IJETST- Vol.||02||Issue||02||Pages 1897-1910||February||ISSN 2348-9480 2015

Open access Journal International Journal of Emerging Trends in Science and Technology

Classification of Clustering with Frequent item-sets

Authors
Z.Sunitha Bai[1], D.R.N.Sravana Lakshmi[2]
[1, 2]Assistant Professor, Department of Computer Science& Engineering
RVR & JC CE,Chowdavarm,Guntur, AP, India
Email: *mithachief@gmail.com*

Abstract

Data mining is having techniques like clustering, classification and algorithm like frequent itemset mining algorithms. In this we discuss about clustering and classification techniques with its applications and frequent itemset mining algorithm with its applications.

Cluster is the process for set of objects in cluster that objects are nearer to the center of cluster and far to other cluster and clustering is used to place the data elements into groups. Classification is the collection of records as training set and each record is having attributes. Frequent itemset algorithms are used for mining the frequent items of the transactions of the databases.

Index Terms— *Data Mining, Clustering, Classification, Frequent itemset mining, Data Mining Applications*.

1. INTRODUCTION

Data mining is having some techniques and algorithms here some techniques and algorithms are identified with applications [1]. Data mining is having techniques like clustering, clustering and frequent item set mining algorithms [14].

Clustering is a collection of objects and it is the set of objects such that each object in the cluster is near to middle of that cluster and not near to the other cluster. Clustering is used in placing the data elements into connected groups. The clustering techniques examples are Biology, Information retrieval, Climate, Psycology, and Medicine, Business, identifying the nature problems, capturing information of spatial cells [14]. Another technique is in data mining is classification, It is the input data for a classification task is a collection of records called training set or as an instance, and this record having attributes. Classification techniques are used in getting student details from college information, Gene Expression of Data Classification, drug activity issue, grouping of text. Frequent itemset algorithms of data mining are the process of identifying frequent itemsets from the large set of database transactions [4]. These algorithms are used in finding of items from all the transactions of database and there are different algorithms specified in frequent itemset algorithms. Frequent item set algorithms are used in online shopping, Adverse Drug reaction Detection, super market analysis, Oracle Bone Inscription and military applications [1].

2. CLUSTERING AND ITS APPLICATIONS

A) About Clustering:

Cluster is the set of objects such that each object in the cluster is near to middle of that cluster and not near to the other cluster [10]. It is the form of a collection of objects and the objects are same at the one cluster and are not same to the objects having the other clusters. Cluster is defined as three curves don‛t form clusters since it become lighter into the noise, as does the form link of the two little rounded clusters [13].Clustering is a technique and it is used for inserting data essentials into associated groups not including proceeding knowledge of the group definitions [8]. Most of the clustering techniques are contain k-means clustering and expectation maximization (EM) clustering. Clustering is the form of separate data into clusters such as high intra-cluster similarity, low inter-cluster similarity informally and discover natural grouping between objects [10].Clustering is the form of dividing data into clusters and it shows inside structure of the data, for example: Market segmentation [10]. Clustering is used for preparing of other AI techniques for example: review of news and clustering techniques are also useful in knowledge discovery in data for example: fundamental rules, reoccurring patterns, and topics, etc [13].

Clustering analysis is a form of data analysis that is divided in the form of groups or subsets and some similarity between the objects of each cluster [10]. The process of dividing data in convenient parts use clustering process as knowledge tool and it is also used in wide selection of fields such as biology, medicine, psychology, economics, sociology, and astrophysics. Main purpose of clustering is dividing the any set of objects of the form as groups based on given feature [13].

B) Techniques in clustering :

1) K-means clustering Technique:

K-means clustering is a prototype-based, partitional clustering technique, that attempts to find a user-specified number of clusters (K), which are represented by their centroids.. In this techniques create a one-level partitioning of the data objects. Two of the most prominent are K-means and K-medoid are used. K-means defines a prototype in terms of a centroid, which is usually the mean of a group of points, and is typically applied to objects in a continuous n-dimensional space.K-medoid defines a prototype in terms of a medoid, which is the most representative point for a group of points, and can be applied to a wide range of data since it requires only a proximity measure for a pair of objects.

1.1. Properties:

K-means clustering Algorithm is easy, and we begin properties as follows:

1. Choose K initial centroids, where K is a user specified parameter.
2. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster.
3. Centroid of each cluster is then updated based on the points assigned to the cluster.
4. Repeat the algorithm and update steps until no point changes clusters, until the centroids remain the same.

1.2. Algorithm:

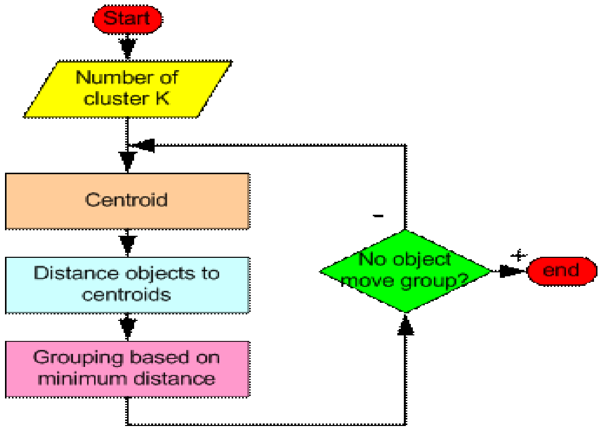
Input: K (no. of cluster)

Output: Given *k*, the *k-means* algorithm consists of four steps:

1. Select initial centroids at random.
2. Assign each object to the cluster with the nearest centroid.
3. Compute each centroid as the mean of the objects assigned to it.
4. Repeat previous 2 steps until no change.

