ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 7, July 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Time and Cost Reduction through Hierarchical and K-mean Method

Renu Bala[1]
M. Tech. Scholar,
Department of Computer Science & Engg., OITM,
Hisar, India

Surender Singh[2]
Assistant Professor,
Department of Computer Science & Engg., OITM,
Hisar, India

Abstract: ***Cluster analysis is a method which is meant for classification of the data into natural groups. Recently Cluster analysis is linked with software automatic categorization system. A categorization system is one that is structured so that entities within the system with common features will be grouped together. This paper describes K-mean Method with hierarchical algorithm – agglomerative algorithm for cluster analysis with empirical study.***

I. INTRODUCTION

Software categorization defined as the activity of labeling software, belonging to different domains. Generally there are two different ways for making automatic classifiers. In a knowledge engineering approach, the knowledge of human experts is described as a set of rules, which are then used in the process of classification. The disadvantage of this approach is that it requires lot of effort to make human knowledge explicit and for each new domain a separate formulation of the rules need to done manually again. In a machine learning approach, the classifier is built automatically and classification for different domains can be learned using the same algorithm. The exactness of all automatic classification system is extremely reliant upon the effort and concern taken during process[1].

A cluster analysis plays big role in software categorization. Cluster analysis is used to separate the data into bunch or groups where no prior information is available. It divides data into groups (clusters) that are meaningful, useful or both and then the clusters should based on the original structure of the data. The cluster analysis is a vital tool in decision making and an effective method to obtaining solutions. The units within a cluster are as similar as possible, and clusters are also as different as possible. The main reasons for doing a cluster analysis are data exploration, visualization, data reduction, hypothesis generation. Partitioning or clustering techniques are used in many areas for a wide spectrum of problems. Cluster analysis can be applied for graph theory, business area analysis, information architecture, information retrieval, resource allocation, image processing, software testing, galaxy studies, chip design, pattern recognition, economics, statistics and biology[2]. Figure-1 shows that the member entities of one cluster are not part of other clusters. The items which are to be clustered (the points) are represented by bullets. The dotted shapes signify clusters; the points within each dotted shape comprise a cluster. The goal of clustering methods is to extract an existing 'natural' cluster structure. However, different methods may come up with different clustering. as a result, a particular algorithm may define a structure rather than find an existing one. It might even be the case that an algorithm 'finds' a structure while there really is no natural structure in the data.

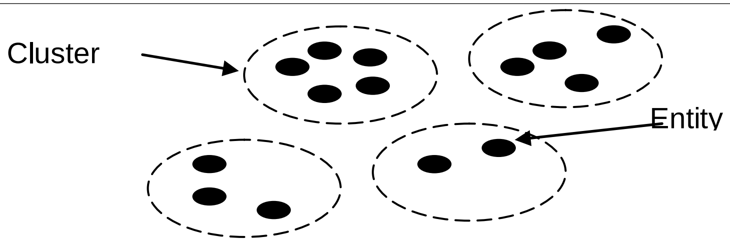

Figure-1

This paper includes four sections, Algorithmic approaches adopted for cluster analysis in section2; Euclidean Distance Metrics and Hierarchical Algorithms for cluster analysis presented in section 2.1 and section 2.2., lastly the conclusion is explained in section 3.

II. Approach Adopted

Cluster analysis is a well established field of research that has been applied to many different disciplines. Many research papers concluded that cluster analysis may also be fit for reengineering purposes and a number of different projects are currently pursuing this hypothesis.Cluster analysis discovers a natural structure within a set of entities. The method is based on discrete descriptions of these entities and is designed to deal with very complex relationships between these entities. Therefore cluster analysis is very well suited to software categorization which involves taking a group of functions with complex relationships involving data structures and other functions and merging them together in logically coherent groups.[3] There are a large number of techniques available, each of which may or may not be suitable for software categorization. Cluster analysis providing the relevant information about a characteristic can be extracted from the code. Firstly, we find out the similarity between two entities. Using the information in the data matrix and retrieved values and the similarity between the entities is calculated using the selected metric based on these values. This work is carried out using a clustering algorithm.

Distance Measure: An important component of a clustering algorithm is the distance measure between entity points. The tree clustering method calculates the dissimilarities (similarities) or distances between the different objects. Similarities are defined as a set of rules on that bases grouping or separation of items are done. This is a significant step to select a distance measure, which will calculate the similarities of two elements. This can be explained through the following figure.

