An automated Data Mining System

M.s.c Jinan Ali
The Islamic University College

An automated Data Mining System

Jinan Ali
The Islamic University College

Abstract:-

Data mining is a new promising and attractive field in computer science. It extracts novel and useful information from large databases.

Data mining has many tasks and methods including mining association rules, classification, clustering, summarization and much more.

In order to apply the suitable method we need the help of experts to analyze data and choose an algorithm.

In this paper, we tried to design an automated data mining system that has the ability to fetch databases, analyze its attributes, and then apply the suitable data mining algorithm, and finally store results in the knowledgebase.

1. Introduction:

Data mining is a technique that discovers previously unknown relationships in data. Although data mining is a valuable technology for many application domains, it has not been widely adopted by business users at large and by the database community more specifically. This can be largely attributed to the nature of the data mining tasks. Data mining is a difficult and laborious activity that requires a great deal of expertise for obtaining quality results. It is also a multidisciplinary it often requires the expertise of numerous individuals working in the selection of techniques, creation of models, and parameter tuning in order to support analytical activities.[Camp05]

Despites its high claims and expectations, DM technology requires a highly trained professional to do an iterative, multistep process of accessing and preparing data, choosing an appropriate algorithm to mine the data, analyzing the learned knowledge, and presenting nontrivial, valuable knowledge to executives or decision makers.[Kerd07]

Many researchers agreed that data mining is a difficult and laborious activity that requires a great deal of expertise and a highly trained professionals to do an iterative, multistep process .

Each stage in data mining is very huge and consists of a lot of methods and algorithms. Data mining includes many tasks like classification, clustering , association, and others. Each one of these tasks includes a lot of algorithms, each with its own way of implementation.

A suggested solution to this fact is to automate the different steps and take important decisions instead of users, and to take use of the metadata file, Since "metadata is used by analysts in understanding data and building models and it can support the selection of suitable mining methods"[Nock02,Shi03, TCC05].

The massive increase in data was due to many factors as:

1-The appearance of new inexpensive high speed storage devices

2-The growth of computing power in general,

3-The availability of increased access to data from web navigation and intranets ,

4-The development of database management systems,

5-The expansion of business all over the world and others.

2. Data Mining Systems

Jiawei Han and Micheline Kamber explained the architecture of a typical data mining system having the following major components[Han06]:

1. Database, data warehouse, World Wide Web, or other information repository:

This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

2. Database or data warehouse server:

The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.

3. Knowledge base:

This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns.

4. Data mining engine:

This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as association analysis, classification, prediction, and cluster analysis.

5. Pattern evaluation module:

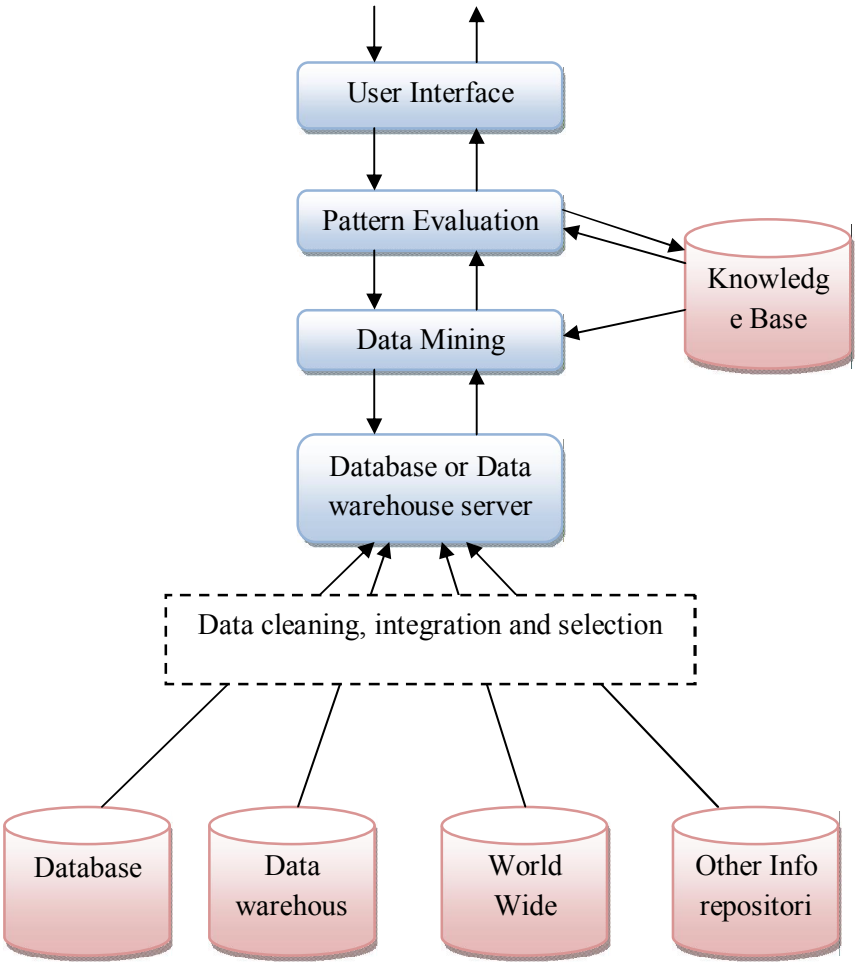
This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search toward interesting patterns.