1.3. K-means clustering example:

The basic step of k-means clustering is simple. we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects can also serve as the initial centroids


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
-
No object move group?
+
end

Step 1: Begin with a decision on the value of k = number of clusters.

Step 2: Put any initial partition that classifies the data into k -clusters. You may assign the training samples randomly, or systematically as the following:

1. Take the first k training sample as single-element clusters
2. Assign each of the remaining (N-k) training samples to the cluster with the nearest centroid.

3. After each assignment, recomputed the centroid of the gaining cluster.

Step 3: Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is no currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4: Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

Example:

Problem: Cluster the following eight points (with (x, y) representing locations) into three clusters A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9). Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2). The distance function between two points *a=(x1, y1)* and *b=(x2, y2)* is defined as: $\rho(a, b) = |x2 - x1| + |y2 - y1|$.

Use k-means algorithm to find the three cluster centers after the second iteration.

Solution:

Iteration 1

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(3, 10)				
A2	(2, 5)				
A3	(5, 9)				
A4	(5, 8)				
A5	(8, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(5, 9)				

First we list all points in the first column of the table above. The initial cluster centers – means, are (2, 10), (5, 8) and (1, 2) - chosen randomly. Next, we will calculate the distance from the first point (2, 10) to each of the three means, by using the distance function:

point	mean1
x1, y1	*x2, y2*
(3, 10)	(3, 10)

$$\rho(a, b) = |x2 - x1| + |y2 - y1|$$

$$\begin{aligned}\rho(point, mean1) &= |x2 - x1| + |y2 - y1| \\ &= |3 - 3| + |10 - 10| \\ &= 0 + 0 \\ &= 0\end{aligned}$$

Point	mean2
x1, y1	*x2, y2*
(3, 10)	(5, 8)

$$\rho(a, b) = |x2 - x1| + |y2 - y1|$$

$$\begin{aligned}\rho(point, mean2) &= |x2 - x1| + |y2 - y1| \\ &= |5 - 3| + |8 - 10| \\ &= 2 + 2 \\ &= 4\end{aligned}$$

point	mean3
x1, y1	*x2, y2*
(3, 10)	(1, 2)

$$\rho(a, b) = |x2 - x1| + |y2 - y1|$$

$$\begin{aligned}\rho(point, mean2) &= |x2 - x1| + |y2 - y1| \\ &= |1 - 3| + |2 - 10| \\ &= 2 + 8 \\ &= 10\end{aligned}$$

So, we fill in these values in the table:

		(3, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(3, 10)	0	4	10	1
A2	(2, 5)				
A3	(5, 9)				
A4	(5, 8)				
A5	(8, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(5, 9)				

So, which cluster should the point (2, 10) be placed in? The one, where the point has the shortest distance to the mean – that is mean 1 (cluster 1), since the distance is 0.

Cluster 1	Cluster 2	Cluster 3
(3, 10)		

So, we go to the second point (2, 5) and we will calculate the distance to each of the three means, by using the distance function:

point	mean1
x1, y1	*x2, y2*
(2, 5)	(3, 10)

$$\rho(a, b) = |x2 - x1| + |y2 - y1|$$
$$\rho(point, mean1) = |x2 - x1| + |y2 - y1|$$
$$= |3 - 2| + |10 - 5|$$
$$= 1 + 5$$
$$= 6$$

point	mean2
x1, y1	*x2, y2*
(2, 5)	(5, 8)

$$\rho(a, b) = |x2 - x1| + |y2 - y1|$$
$$\rho(point, mean2) = |x2 - x1| + |y2 - y1|$$
$$= |5 - 2| + |8 - 5|$$
$$= 3 + 3$$
$$= 6$$

point	mean3
x1, y1	*x2, y2*
(2, 5)	(1, 2)

$$\rho(a, b) = |x2 - x1| + |y2 - y1|$$
$$\rho(point, mean2) = |x2 - x1| + |y2 - y1|$$
$$= |1 - 2| + |2 - 5|$$
$$= 1 + 3$$
$$= 4$$

So, we fill in these values in the table:

Iteration 1

		(3, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(3, 10)	0	4	10	1
A2	(2, 5)	6	6	4	3
A3	(5, 9)				
A4	(5, 8)				
A5	(8, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(5, 9)				

So, which cluster should the point (2, 5) be placed in? The one, where the point has the shortest distance to the mean – that is mean 3 (cluster 3), since the distance is 0.

Cluster 1	Cluster 2	Cluster 3
(3, 10)		(2, 5)

