ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (221-232), Month: July-September 2014, Available at: www.researchpublish.com

# COMPARATIVE STUDY OF DIFFERENT CLUSTERING AND DECISION TREE FOR DATA MINING ALGORITHM

[1]Ms. A. Sivasankari, [2] Mrs. S. Sudarvizhi, [3] S. Radhika Amirtha Bai

[1, 2, 3]Department of Computer Science, D.K.M College for Women, Vellore, TamilNadu, India

***Abstract:*** **Data Mining is a very interesting area to mine the data for knowledge. Several techniques are available. A data mining is one of the fast growing research field which is used in a wide areas of applications. The data mining consists of classification algorithms, association algorithms and searching algorithms. We are using clustering and decision tree methods to mine the data by using hybrid algorithms K-means, SOM and HAC algorithms from clustering and ID3 and C4.5 and CART algorithms from decision tree and it can produce the better results than the traditional algorithms. The comparative study of these algorithms to obtain which one is high accuracy, frequency, measure, procedure, Pruning, error rate and time complexities from the decision tree algorithm. At the same time the comparative study of these algorithms to obtain which one is quality, error rate, computation time, accessing time, performance, number of cluster, Map topology, type of software, type of data, data set size from the clustering algorithm.**

***Keywords*: K-MEANS, SOM, HAC, ID3, C4.5, CART**

## 1. INTRODUCTION

The comparative study to analyze the similarities and differences between the clustering algorithms and decision trees. K-means is one of the simplest unsupervised learning algorithms for clustering problems. The algorithm aims at forming k clusters of n objects such that the resulting intra-cluster similarity is high but the inter-cluster similarity is low.

The self-organizing maps (SOM) introduced by Teuvo Kohonen. The SOM are deemed as being highly effective as a sophisticated visualization tool for visualizing high dimensional, complex data with inherent relationships between the various features comprising the data. Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters. Bottom-up hierarchical clustering is therefore called hierarchical agglomerative Clustering or HAC. Top-down clustering requires a method for splitting a cluster. It proceeds by splitting clusters recursively until individual documents are reached.

In Decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm it is used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.

C4.5 is an algorithm used to generate a decision tree. C4.5 is an extension of ID3 algorithm. The decision trees generated by C4.5 can be used for classification. C4.5 is also called a statistical classifier. C4.5 using the concept of information entropy.

CART stands for Classification and Regression trees. It builds both classifications and regression trees. The classification tree construction by CART is based on binary splitting of the attributes. It uses many single-variable splitting criteria like gini index CART handles both categorical and continuous attributes to build a decision tree. It handles missing values**.**

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (221-232), Month: July-September 2014, Available at: www.researchpublish.com

## 2. CLUSTERING OVERVIEW

In this section the main categories of clustering algorithms will be presented (1) partitional, (2) hierarchical, (3) density-based and (4) grid-based clustering techniques.

Clustering is a form of unsupervised learning which involves the task of finding groups of objects which are similar to one another and different from the objects in another group. The goal is to minimize intracluster distances and maximize intercluster distances.

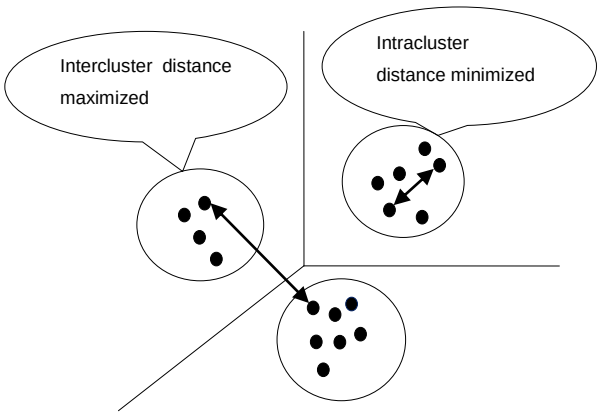


**Figure 2.1 Graphical representation of clustering**

### 2.1. K- means method

The **K-Means** algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the 'inertia' of the groups. This algorithm requires the number of cluster to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields. It is also equivalent to the expectation-maximization algorithm when setting the covariance matrix to be diagonal, equal and small. The K-means algorithm aims to choose centroids that minimize the within cluster sum of squares objective function with a dataset with samples.