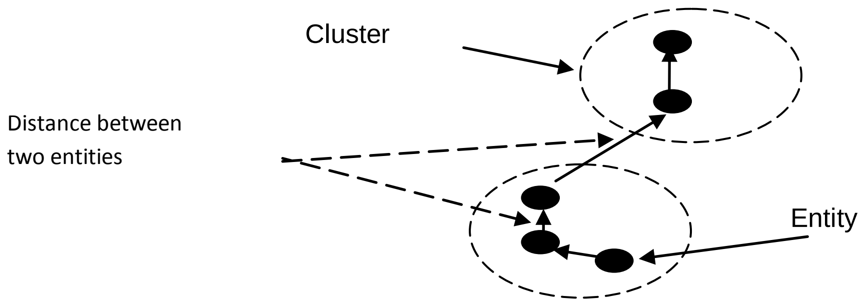

Figure-2 Distance Between Two Closest Entities

Figure-2 shows distance between two closest entities. Notice that this is not only a graphic issue: the problem arises from the mathematical formula, which is used to combine the distances between the single components of the data feature vectors into a unique distance measure. That can be used for clustering purposes as different formulas leads to different clustering.[4] And domain knowledge is used to direct the formulation of a suitable distance measure for each particular function. Data consisting of measures of dissimilarity between all pairs of two units can be represented using a dissimilarity matrix D of the form. The dissimilarity matrix D can be constructed by a distance measure, called a metric. The most commonly used distance measure is a Euclidean distance based on the sum of the squared differences between pairs of measurements.

Hierarchical algorithms

There are two kinds of hierarchical algorithms: agglomerative and divisive algorithms. The hierarchy of clusters is built in a manner that each level contains the same clusters, which are on the lower hierarchy. Following figure-3 shows an example of such hierarchy for three entities.

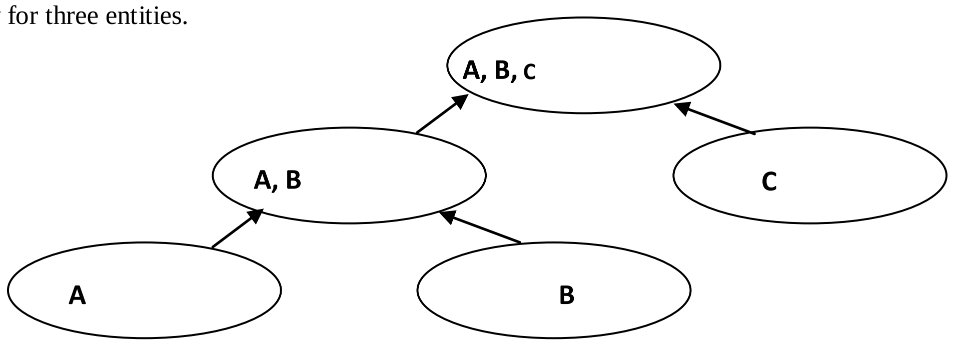

Figure-3 Hierarchical clustering

Hierarchical algorithms can be agglomerative ("Bottom-Up") or divisive ("Top-Down"). Agglomerative algorithm starts with the bottom of the hierarchy; at the starting point there are N clusters (each containing one entity). In each following step two clusters are joined. After N-1 steps all entities are contained in one cluster. Each level in the hierarchy defines a clustering. Now a cut point is the resulting clustering.[5] The resulting hierarchy of the hierarchical method is usually visualized in dendrogram.

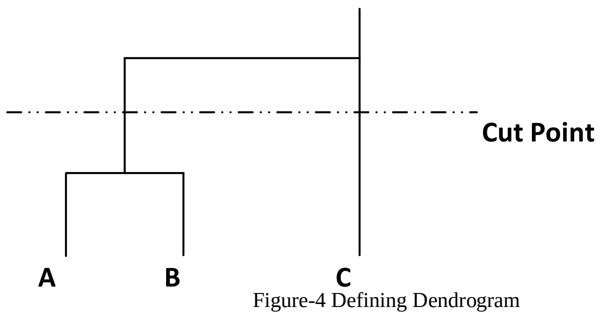

Figure-4 Defining Dendrogram

Figure-4 is graphically defining dendrogram for three entities. In divisive clustering all entities are contained in one cluster. In each step a cluster is split into two clusters. After N-1 steps there are N clusters each containing one entity.