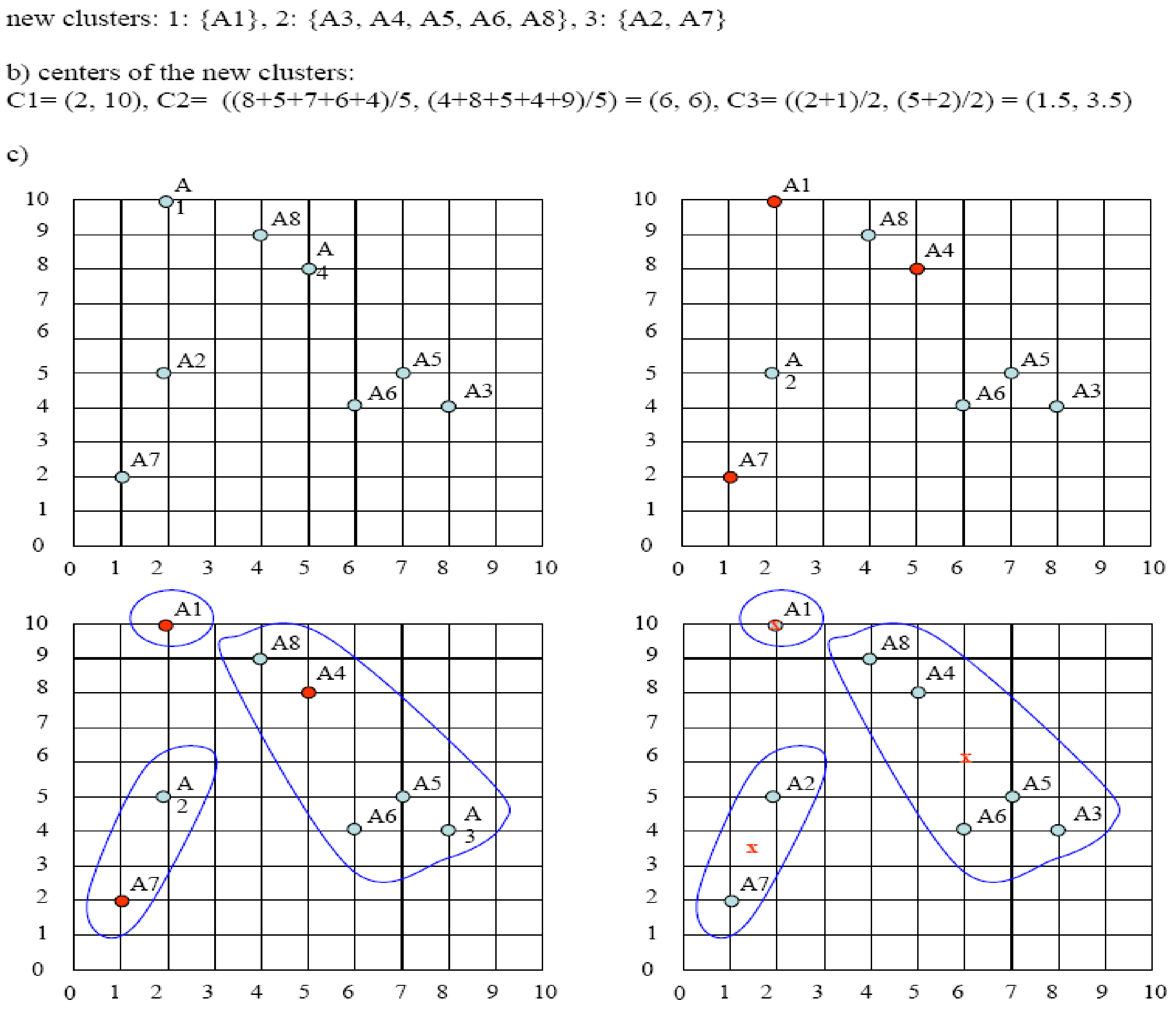
Analogically, we fill in the rest of the table, and place each point in one of the clusters: Iteration 1

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(3, 10)	0	4	10	1
A2	(2, 5)	6	6	4	3
A3	(5, 9)	3	1	11	2
A4	(5, 8)	5	0	10	2
A5	(8, 5)	10	6	10	2
A6	(6, 4)	9	5	7	2
A7	(1, 2)	10	10	0	3
A8	(5, 9)	3	1	11	2

Cluster 1	Cluster 2	Cluster 3
(3, 10)	(5, 9)	(2, 5)
	(5, 8)	(1, 2)
	(8, 5)	
	(6, 4)	
	(5, 9)	

Next, we need to re-compute the new cluster centers (means). We do so, by taking the mean of all points in each cluster.

For Cluster 1, we only have one point A1(3, 10), which was the old mean, so the cluster center remains the same.For Cluster 2, we have ((5+5+8+6+5)/5, (9+8+5+4+9)/5) = (5.8,9)For Cluster 3, we have ((2+1)/2, (5+2)/2) = (1.5, 3.5)



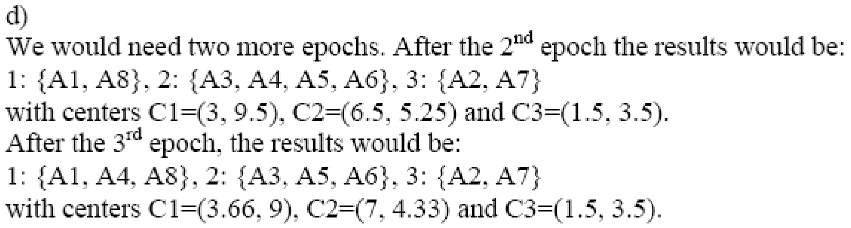
new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}

b) centers of the new clusters:
C1= (2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)