1. Select k points as the initial centroids in a random way.
2. Assign all objects to the closest centroid.
3. Recalculate the centroid of each cluster.
4. Repeat steps 2 and 3 until a termination criterion is met.
5. Pass the solution to the next stage.

#### *2.1.1. Advantages*

1. K-means is computationally fast.
2. It is a simple and understandable unsupervised learning algorithm

#### *2.1.2. Disadvantages*

1. Difficult to identify the initial clusters.
2. Prediction of value of K is difficult because the number of clusters is fixed at the beginning.
3. The final cluster patterns is dependent on the initial patterns.

### 2.2. SOM

The self-organising maps (SOM) introduced by Teuvo Kohonen are deemed as being highly effective as a sophisticated visualization tool for visualizing high dimensional, complex data with inherent relationships between the various features comprising the data. The SOM's output emphasises the salient features of the data and subsequently lead to the automatic formation of clusters of similar data items. We argue that this particular characteristic of SOMs alone qualifies them as a potential candidate for data mining tasks that involve classification and clustering of data items. Kohonen's SOM is called a topology-preserving map because there is a topological structure imposed on the nodes in the network. A topological map is simply a mapping that preserves neighborhood relations. The Kohonen implementation produces the following output

- A weight matrix (list of weights for all nodes),
- A list of clusters with assigned data objects,
- The sum-of-square error for each cluster and for the whole partition where the square-error is measured as distance from the weight vector

#### *2.2.1. Advantages*

- Data mapping is easily interpreted
- Capable of organizing large, complex data sets

#### *2.2.2. Disadvantages*

- Difficult to determine what input weights to use
- Mapping can result in divided clusters
- Requires that nearby points behave similarly

### 2.3. HAC

A hierarchical method creates a hierarchical decomposition of the given set of data objects. Here tree of clusters called as dendrogram is built. Every cluster node contains child clusters, sibling clusters partition the points covered by their common parent. In hierarchical clustering we allocate each item to a cluster such that if we have N items then we have N clusters. Find closest pair of clusters and merge them into single cluster. Calculate distance between new cluster and each of old clusters. We have to repeat these steps until all items are clustered into K no. of clusters. It is of two types,

1. Agglomerative clustering

2. Divisive clustering

#### *I. Agglomerative (bottom up)*

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. It starts by letting each object form its own cluster and iteratively merges cluster into larger and larger clusters, until all the objects are in a single cluster or certain termination condition is satisfied. The single cluster becomes the hierarchy's root. For themerging step, it finds the two clusters that are closest to each other, and combines the two to form one cluster.

#### *II. Divisive (top down)*

A top-down clustering method and is less commonly used. It works in a similar way to agglomerative clustering but in the opposite direction. This method starts with a single cluster containing all objects, and then successively splits resulting clusters until only clusters of individual objects remain. Divisive clustering: It is a top-down clustering method which works in a similar way to agglomerative clustering but in the opposite direction. This method starts with a single cluster containing all objects and then successively splits resulting clusters until only clusters of individual objects remain.

1. Compute the proximity matrix containing the distance between each pair of patterns. Treat each pattern as a cluster.
2. Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.3. If all patterns are in one cluster, stop. Otherwise, go to step 2.

#### *2.3.1. Advantages*

The advantages of the hierarchical clustering algorithms are the reason this algorithm was chosen for discussion. These advantages include,

- Embedded flexibility regarding a level of granularity.
- Easy of handling of any forms of similarity or distance.
- Consequently applicability to any attributes types.
- Hierarchical clustering algorithms are more versatile.
- It ranks the objects for easier data display.
- Small clusters are obtained which is easier to analyze and understand.
- Number of clusters is not fixed at the beginning. Hence, user has the flexibility of choosing the clusters dynamically.

