEEE Transactions on Computers (1973). RA Jervis, EA Patrick, "Clustering Using Similarity Measure Based on Shared Nearest Neighbor".
- [6] Y. Kanellopoulos and C. Tjortjis, "Data Mining Source Code to Facilitate Program Comprehension: Experiments on Clustering Data Retrieved from C++ Programs", Proc. IEEE 12th Int'l Workshop Program Comprehension (IWPC 2004), IEEE Comp. Soc. Press, 2004, pp. 214-223.
- [7] Ying Zhao, George Karypis, "Evaluation of hierarchical clustering algorithms for document datasets" article 2002.
- [8] Difference between Hierarchical and Partitional Clustering by Indika .Posted on May 29th, 2011.
- [9] U. Fayyad, G. Piatetsky-Shapiro, and R. Uthurusamy, "From Data Mining to Knowledge Discovery: An Overview", in Advances In Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, 1996.
- [10] T.M. Pigoski, Practical Software Maintenance: Best Practices for Managing your Software Investment, Wiley Computer Publishing, 1996.
- [11] M.H. Dunham, Data Mining, Introductory and advanced topics, Prentice Hall, 2002.