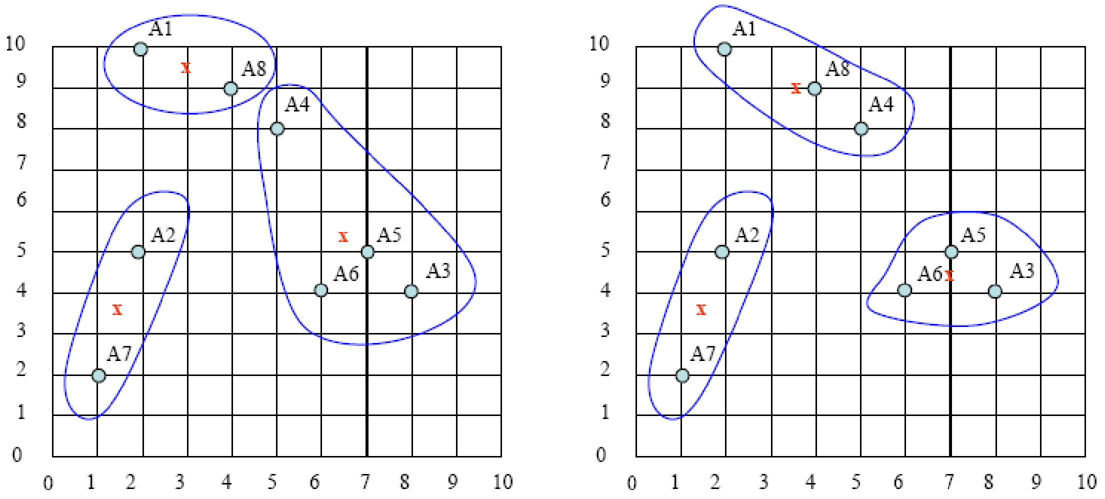
c)

The initial cluster centers are shown in red dot. The new cluster centers are shown in red x.

That was Iteration1 (epoch1). Next, we go to Iteration2 (epoch2), Iteration3, and so on until the means do not change anymore. In Iteration2, we basically repeat the process from Iteration1 this time using the new means we computed.



d)
We would need two more epochs. After the 2nd epoch the results would be:
1: {A1, A8}, 2: {A3, A4, A5, A6}, 3: {A2, A7}
with centers C1=(3, 9.5), C2=(6.5, 5.25) and C3=(1.5, 3.5).
After the 3rd epoch, the results would be:
1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7}
with centers C1=(3.66, 9), C2=(7, 4.33) and C3=(1.5, 3.5).



1.4. ***Applications:***

14.1. It is relatively *efficient and fast.* It computes result at **O(tkn),** where n is number of objects or points, k is number of clusters and t is number of iterations.

1.4.2.**.** k-means clustering can be applied to *machine learning or data mining.*

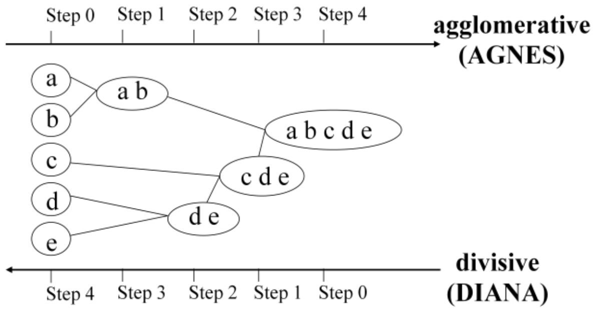
1.4.3. Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation)

1.4.4.Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.

2) Hierarchical Clustering Technique:

The process of hierarchical method is used for creating the hierarchical breakdown of the given set of data objects and the trees of clusters are also built which called as dendogram [21]. In this method each cluster node having the child clusters and clusters division of the points identified by their frequent parent. In this process of hierarchical clustering each cluster having item for that if we contain N items then we get N clusters and we have to get nearest two kinds of clusters and combine them into one cluster. Calculate distance between the new cluster and every of previous clusters for that we repeat these steps up to all items are clustered into K no. of clusters [8].

Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition.



2.1) Agglomerative (bottom up) :

It is bottom up method and in this method each cluster is having the sub-clusters and which in turn having sub-trees [10]. Here the procedure is starts with singleton cluster and recursively add two or more suitable cluster. In this method all objects are form as a single cluster and this single cluster is hierarchical root of the cluster after merging the step it for nearer to each other and add the two for appear as one cluster. The process stops when we get original cluster [13].

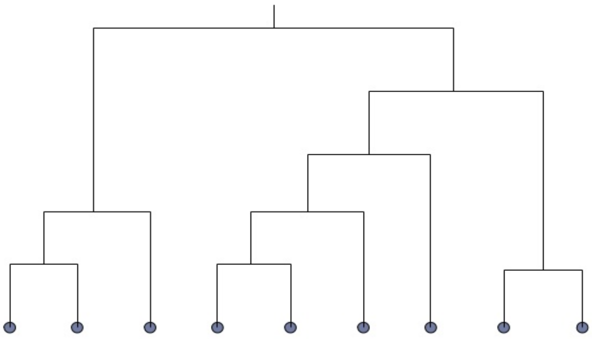
2.2) Divisive (top down):

Divisive clustering technique is top-down method and this clustering technique is not as much of used because of it works same as like agglomerative clustering but in the reverse way. This clustering technique is as top down process, the procedure is take the big cluster and starts with big cluster and divides that as smaller clusters. The process is stops when we get the original cluster [13].

2.3) A *Dendrogram* Shows How the Clusters are merged hierarchically:

2.3.1) Decompose data objects into several levels of nested partitioning (tree of clusters), called a dendrogram.

2.3.2) A clustering of the data objects is obtained by cutting the dendrogram at the desired level. Then each connected component forms a cluster.



2.3) Applications:

i) Adding items in shopping complex

This clustering is having application and this application is used as form top to bottom and bottom to top and the process of clustering having these two approaches [19]. We used these two approaches as add the items from anywhere in shopping complexes and the process of adding involves two approaches.