#### *2.3.2. Disadvantages*

- If objects are grouped incorrectly at the initial stages, they cannot be relocated at later stages.
- The results vary based on the distance metrics used.

### 2.4. Clustering benefits

Clustering provides

- Reduced single points of failure through Exchange Virtual Server (EVS) failover functionality.
- Ability to perform maintenance and upgrades with limited downtime.
- Ability to easily scale up your cluster to a maximum of seven active EVSs.

### 2.5. Clustering limitations

Clustering does not provide protection from

- Shared storage failures.
- Network service failures.
- Operational errors.
- Site disasters unless a geographically dispersed clustering solution has been implemented.

## 3. DECISION TREE OVERVIEW

Decision Trees are considered to be one of the most popular approaches for representing classifiers. Decision tree is a tree structure, where internal nodes denote a test on an attribute, each branch represents the outcomes of the test and the leaf node represents the class labels. In general, decision tree classifiers have good accuracy, but successful use of it depends on the data at hand. Decision trees are used for classification and classification rules are easily generated from them. Construction of decision trees is simple and fast, and does not need any domain knowledge and hence appropriate for exploratory knowledge discovery. The algorithm creates a node, and then applies attribute selection method to determine the best splitting criteria and the created node is named by that attribute.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (221-232), Month: July-September 2014, Available at: www.researchpublish.com

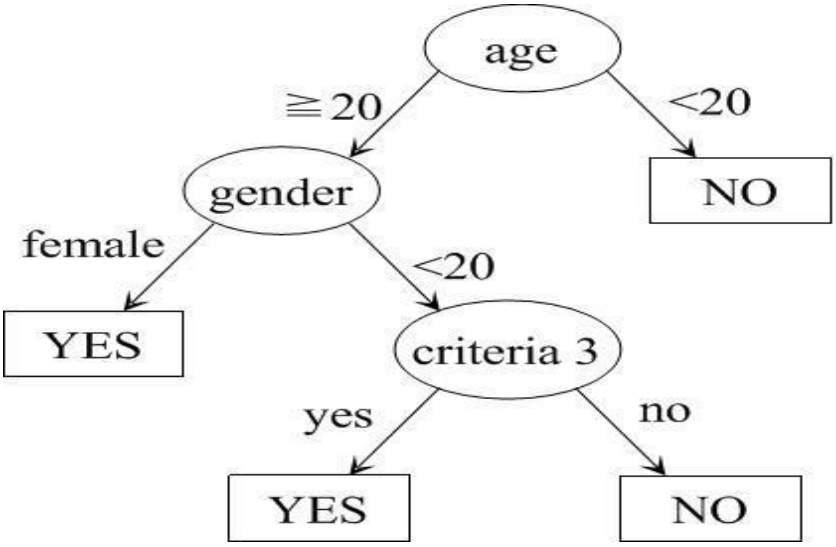


**Figure 3.1 Illustration of Decision Tree**

The decision tree is used in image processing for two main reasons.

1. Classification and
2. Regression.

There are various methods proposed in classification but some specific reasons to choose the DT for classification are

- v Provide human readable rules of classification
- v Easy to interrupt
- v Construction of the decision tree is fast
- v Yields better accuracy

v A Decision tree is constructed to model the classification process. Once the tree is built, it is applied to each tuple in the database and results in a classification for that tuple. There are two basic steps in the technique, (i) Building the tree and (ii) Applying the tree to the Database. For example is the illustration of a basic decision tree, where circle means decision node and square means leaf node, in this example, we have three splitting attributes age, gender and criteria 3, along with two class labels YES and NO each path from the root node to leaf node forms a classification rule.