Agglomerative hierarchical algorithms are most widely used technique as it is not feasible to judge all possible divisions of the first large clusters (2^{N-1}-1 possibilities in the first step). Comprehensively, the algorithm consists of the subsequent steps:

- Compute the proximity matrix.
- Repeat
- Merge the closest two clusters.
- Update the proximity matrix to reveal the closeness between the new cluster and the original clusters.

There are several regulations for deciding how the combined units should be treated. These rules are based on the idea *linkage*, the features of the two groups joined carried over to the combination of the groups.

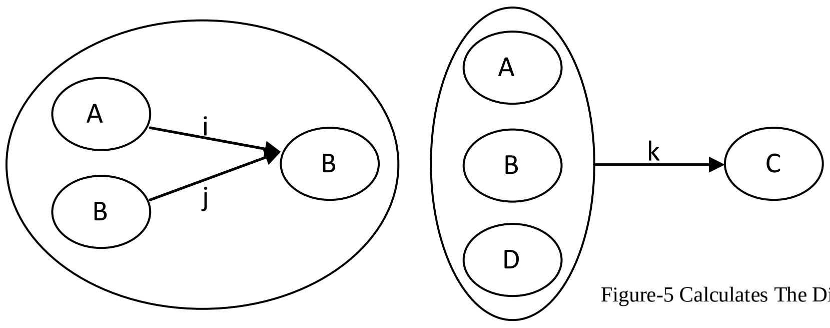

Figure-5 Calculates The Distance Between Various Clusters

Figure-5 Calculates the distance between various clusters. Suppose there are three clusters, A, B and C, with the distance between A and C being i and the distance between B and C being j. Let A and B are the most alike pair of entities, they must be clustered jointly into a new cluster, D. We then need to calculating the distance between C and D, K. The Three algorithms that have been used are differentiated by the way they deal with this issue.

- ✓ Single Linkage
- ✓ Complete Linkage
- ✓ Average Linkage

The single-linkage method, which is also called nearest-neighbour method.[6] The MIN version of hierarchical clustering, the closeness of two clusters is defined as the minimum of the distance (maximum of the similarity). Starting with all the points as single cluster and add links between these points at a time, smallest links first, then these single links combines the points into clusters.

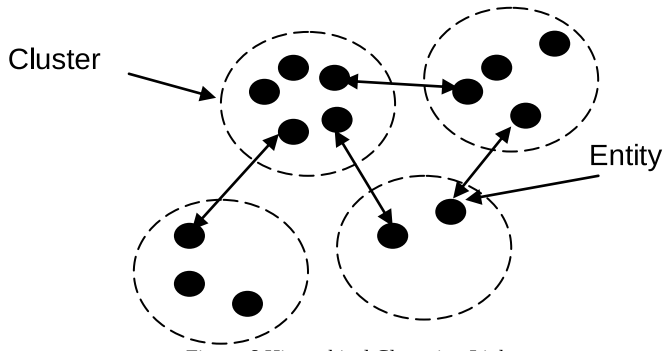

Figure-6 Hierarchical Clustering Links

Figure 6 shows single link method for hierarchical clustering: K = min (i, j). The single link technique is fit for handling non-elliptical shapes, but is sensitive to noise and outliers. This form of linkage means that a single link is enough to join to groups, and this feature will allow clusters to elongate and not essentially sphere-shaped.

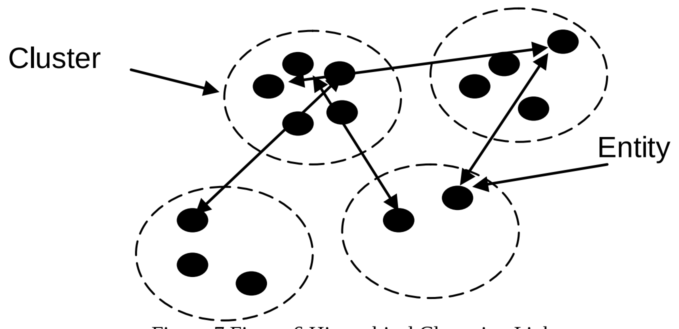

Figure-7 Figure-6 Hierarchical Clustering Links

Figure-7 shows complete link method for hierarchical clustering: K = max (i, j). The complete link or MAX version of hierarchical clustering, the proximity of two clusters is defined as the maximum of the distance (minimum of the similarity) between any two points in the two different clusters. Graphically, started with all points as singleton clusters and add links between these points one at a time, shortest links first, then a group of points is not a cluster until all the points in it are complete linked, i.e., form a *clique*. Complete linkage is less sensitive to noise and outliers, but it can break large clusters and it favors globular shapes.