CLASSIFICATION AND ITS APPLICATIONS

3.1.1) About classification

Classification is machine learning technique and it is used to predict collection of membership for data instance. One of the best examples is classification to predict weather of one day will be ―sunny‖, ―rainy‖ or ―cloudy‖ [3]. Most of the classification techniques consist of decision trees and neural networks. It is the process of classifying the data based on the training set and values in that it is classify the attribute and uses these values in new classifying data [4].Classification is the process to form a collection of records this record is called training set. In this every record has contain a group of attributes and one of the attribute is class. Classification is used to find a representation for class attribute in the form of function and values of other attributes.

Goal of classification is assigned a class as exactly as possible for the previously unseen records. To verify the correct data of the model use test set and usually given data set is separated into training and test sets. This training set used to construct representation and test set used to confirming and the problem is a failure when the class is numerical.

3.1.2 Techniques in classification

1) Decision Tree Induction:

It is the process of classifying the instances by categorization them based on feature values and it is very easy but powerful learning model [2]. The procedure in this method is having a set of training set and it is breaks into small subsets and at the same time expand decision tree. At this end of the learning process or induction method decision tree is manage all the training sets and the training set is returned. In the decision tree every node represents the feature of an instance after the process of classification [21].

1.1) Characteristics:

1. Decision tree having the Attribute-value opposite elements
2. Separate the target function
3. Decision tree is having disjunctive data of target function
4. Any missing or erroneous training data is find then works well with that one

The following figure shows the decision tree process, the example is whether report in the form of tree.

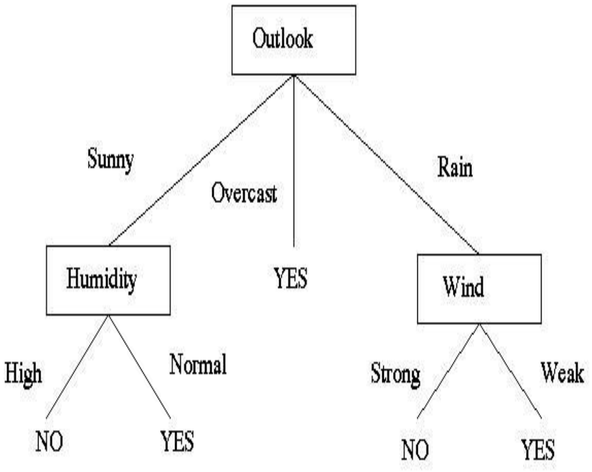


Fig: Decision Tree

(Outlook = Sunny $\wedge$ Humidity = Normal) $\vee$ (Outlook = Overcast) $\vee$ (Outlook = Rain $\wedge$ Wind = Weak)

1.2) Algorithm:

```
1. Create a node N;
2. If samples are all of the same class, C then
3. Return N as a leaf node labeled with the class C;
4. If attribute-list is empty then
5. Return N as a leaf node labeled with the most common class in samples;
6. Choose test-attribute, the attribute among attribute-list with the highest information gain; Label node N with test-attribute;
7. For each known value $a_i$ of test-attribute
9. Produce a branch from node N for the condition test-attribute=$a_i$;
10. Let $s_i$ be the set of samples for which test -attribute= $a_i$;
11. If $s_i$ is empty then add a leaf labeled with the most common in samples;
12. Else add the node returned by Generate_decision_tree ($s_i$, attribute-list_test-attribute)
```

The above algorithm for creating the decision tree, as per the algorithm first create the node N and all the samples are in class as C and the node N is returned as a leaf node labeled with the class C. If attribute-list is null then return the node N as leaf node of the common class in samples and choose the test-attribute among all the attribute-list for get main information [3]. Now node N is with the test-attribute for each value a_i of test-attribute and produce division from node N for the condition test-attribute a_i . Let s_i be the sample set for each test attribute and if s_i is unfilled then add leaf labeled with frequent class in samples or else add the node returned by generating decision tree [4]. Decision trees are based on the only one quality at each internal node, by this decision trees are regularly unvaried and most of the decision tree algorithms are not perform well in all the problems because that are need diagonal partitioning. [3] Decision trees considerably more complex representation due to the duplication problem and the solution to avoid duplication we implement the algorithm in complex feature at the nodes [3].

1.3) Applications:

i) Getting student information:

This application is used for finding information of student from any branch in total details in college. The process in this application is select the student name by name and the details as shown like if his percentage is represent in various grades. This application is also used for get the student from all students from college details and search by his id given by college [19].

ii) Selection of item in stores:

This application is used in general stores like the decision is applied for customer purchasing process. In this process customer purchase product by having options like good quality and good look. These details are representing in the form of tree form, this representation is for customer understanding about the product.

2) K-nearest neighbor classifier:

This process based on learning by similarity [7]. It is having the different learning‘s as Eager learning is to clear data of target function on the complete training set and another learning is Instance- based learning is for saving the all training instances and assign target function for next instance. [7] Every training sample as a point in an n-dimensional space and total the training samples are saved in an n-dimensional pattern space.

2.1) Instance-based learning:

This learning method is one of the lazy-learning algorithms and in this procedure as generalization total process up to classification is performed [8]. This need fewer calculation time in training phases when compare to another algorithms and one of the best algorithm in this method is nearest neighbor algorithm [11].

Procedure InstanceBaseLearner (Testing Instances):

```
For each testing instance
{
Find the k most nearest instances of the training set
according to a Distance metric
Resulting Class= most common
class Label of the k nearest
instances
}
```

The above procedure is for the instance based learning process as per that for each instance of testing first finds the K of majority adjacent instance of the training set. The training set is a distance metric and the result class of this training test case is most common class [7].