### 3.1. Advantages of decision trees

Several advantages of the decision tree as a classification tool have been pointed out in the literature,

1. Decision trees are self–explanatory and when compacted they are also easy to follow. In other words if the decision tree has a reasonable number of leaves, it can be grasped by non professional users.
2. Furthermore decision trees can be converted to a set of rules. Thus, this representation is considered as comprehensible.
3. Decision trees can handle both nominal and numeric input attributes.
4. Decision tree representation is rich enough to represent any discrete–value classifier.
5. Decision trees are capable of handling datasets that may have errors.
6. Decision trees are capable of handling datasets that may have missing values.
7. Decision trees are considered to be a nonparametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

### 3.2. Disadvantages of decision trees

On the other hand, decision trees have such disadvantages as,

1. Most of the algorithms (like ID3 and C4.5) require that the target attribute will have only discrete values.
2. As decision trees use the "divide and conquer" method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present.
3. One of the reasons for this is that other classifiers can compactly describe classifier that would be very challenging to represent using a decision tree.

### 3.3. ID3

The ID3 algorithm is considered as a very simple decision tree algorithm (Quinlan, 1986). ID3 uses information gain as splitting criteria. The growing stops when all instances belong to a single value of target feature or when best information gain is not greater than zero. ID3 does not apply any pruning procedures nor does it handle numeric attributes or missing values.

#### *3.3.1. Entropy*

In order to define information gain precisely, we need to discuss entropy first. First, let's assume, without loss of generality, that the resulting decision tree classifies instances into two categories, we'll call them P (positive) and N (negative).

Information gain obtained by branching the training set on the attribute. The algorithm is recursively applied for the subsets till all the members of the set belongs to the same class.

Given a set S, containing these positive and negative targets, the entropy of S related to this Boolean classification is

Entropy(S)=

P(positive)log2P(positive) -

P (negative)log2P(negative)

P (positive): proportion of positive examples in S

P (negative): proportion of negative examples in S

For example, if S is (0.5+, 0.5-) then Entropy(S) is 1, if S is (0.67+, 0.33-) then Entropy(S) is 0.92, if P is (1+, 0-) then Entropy(S) is 0. Note that the more uniform is the probability distribution, the greater is its information.

You may notice that entropy is a measure of the impurity in a collection of training sets. But how it is related to the optimization of our decision making in classifying the instances. What you will see at the following will answer this question.

#### *3.3.2. Information gain*

As we mentioned before, to minimize the decision tree depth, when we traverse the tree path, we need to select the optimal attribute for splitting the tree node, which we can easily imply that the attribute with the most entropy reduction is the best choice.

We define information gain as the expected reduction of entropy related to specified attribute when splitting a decision tree node.

The information gain, Gain(S, A) of an attribute A,
Gain(S, A)= Entropy(S) –

Sum for v from 1 to n of (|Sv|/|S|) * Entropy (Sv)

We can use this notion of gain to rank attributes and to build decision trees where at each node is located the attribute with greatest gain among the attributes not yet considered in the path from the root.

The intention of this ordering is

1. To create small decision trees so that records can be identified after only a few decision tree splitting.

2. To match a hoped for minimalism of the process of decision making.

#### *3.3.3. Advantages of using id3*

- Understandable prediction rules are created from the training data.
- Builds the fastest tree.
- Builds a short tree.
- Only need to test enough attributes until all data is classified.
- Finding leaf nodes enables test data to be pruned, reducing number of tests.
- Whole dataset is searched to create tree.

#### *3.3.4. Disadvantage of using id3*

- Data may be over-fitted or over-classified, if a small sample is tested.
- Only one attribute at a time is tested for making a decision.
- Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

### 3.4. C4.5

C4.5 is an evolution of ID3, presented by the same author (Quinlan, 1993). It uses gain ratio as splitting criteria. The splitting ceases when the number of instances to be split is below a certain threshold. Error–based pruning is performed after the growing phase. C4.5 can handle numeric attributes. It can induce from a training set that incorporates missing values by using corrected gain ratio criteria as presented above.

#### *3.4.1. Pruning*

This important step to the result because of the outliers. All data sets contain a little subset of instances that are not well-defined, and differs from the other once on its neighborhood**.**

After the complete creation of the tree, that must classify all the instances in the training set, it is pruned. This is to reduce classification errors, caused by specialization in the training set; this is done to make the tree more general.