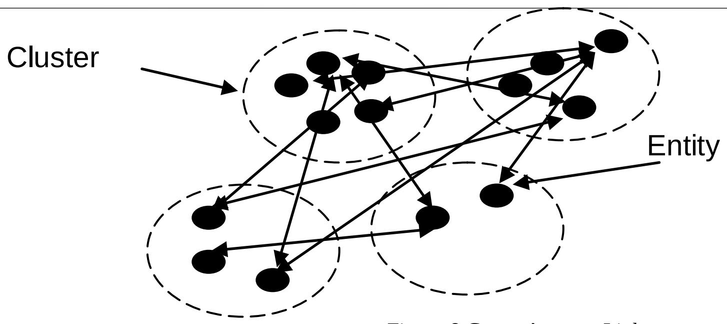

Figure-8 Group Average Link

Figure 8 shows Group Average Link: K = (I, j) / 2. In the average linkage method, also called un-weighted pair-group method using arithmetic averages (UPGMA), the proximity of two clusters is defined as the average pair-wise proximity among all pairs of points in the different clusters. This is an intermediate approach between the single and complete link approaches. The average linkage method defines spherically-shaped clusters.

Empirical Study: Consider a data set consisting of measurements of the variables Line of Code and Function for 15 Software Modules. The data set used is represented by the 15 x 2 data matrix.[9]

Table- 1 Summary Table for Software Modules

Software Modules	Line of Code	Function
SM 1	43	20
SM 2	29	21
SM 3	29	16
SM 4	47	23
SM 5	30	14
SM 6	21	11
SM 7	21	12
SM 8	18	9
SM 9	62	18
SM 10	23	12
SM 11	17	11
SM 12	20	19
SM 13	22	9
SM 14	15	6
SM 15	15	7

The table 6.1 is showing which module has how much amount of Line of Code (LOC) and FP after examine of each module.

Table- 2 Dissimilarity Matrix

Case	Euclidean Distance														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	14.036	14.56	5	14.318	23.77	23.409	27.313	19.105	21.541	27.514	23.022	23.707	31.305	30.871
2	14.036	0	5	18.111	7.071	12.806	12.042	16.279	33.136	10.817	15.62	9.22	13.892	20.518	19.799
3	14.56	5	0	19.313	2.236	9.434	8.944	13.038	33.061	7.211	13	9.487	9.899	17.205	16.643
4	5	18.111	19.313	0	19.235	28.636	28.231	32.202	15.811	26.401	32.311	27.295	28.653	36.235	35.777
5	14.318	7.071	2.236	19.235	0	9.487	9.22	13	32.249	7.28	13.342	11.18	9.434	17	16.553
6	23.77	12.806	9.434	28.636	9.487	0	1	3.606	41.593	2.236	4	8.062	2.236	7.81	7.211
7	23.409	12.042	8.944	28.231	9.22	1	0	4.243	41.437	2	4.123	7.071	3.162	8.485	7.81
8	27.313	16.279	13.038	32.202	13	3.606	4.243	0	44.911	5.831	2.236	10.198	4	4.243	3.606
9	19.105	33.136	33.061	15.811	32.249	41.593	41.437	44.911	0	39.459	45.541	42.012	41	48.508	48.27

10	21.541	10.817	7.211	26.401	7.28	2.236	2	5.831	39.459	0	6.083	7.616	3.162	10	9.434
11	27.514	15.62	13	32.311	13.342	4	4.123	2.236	45.541	6.083	0	8.544	5.385	5.385	4.472
12	23.022	9.22	9.487	27.295	11.18	8.062	7.071	10.198	42.012	7.616	8.544	0	10.198	13.928	13
13	23.707	13.892	9.899	28.653	9.434	2.236	3.162	4	41	3.162	5.385	10.198	0	7.616	7.28
14	31.305	20.518	17.205	36.235	17	7.81	8.485	4.243	48.508	10	5.385	13.928	7.616	0	1
15	30.871	19.799	16.643	35.777	16.553	7.211	7.81	3.606	48.27	9.434	4.472	13	7.28	1	0

Table shows the matrix of proximities between variables. These values represent the similarity or dissimilarity between each pair of items. For dissimilarities, larger values indicate items which are very different. Smaller values indicate items which are very similar. This relationship is reversed if a similarity measure is used.