Finally the process of instance base learning is getting the label of the K nearest instances. In the nearest neighbor all the instances are corresponds to the points in n-dimensional Euclidean space and the classification is deferred up to new instance is arrived [7].

2.2 Algorithm:

The algorithm of nearest neighbor classifier can be summarized as follows

1. Get the new sample by specifying a positive integer K
2. We choose the K entries in our database which are closest to the new sample
3. By using all of these entries we find the most common classification
4. This is the classification we give to the new sample The above algorithm is for getting the new sample in the database of all the instances as per that first we specify the positive integer K along with new sample [7]. After getting new sample choose the K entries in our database which are neighboring to the new sample in database of all samples. By this process we get the all entries, in these entries we find the most common classification and this classification is for giving new sample [7].

2.3) Applications:

i) Euclidean space:

This application is used in the space for identifying neighboring points in the space complexity process. In the procedure of finding points is a start form one point in space and the nearest points of that space are identified by using classification. Classification is used for getting the stored training examples in the tree and this having high effectiveness [19].

3) Bayesian Networks:

This Bayesian networks is represents in the form of graphical model for possibility relations between set of variables [5]. These networks having the forms like declaring the node it has no parents and declares the leaf node it has no children node. Another form is declared the node that it is not directly connected to another network and declares the two nodes are not depends on given condition-set [5].

3.1) Bayesian Classifier:

Bayesian classifier is consists of the evaluation of the uncertain probability distribution of every attribute in the given the class. It is define as a set of classes and a set of attributes and the training of the Bayesian Classifier is a simplified with attributes and here the attributes are conditionally independent [5].

$$P(C_j|V) \infty P(C_j)\prod P(v_i|C_j)$$

Bayesian classifier is works well with the process of complete databases and the methods are for classify the incomplete databases [5].

3.2) Bayesian Theorem:

The process of Bayesian theorem is can be implemented by using the Bayesian classification. The following formula evaluates the procedure of Bayesian theorem [5].

$$P\ (h/D) = P\ (D/h)\ P\ (h)/P\ (D)$$

Here D is the given training data and h is the posteriori probability of a hypothesis. P(h|D) is the Bayes theorem as shown in the above formula.

3.3) Application:

i) Identifying the set of items

This application is used for selecting the set of items from any shopping complex. In this items are in the form of sets and selecting procedure is in the form of probability. The probability items are selected form all the sets of items in any shopping complex [19].

4.FREQUENT ITEMSET MINING ALGORITHMS

A) about Frequent itemset Mining

It is the procedure for finding frequent itemsets from the transactional databases. These items of database is get from the all the transactions in the database and the items which are present in the itemset are called the high frequent itemsets [17]. The procedure of identifying association rule having two steps first step is FI in the database and second step is use the set of FI for generating the attractive patterns [17].

Database formation is also is significant for the effectiveness of resulting of the frequent item sets from all the transactions of database.[15] The procedure of finding frequent itemset is also including FCI (Frequent Closed Itemset) and this FCI process is to generate the entire set of nearest itemsets. The process of finding frequent itemsets starts after generating the tree form in this the itemset is categorizing every node will be defined as the node‗s head [15].Most essential problem in the data mining area is Frequent itemset Mining. The process of complete set of frequent itemset is unfortunately large due to redundancy when the minimum threshold value of all the items in the transactions of database.[16] Here the minimum threshold value of the item is find out from the all the items of transactions from the database. The main task of frequent itemset mining is to find all frequent items with minimum support of the threshold value from the entire transactions in database [17].

Frequent sets play an essential role in many Data Mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one of the most popular problems of all these. The identification of sets of items, products, symptoms and characteristics, which often occur together in the given database, can be seen as one of the most basic tasks in Data Mining.

The original motivation for searching frequent sets came from the need to analyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products (Agrawal et al., 1993). Frequent sets of products describe how often items are purchased together.

Formally let I be the set of items.

A transaction over I is a couple T = (tid, I) where tid is the transaction identifier and I is the set of items from I.

A database D over I is a set of transactions over I such that each transaction has a unique identifier. We omit I whenever it is clear from the context

A transaction T = (tid, I) is said to support a set X, if X C I. The cover of a set X in D consists of the set of transaction identifiers of transactions in D that support X. The support of a set X in D is the number of transactions in the cover of X in D. The frequency of a set X in D is the probability that X occurs in a transaction, or in other words, the support of X divided by the total number of transactions in the database. We omit D whenever it is clear from the context.

A set is called frequent if its support is no less than a given absolute minimal support threshold min_sup with 0≤min_supabs≤|D|. When working with frequencies of sets instead of their support, we use the relative minimal frequency threshold min_suprel, with 0≤min_suprel≤1. Obviously min_supabs = [min_suprel * |D|]. In this paper we will mostly use the absolute minimal support threshold and omit the subscript abs.

Let D be a database of transactions over a set of items I, and min_sup the minimal support threshold. The collection of frequent sets in D with respect to min_sup is denoted 4 by F(D, min_sup):={X C I | support(X, D)≥min_sup} or simply F if D and minsup are clear from the context.

Given a set of items I, a database of transactions D over I, and a minimal support threshold min_sup, find F (D, min_sup). In practice we are not only interested in the set of sets F, but also in the actual supports of these sets.