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of over fitting the data. Such methods typically use statistical measures to remove the least reliable branches, generally resulting in faster classification and an improvement in the ability of the tree to correctly classify independent test data.

There are two common approaches to tree pruning, which are pre-pruning and post- pruning. In the pre-pruning approach, a tree is "pruned" by halting its construction early by deciding not to further split or partitioned the subset of training samples at a given node. Upon halting, the node becomes a leaf.

In the post-pruning approach, a tree is pruned after it is "fully grown". A tree node is pruned by removing its branches. The lowest unpruned node becomes a leaf and is labeled by the most frequent class among its former branches.

The whole process of C4.5 algorithm is described in Algorithm 1. The information gain ratio criteria computation is performed in lines 11~21 using above equations, and a recursive function call is done in Line 25.

#### *3.4.2. Construction*

Some premises guide this algorithm, such as the following,

- If all cases are of the same class, the tree is a leaf and so the leaf is returned labeled with this class
- For each attribute, calculate the potential information provided by a test on the attribute based on the probabilities of each case having a particular value for the attribute. Also calculate the gain in information that would result from a test on the attribute based on the probabilities of each case with a particular value for the attribute being of a particular class
- Depending on the current selection criterion, find the best attribute to branch on.

#### *3.4.3. Advantages of c4.5 algorithm*

The algorithm C4.5 has following advantages,

- Handling attributes with different costs.
- Handling training data with missing attribute values-C4.5 allows attribute values to be marked as '?' For missing.
- Missing attribute values are simply not used in gain and entropy calculations.
- Handling both continuous and discrete attributes- in order to handle continuous attributes.
- C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.
- Pruning trees after creation- C4.5 goes back through the tree once it has been created and attempts to remove branches that do not help by replacing them with leaf nodes.

### 3.5. CART

CART stands for Classification and Regression Trees (Breiman *et al.*, 1984). It is characterized by the fact that it constructs binary trees, namely each internal node has exactly two outgoing edges.

The splits are selected using the towing criteria and the obtained tree is pruned by cost–complexity Pruning. When provided, CART can consider misclassification costs in the tree induction. It also enables users to provide prior probability distribution.

An important feature of CART is its ability to generate regression trees. Regression trees are trees where their leaves predict a real number and not a class. In case of regression, CART looks for splits that minimize the prediction squared error the least–squared deviation. The prediction in each leaf is based on the weighted mean for node.Usually the splitting algorithm will isolate outliers in individual node or nodes. An important practical property of CART is that the structure of its classification or regression trees is invariant with respect to monotone transformations of independent variables. One can replace any variable with its logarithm or square root value, the structure of the tree will not change.

CART methodology consists of three parts,

1. Construction of maximum tree
2. Choice of the right tree size
3. Classification of new data using constructed tree

#### *3.5.1. Gini splitting rule*

Gini splitting rule (or Gini index) is most broadly used rule. It uses the following impurity function i(t):

$$I(t) = \sum_{k \neq l} p(k|t)\, p(l|t)$$

where k, l 1, . . . , K - index of the class; p (k|t) - conditional probability of class k provided we are in node t. Applying the Gini impurity function 2.2 to maximization problem 2.1 we will get the following change of impurity measure $\Delta i(t)$: Gini

algorithm will search in learning sample for the largest class and isolate it from the rest of the data. Gini works well for noisy data.

### *3.5.2. Characteristics of the cart algorithm*

1. Each splitting is binary and considers one feature at a time.

2. Splitting criterion is the information gain or the Gini index.

### *3.5.3. Advantages of cart algorithm*

- Easy to use, understand.
- Produce rules that are easy to interpret and implement.
- Variable selection and reduction is automatic.
- Do not require the assumption of statistical models.
- Can work without extensive handling of missing data.

### *3.5.4. Disadvantages of cart algorithm*

- May not perform well where there is stricter in the data that is not well captured by horizontal or vertical splits.
- Since the process deals with one variable at a time, no way to capture interactions between variable.