Between Groups Linkage Agglomeration Schedule

Table (given below) shows how the cases are clustered together at each stage of the cluster analysis. The clusters are formed by merging cases and clusters; and the process continues, until all cases are joined in one big cluster. At each stage, one case or cluster is joined with another case or cluster. For instance, in this example, cases 6 and 8 are joined at stage 3. This is shown in the Clusters Combined columns. When clusters or cases are joined, they are subsequently labeled with the smaller of the two cluster numbers. The Coefficients column indicates the distance between the two clusters (or cases) joined at each stage. These values are based on the closeness among these clusters and linkage method is applied for this analysis. For a good cluster solution, a sudden jump in the distance coefficient (or a sudden drop in the similarity coefficient) is calculated. The stage before the sudden change indicates the optimal solution point for the merged clusters.

Table- 3 Between Groups Linkage Agglomeration Schedule

Agglomeration Schedule			
Stage	**Cluster Combined**		**Coefficients**
	Cluster 1	**Cluster 2**	
1	14	15	1
2	6	7	1
3	6	10	2
4	8	11	2
5	3	5	2
6	6	13	3
7	8	14	4
8	1	4	5
9	2	3	6
10	6	8	6
11	6	12	10
12	2	6	13
13	1	9	17
14	1	2	31

As shown in the above Table, first column of agglomeration schedule shows the number of stages which are total fourteen in numbers, because, if we have n number of modules then agglomerative method generates n-1 stages.

Second and third column shows the two clusters that are combined on similarity basis. In this table we see that cluster 14th and cluster 15th are very similar than other clusters which are shown by dissimilarity matrix. As such it is clear that when two clusters fuse with each other then the higher number cluster fuse with lower number cluster. So all fifteen cluster are combined to follow this procedure.

Fourth column shows the coefficient values that are generated by using Between Groups linkage method. These values describe how much the modules are similar on the basis of parameters LOC and Function Point. Coefficient values are helpful in strategy to discover the total number of clusters.

K-mean Method

K-Means clustering gives more stable clusters, since it is an interactive procedure compared with hierarchical methods. However, it needs a pre-specified number of starting points, to get an initial position. After obtaining initial cluster centers, the procedure is:

1. **Assigns cases to clusters based on distance from the cluster centers.**
2. **Updates the locations of cluster centers based on the mean values of cases in each cluster.**
3. **These steps are repeated until any reassignment of cases would make the clusters more internally variable or externally similar.**

It involves the following five stages:

- ✓ Initial Cluster Center
- ✓ Changes in Cluster Center
- ✓ Cluster Membership
- ✓ Final Cluster Center
- ✓ Each Cluster for Number of Software Modules

Final Cluster Center

Table: 4 Analysis of Final Cluster Centers.

	Cluster				
	1	2	3	4	5
LOC	45	21	16	62	29
FP	22	13	8	18	17

Table shows the vales for final clusters comprise the values for Line of code as well as the values for Function Points for final clusters. It is extracted that the value for each cluster is calculated by averaging the values of LOC and FP of modules.

Number of Software Modules in Each Cluster

Table shows the fifteen software modules that combined in nine clusters. As shown cluster 1 combine 2 software modules, cluster 2 combine 5 software modules, cluster 3 combines 4 software modules, cluster 4 combines 1 software module and finally cluster 5 is a combination of 3 software modules.

Table: 5 Analysis of Each Cluster for Number of Software Modules.

Cluster	1	2.000
	2	5.000
	3	4.000
	4	1.000
	5	3.000
Valid		15.000
Missing		.000

The results of all the tables shown above of this chapter are calculated through SPSS 13.0.

Efforts and Development Time before Clustering Approach

We collect the LOC of 15 modules and put these into the tool (COCOMO Basic) which gives the efforts (persons-month) and development time based on organic system (mentioned in introduction part). Cluster analysis is applied to this problem to reduce efforts of the persons and the development time so that the cost would be reduced and the system would be more effective and efficient in terms of efforts and development time. Clustering method combines the similar software modules in one cluster so that it would become time and cost efficient.

Table: 6 Calculations of Efforts and Development Time before Cluster analysis.