For example, consider the database shown in the following table over the set of items

I = {beer, chips, pizza, wine}:

Tid	Set of items
100	{beer, chips, wine}
200	{beer, chips }
300	{pizza, wine}
400	{ chips, pizza}
500	{beer, pizza }

The following table shows all frequent sets in D with respect to a minimal support threshold equal to 1, their cover in D, plus their support and frequency:

Set	Cover	Support	Frequency
{}	{100,200,300, 400,500}	5	100%
{beer}	{100,200,500}	3	75%
{chips}	{100,200,400}	3	75%
{pizza}	{300,500}	2	50%
{wine}	{100,300}	2	50%
{beer, chips}	{100,200}	2	50%
{beer, wine}	{100}	1	25%
{chips, wine}	{100}	1	25%
{beer, pizza}	{500}	1	25%
{chips, pizza}	{400}	1	25%
{wine, pizza}	{300}	1	25%

If we are given the support threshold min_sup, then every frequent set X also represents the trivial rule X=> {} which holds with 100% confidence.

The task of discovering all frequent sets is quite challenging. The search space is exponential in the number of items occurring in the database and the targeted databases tend to be massive, containing millions of transactions. Both these characteristics make it a worthwhile effort to seek the most efficient techniques to solve this task.

B) *Frequent Itemset Mining Algorithms:*

1)Apriori Introduction:

The introduction of the frequent set mining problem, also the first algorithm to solve it was proposed, later denoted as AIS. Shortly after that the algorithm was improved by R. Agrawal and R. Srikant and called Apriori. It is a seminal algorithm, which uses an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets.

1.P(I) I is not frequent

2.P(I+A) I+A is not frequent either

3. Antimonotone property – if a set cannot pass a test, all of its supersets will fail the same test as well.

In the next, we will see how the apriori property is used in the Apriori algorithm: Let us look at how Lk-1 is used to find Lk, for k>=2. We can distinct two steps: **join and prune.**

1.1)Join :

1. Finding Lk, a set of candidate k-itemsets is generated by joining Lk-1 with itself
2. The items within a transaction or itemset are sorted in lexicographic order
3. For the (k-1) itemset: li[1]<li[2]<…
4. The members of Lk-1 are joinable if their first(k-2) items are in common
5. Members l1, l2 of Lk-1 are joined if (l1[1]=l2[1]) and (l1[2]=l2[2]) and … and (l1[k-2]=l2[k-2]) and (l1[k-1]-no duplicates.
6. The resulting itemset formed by joining l1 and l2 is l1[1], l1[2],…, l1[k-2], l1[k-1], l2[k-1]

1.2) Prune:

1. Ck is a superset of Lk, Lk contain those

candidates from Ck, which are frequent

2. Scanning the database to determine the count of each candidate in Ck – heavy computation
3. To reduce the size of Ck the Apriori property is used: if any (k-1) subset of a candidate k-itemset is not in Lk-1, then the candidate cannot be frequent either,so it can be removed from Ck. – subset testing (hash tree)

Let us take the following example:

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3,15
T900	I1, I2, I3

The join and prune steps for this example:

1. Scan D for count of each candidate

¡ C1: I1 – 6, I2 – 7, I3 -6, I4 – 2, I5 – 2

2. Compare candidate support count with minimum support count (min_sup=2)

¡ L1: I1 – 6, I2 – 7, I3 -6, I4 – 2, I5 - 2

3. Generate C2 candidates from L1 and scan D for count of each candidate

¡ C2: {I1,I2} – 4, {I1, I3} – 4, {I1, I4} – 1, …

4. Compare candidate support count with minimum support count

¡ L2: {I1,I2} – 4, {I1, I3} – 4, {I1, I5} – 2, {I2, I3} – 4, {I2, I4} - 2, {I2, I5} – 2 5.Generate C3 candidates from L2 using the join and prune steps

¡ Join: C3=L2xL2={{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}

¡ Prune: C3: {I1, I2, I3}, {I1, I2, I5}

6. Scan D for count of each candidate

¡ C3: {I1, I2, I3} - 2, {I1, I2, I5} – 2

7. Compare candidate support count with minimum support count

¡ L3: {I1, I2, I3} – 2, {I1, I2, I5} – 2

8. Generate C4 candidates from L3

¡ C4=L3xL3={I1, I2, I3, I5}

¡ This itemset is pruned, because its subset {{I2, I3, I5}} is not frequent => C4=null

2) The Apriori algorithm:

Input:

§ D, database of transactions;

§ min_sup, the minimum support count threshold

Output: L, frequent itemsets in D Method:

```
L1=find_frequent_1-itemsets(D);
for (k=2; Lk-1!=null;k++){
    Ck=apriori_gen(Lk-1);
    for each transaction t Є D{ // scan D for counts
        Ct = subset(Ck, t); // get the subsets of t that are candidates
        for each candidate c Є Ct
            c.count++;
    }
    Lk={c Є Ck | c.count≥min_sup}
}
Return L=UkLk
procedure apriori_gen(Lk-1: frequent(k-1)-itemsets)
for each itemset l1 Є Lk-1
    for each itemset l2 Є Lk-1
        if(l1[1]=l2[1])^(l1[2]=l2[2])^…^(l1[k-2]=l2[k-2])^(l1[k-1] then{
            c=l1xl2; //join step: generate candidates
            if has_infrequent_subset(c,Lk-1) then
                delete c; //prune step: remove unfruitful candidate
            else add c to Ck;
        }
Return Ck;
procedure has_infrequent_subset (c: candidate k-itemset; Lk-1: frequent (k-1)-itemsets);
//use priori knowledge
for each (k-1)-subset s of c
    if s !Є Lk-1 then
        Return TRUE;
Return FALSE;
```

2.1) Association rules generating from frequent itemsets:

Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them, where strong association rules satisfy both minimum support and minimum confidence. This can be done using the following equation:

Confidence (A=>B)=P(B|A) = support_count (AUB) / support_count(A)

The conditional probability is expressed in terms of itemset support count, where: support_count(AUB) is the number of transactions containing the itemsets AUB and support_count(A) is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:

§ For each frequent itemset l, generate all nonempty subsets of l.