## 4. RESULTS

### 4.1. CLUSTERING ALGORITHMS

I am using "Academic Activities" dataset it has real values there are approximately 15 descriptors and 5540 instances. Some of them will be used as an input attributes and other are used as an output attributes. After implementation of these algorithms on Academic Activities data set, the following results obtained.

**Table 4.1 Overall comparisons of clustering algorithms**

| S. No. | Features | K-Mean | SOM | HAC |
|---|---|---|---|---|
| 1. | Quality | 750 | 840 | 760 |
| 2. | Error rate | 0.8189 | 0.8456 | 0.8379 |
| 3. | Computation Time | 297MS | 1281MS | 1341MS |
| 4. | Accessing Time | FAST | SLOW | SLOW |
| 5. | Performance | 85 | 89 | 92 |
| 6. | Number of Cluster | 6 | 6 | 6 |
| 7. | Map Topology | 6 | 6 | 6 |
| 8. | Type of Software | LNKnet Package And Cluster and Tree view Package | LNKnet Package And Cluster and Tree view Package | LNKnet Package And Cluster and Tree view Package |
| 9. | Type of Dataset | 830 | 910 | 850 |
| 10. | Dataset Size | 89 | 95 | 91 |

### 4.2. DECISION TREE ALGORITHMS

The Primary tumor data is collected from UCI machine learning Repository and Colon tumor data is collected from Bioinformatics Group Seville, which are publicly available. The results were calculated and analyzed by Weka tool on the data using 10-fold cross validation to test the accuracy and time complexity of ID3, C4.5 and CART algorithms. The following results obtained.

**Table 4.2 Overall comparisons of decision tree algorithms**

| S. No. | Features | ID3 | C4.5 | CART |
|---|---|---|---|---|
| 1. | High accuracy | 50.27% | 79.12% | 77.29% |
| 2. | Frequency | 68 | 9 | 40.9 |
| 3. | Measure | Entropy<br>info- gain | Entropy<br>info- gain | Gini diversity index |
| 4. | Procedure | Top-down decision tree construction | Top-down decision tree construction | Constructs binary decision tree |
| 5. | Pruning | Pre pruning using a single pass algorithm | Pre pruning using a single pass algorithm | Post pruning based on cost complexity measure |
| 6. | Error rate | 0.4108 | 0.1937 | 0.3930 |
| 7. | Time Complexities | 1.41 | 4.34 | 47.83 |

## 5. CONCLUSION

As the number of clusters, K becomes greater; the performances of SOM algorithm become lower. The performance of K-means algorithms is better than hierarchical clustering algorithm.SOM algorithm shows more accuracy in classifying most the objects into their suitable clusters than other algorithms.K- Means have the less quality and accuracy of than the other.The quality of K-Means algorithm becomes very good results when using huge dataset.Hierarchical clustering and SOM algorithms show good results when using small dataset.Hierarchical clustering and SOM algorithms give better results compared to K-Means algorithm when using random dataset and the vice versa.K-Means are very sensitive for noise in dataset. This noise makes it difficult for the algorithm to cluster an object into its suitable cluster. This will affect the results of the algorithms.

C4.5 is the best algorithm for small datasets among all the three because it provides better accuracy and efficiency than the other algorithms. The main disadvantages of serial decision tree algorithm ID3, C4.5 and CART are low classification accuracy when the training data is large. But all these are used only for small data set.The experimentation result shows that the CART has the best classification accuracy when compared to ID3 and C4.5. The Experimental Results show that CART is the best algorithm for classification of medical data. It is also observed that CART performs well for classification on medical data sets of increased size.The time complexity of ID3 algorithm is less to build a model among the three classifiers. Coming to the accuracy, C4.5 and ID3 algorithms exhibit better accuracy than CART algorithm.

## REFERENCES

[1] Breiman, Friedman, Olshen, and Stone. "Classification and Regression Trees", Wadsworth, 1984. Mezzovico, Switzerland.

[2] L. Breiman, J. Friedman., R. Olshen, C. Stone,"Classification and Regression Trees", Wadsworth, Belmont, CA.