Module	Line of Code (LOC)	Efforts	Development Time
1	43	124.55	15.64
2	29	82.36	13.36
3	29	82.36	13.36
4	47	136.75	16.2
5	30	85.35	13.55
6	21	58.69	11.75
7	21	58.69	11.75
8	18	49.92	11.05
9	62	182.9	18.1
10	23	64.57	12.18
11	17	47	10.8
12	20	55.76	11.52
13	22	61.62	11.97
14	15	41.22	10.27
15	15	41.22	10.27
Total	412	1172.96	191.77

Efforts and Development Time after Clustering Approach

This time we collect LOCs and put these into the same tool (COCOMO Basic), after calculation we find the efforts and development time as shown in the table.

Table: 7 Calculations of Efforts and Development Time after Cluster Analysis.

Clusters	LOC	Efforts	Development Time
1	45	130.64	15.92
2	21	58.69	11.75
3	16	44.11	10.54
4	62	182.9	18.1
5	29	82.36	13.36
Total	173	498.7	69.67

III. Conclusion

In this paper we have presented a empirical study of the field of Hierarchical and K-mean clustering and found the reduction of cost and time after using clustering methods. Clustering methods seem a very good starting point for the automatic categorization of modules. This is because the goal of clustering methods is to group related entities. An algorithm is selected which will satisfy the constraints of a good categorization of modules.

The paper is also discussed in detail both Euclidean distance method for distance measure between two closest entities with the help of dissimilar matrix and agglomerative algorithm with the help of single and complete linkage method. Clustering technique plays an important role in data analysis and also extracts the most possible significant solution.

References

1. S. S. Parvinder, Madhu and S. Hardeep, “Automatic Categorzation of Software Modules” IJCSNS International Journal of Computer Science and Network Security,VOL.7 No.8, August 2007.
2. Popchev Ivan and Vania Peneva, “Cluster – A Package for cluster analysis” 10th annual international conference, 1988.

3. Wiggerts T.A., "Using Clustering Algorithms in Legacy Systems Remodularization" IEEE Fourth working conference on Reverse Engineering, pages 33-43, 06-08 oct. 1997.
4. Li Kai and Cui Lijuan, "Study of Models Clustering and Its Application to Ensemble Learning" Third International conference on Natural Computation (ICNC) 2007.
5. A. van Deursen and T. Kaupers, "Identifying objects using cluster and concept analysis" In Proceedings of the 21st International Conference on Software Engineering (ICSE 1999), Pages 246- 255. IEEE Compter Society, 1999.
6. Y. Cheng, "Mean Shift, mode seeking, and clustering" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, pp. 1197-1203, 1999.
7. J. Handl and J. Knowles, "Exploiting the tradeoff-the benefits of multiple objectives in data clustering" In Proc. 3rd International Conference on Evolutionary Multi-Criterion Optimization (EM'O05), vol. 3410 of NCS, pages 547-560, Springer-Verlag, 2005.
8. Mancoridis S., Mitchell B.S., Rorres C., Chen Y., Gansner E.R. "Using Automatic Clustering to Produce High-Level System Organizations of Source Code", IEEE International Workshop on Program Comprehension, pp 45-52, 1998.
9. Jalender B., Govardhan A., Premchand P., "Breaking the Boundaries for Software Components Reuse Technology", International Journal of Computer Applications, 13(6), 2011.
10. Fokaefs M., Tsantalis N., Chatzigeorgiou A., Sander J., "Decomposing Object-Oriented Class Modules Using an Agglomerative Clustering Technique", IEEE, Proc. ICSM, Canada, 2009.
11. Rothenberger M. A., Dooley K. J., Kulkarni U. R., Nada N., "Strategies for Software Reuse: A Principal Component Analysis of Reuse Practices", IEEE Transaction on Software Engineering, 29(9), 2003.
12. Kanungo T., Mount D. M., Netanyahu N. S., Piatko C. D., Silverman Wu A. Y., "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 2002.
13. Jeng J. J., Cheng B. H. C., "Using Formal Methods to Construct a Software Component Library", Proc. of 4th Eur. Soft. Eng. Conf., Lect Notes in Comp. Science, 717, 397-417, 1993.
14. Vesanto J., Alhoniemi E., "Clustering of the Self-Organizing Map", IEEE Transaction on Neural Networks, 11(3), 586-600, 2000.
15. Wiggerts T. A., "Using Clustering Algorithms in Legacy Systems Remodularization", IEEE, 33-47, 1997.
16. Jeng J. J., Cheng B. H. C., "Specification Matching for Software Reuse: A Foundation", ACM, 97-105, 1995.
17. http://mathworld.wolfram.com/ClusterAnalysis. html