§ For every nonempty subset s of l, output the rule “s => (l-s)” if support_count (l) / support_count(s)>=min_conf,

where min_conf is the minimm confidence threshold.

Let’s try an example based on the transactional data shown on Table 3. Suppose the data contain the frequent itemset l = {l1, l2, l5}. The nonempty subsets of l are: {l1, l2}, {l1, l5}, {l2, l5}, {l1}, {l2} and {l5}.

I1 and I2=>I5	Conf=2/4=50%
I1 and I5=>I2	Conf=2/2=100%
I2 and I5=> I1	Conf=2/2=100%
I1=>I2 and I5	Conf=2/6=33%
I2=>I1 and I5	Conf=2/7=29%
I5=>I1 and I2	Conf=2/2=100%

If the minimum confidence threshold is 70%, then only the second, third and last rules above are output, because these are the only ones generated that are strong.

2.2) Applications:

i) Adverse Drug reaction Detection (ADR)

The process of Apriori Algorithm is used in the health care data as Adverse Drug Detection and the characteristics of this algorithm are used to perform association analysis of patients [16]. The analysis of this algorithm is used to detection of drug and their primary diagnosis, by this analysis used in co-morbid conditions, and the adverse events got the experience. This algorithm produce association rules, these rules are used to indicate the combinations of medications and patient individuality lead to ADR [19].

ii) Oracle Bone Inscription (OBI)

The analysis of Apriori algorithm is used in the Oracle Bone Inscription (OBI), it oldest writing in the world. In this 6000 words found till now only about 1500 words explicated explicitly and so the open problem in this field is explication of OBI [19]. Association rule is used for the research in explication of OBI by explore the correlation between the words. Input for this algorithm is first extract data of OBI and we get the frequent itemset, by this we produce interesting measurements between the OBI words [19].

CONCLUSION

In this paper the discussion is about data mining techniques and algorithms with its applications [1]. Data mining is having some techniques like clustering, classification and frequent item set algorithms and we have to discuss about clustering and classification techniques with its applications [14]. In this we have also discuss about frequent itemset mining algorithms ofdata mining with its applications and these algorithms used for finding frequent items of transactions of database [17].

REFERENCES

1. Agrawal R., (Gehrke J., Gunopulos D., Raghavan P: ―Autima_iic Subspace Clustering of High Dimentional Data for Data Mining Applications‖. Proc. ACM SIGMOD‘98 Int. ―Conf. on Manigekent of Data, Seattle, WA, 1998, pp. 94-105.
2. Breslow, L. A. & Aha, D. W. (1997). Simplifying decision trees: A survey. *Knowledge Engineering Review 12:* 1–40.
3. Baik, S. Bala, J. (2004), A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection, Lecture Notes in Computer Science, Volume 3046, Pages 206 – 212.
4. Baik, S. Bala, J. (2004), A Decision Tree Algorithm for Distributed Data Mining:

Towards Network Intrusion Detection, Lecture Notes in Computer Science, Volume 3046,Pages 206 – 212.

5. Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach *Artificial Intelligence* 137: 43–90.
6. Clark, P., Niblett, T. (1989), The CN2 Induction Algorithm. Machine Learning, 3(4):261-283.
7. Digital Image Processing and Analysis-byB.Chanda and D.Dutta Majumdar.
8. Guha S., Rastogi R., Shim K.: ―CURE: An Efficient Clustering Algorithms for Large Databases‖, P-oc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, WA,1998, pp. 73-84.
9. J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. Proc. Conf. on the Management of Data (SIGMOD‘00, Dallas, TX), 1–12. ACM Press, New York, NY, USA 2000.
10. Jain A. K., Dubes R. C.: ―Algorithms for Clustering Data,‖Prentice-Hall, Inc., 1988.
11. M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining* (*KDD-96*), pp. 226-231, Portland, OR, USA, August 1996.
12. M. S. Chen, J. Han, P. S. Yu. Data mining: an overview from database perspective. To appear in *IEEE Transactions on Knowledge and data Engineering*, 1997.
13. learning tools and techniques", 2nd Edition, Morgan Kaufmann,San Francisco, 2005
14. J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108
15. H. Zha, C. Ding, M. Gu, X. He and H.D. Simon. "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.
16. R. Agarwal, C. Aggarwal, and V. V. V. Prasad: A Tree Projection Algorithm for Generation of Frequent Itemsets. Journal of Parallel and Distributed Computing (special issue on high performance data mining), (to appear), 2000.
17. *R. Agrawal and* R. Srikant. Fast Algorithms for Mining Association Rules. Proceedings. 20th Int. Conf. on very Large Databases (VLDB 1994, Santiago de Chile), 487–499. Morgan Kaufmann, San Mateo, CA, USA 1994.
18. S. Azad Razvi, Mr. S. Vikram Phaneendra|, Concise Range Queries with Efficient and Optimal Representation‖, IFRSA‘s International Journal Of Computing,Vol2,issue 3,July 2012, pp 657 - 662.
19. Wilson, D. R. & Martinez, T. (2000). Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning* 38:257–286.
20. Witten, I. & Frank, E. (2005), "Data Mining: Practical machine