[3] J.R. Quinlan,"Decision trees and decision-making",IEEE Transactions on Systems, Man and Cybernetics,vol.20, no.2,(1990),pp.339-346.

[4] J.R. Quinlan,"C4.5: programs for machine learning", Morgan Kaufmann, (1993).

[5] J.R. Quinlan,"Improved use of continuous attributes in C4.5", arXiv preprint cs/9603103, (1996).

[6] J.R. Quinlan,"Induction of decision trees",Machine Learning, vol.1, no.1, (1986), pp.81-106.

[7] W. Wang, J. Yang, and R. Muntz, "Sting: A statistical information grid approach to spatial data mining," 1997

[8] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," pp. 94– 105, 1998

[9] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A multiresolution clustering approach for very large spatial databases," in Proc. 24th Int. Conf. Very Large Data Bases, VLDB, pp. 428– 439, 24–27 1998

[10] C. Fraley and A. Raftery, "Mclust: Software for model-based cluster and discriminant analysis," 1999

[11] L. Kaufman, P. J. Rousseeuw, "Finding Groups in Data: An introduction to Cluster Analysis," John Wiley and Sons, 1990.

[12] A.L.Radaideh, Q.A.AI-Shawakfa, E.M. AI-Najjar, "Mining student data using Decision Tree" International Arab Conference on Information Technology (ACIT 2006),Yarmouk University, 2006.

[13] Sunita B.Aher, L.M.R.J. lobo, "A Comparative study of classification algorithms", International Journal of Information Technology and Knowledge Management, July-December 2012, Volume 5, N0.2, pp 239-243.

[14] Matthew N.Anyanwu, Sajjan G.Shiva, "Comparative Analysis of Serial Decision Tree Classification Algorithms", International journal of Computer science and Security, volume 3.

[15] Breiman, Friedman, Olshen, and Stone. "Classification and Regression Trees", Wadsworth, 1984. Mezzovico, Switzerland.

[16] D.Lavanya, Dr.K.Usha Rani, "Performance Evaluation of Decision Tree Classifiers on Medical Datasets." International Journal of ComputerApplications 26(4):1-4, July 2011.

[17] J.R.Quinlan, "Induction of decision tree". Journal of Machine Learning 1, 1986, Pg.no:81-106.

[18] G Stasis, A.C. Loukis, E.N. Pavlopoulos, S.A. Koutsouris, D. "Using decision tree algorithms as a basis for a heart sound diagnosis decision support system", Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference, April 2003.

[19] Esposito F., Malerba D. and Semeraro G., "A Comparative Analysis of Methods for Pruning Decision Trees". EEE Transactions on Pattern Analysis and Machine Intelligence, 19(5):476-492, 1997.

[20] Tanagra Data Mining tutorials, http://data-mining-tutorials.blogspot.com/ this website provides detailed information on the basics of Data Mining Algorithms.

[21] WEKA, University of Waikato, New Zealand, http://www.cs.waikato.ac.nz/ml/weka/ (accessed July 18, 2012).

[22] J.R.Quinlan,"c4.5: Programs for Machin Learning", Morgan Kaufmann Publishers, Inc, 1992.

[23] Rakesh Agrawal , Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", Proceedings of the 20th International Conference on Very Large Data Bases, p.487-499, September 12-15, 1994.

[24] Fahim A. M., et al, "An efficient k-means with good initial starting points", Georgian Electronic Scientific Journal: Computer Science and Telecommunications, Vol. 2, No. 19, pp. 47-57,2009.

[25] Bashar Al Shboul et.al "Initializing K-Means Clustering Algorithm by using Genetic Algorithm", World Academy of Science, Engineering and Technology 54 2009.

[26] Primary tumor," http :// www. wisegeek. com/ what is- a-primary-tumor.htm"

[27] Colon Tumor," http: //www. wisegeek. Com /what-isa- colon-tumor.htm"

[28] UCI Machine Learning Repository, www.ics.uci.edu/~mlearn/MLRepository.html

[29] Bio informatic Group Seville, http://www.upo.es/eps/bigs/datasets.html