6. User interface:

This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

Figure (1) depicts the components of a typical data mining system.

As outlined above, the data mining system process involves many steps. Furthermore, these steps require technologies from other fields. In particular, methods and ideas from machine learning, statistics, database systems, data warehousing and data visualization [Zaki03,Han06,Fayy96].



Figure(1) Data mining system

3. Data mining tasks:

Data mining methods` and tasks are used to specify the kind of patterns to be found in data mining tasks. In general, data

mining tasks can be classified into two categories: descriptive and predictive.

Descriptive mining tasks characterize the general properties of the data in the database.

Predictive mining tasks perform inference on the current data in order to make predictions.

There are many data mining methods(Table 1) , we choose to put in our system three of these methods, since they are from the most common and popular ones.

Table(1) Data mining tasks

DM Goals	DM Tasks
Prediction	Classification
	Regression
	Time-series analysis
Description	Clustering
	Summarization
	Association
	Sequence discovery

3.1. Mining Association Rules:

Since its introduction in 1993, the task of association rule mining has received a great deal of attention. Today the mining of such rules is still one of the most popular pattern-discovery methods.[Hipp00, Alba07]

The goal of association rule discovery is to find associations among items from a set of transactions, each of which contains a set of items. Generally the algorithm finds a subset of association rules that satisfy certain constraints.[Zhen01]

Apriori algorithm one of the most important algorithms in mining association rules.[Han06,Wu08,Wu09]

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties, as we shall see later.

By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. The number of items in an itemset is called its *size* and an itemset of size k is called a k-itemset. Let the set of frequent itemsets of size k be $F_k$and their candidates be C_k . Both F_k and C_k maintain a field, support count.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets.

First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted F_1.

Next, $F_1$is used to find F_2, the set of frequent 2-itemsets, which is used to find F_3, and so on, until no more frequent k-itemsets can be found. The finding of each F_k requires one full scan of the database.

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented below, is used to reduce the search space.

Apriori property: All nonempty subsets of a frequent itemset must also be frequent.

This property belongs to a special category of properties called antimonotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well. It is called antimonotone because the property is monotonic in the context of failing a test.

$C_{k+1}$is generated from $F_k$in the following two step process:

1. **Join step**: Generate C_{k+1}, the initial candidates of frequent itemsets of size k + 1 by taking the union of the two frequent itemsets of size k, $P_k$and Q_k that have the first k-1 elements in common.

$C_{k+1} = P_k \cup Q_k = \{item_1, \ldots, item_{k-1}, item_k, item_{\acute{k}}\}$

$P_k = \{item_1\ item_2, \ldots, item_{k-1}, , item_k\}$

$Q_k = \{item_1, item_2, \ldots item_{k-1}, , item_{\acute{k}}\}$

where, $item_1 < item_2 < \cdots <$, $item_k < item_{\acute{k}}$.

2. **Prune step**: To reduce the size of C_{k+1} the Apriori property is used as follows. Any (k)-itemset that is not frequent cannot be a subset of a frequent (k+1)-itemset. Hence, if any (k)-subset of a candidate (k+1)-itemset is not in F_k, then the candidate cannot be frequent either and so can be removed from C_{k+1} .

Apriori algorithm is shown in figure 2.

3**. Generating Association Rules 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 by calculating confidence:

$$confidence(X \Longrightarrow Y) = P(Y|X) = \frac{support\_count(X \cup Y)}{support\_count(X)} \quad \ldots 1$$

Where $support\_count(X \cup Y)$ is the number of transactions containing itemsets $X \cup Y$, and $support\_count(X)$ is the number of transactions containing itemset X.

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 " $s \rightarrow (l - s)$" if

$$\frac{support\_count(l)}{support\_count(s)} \geq minconf,$$

where $minconf$ is the minimum confidence threshold.

Algorithm: Apriori
Input: D, a database of transactions
Min_sup
Output: L, frequent itemsets in D

Method:
(1) L_1 = find frequent 1-itemsets(D);
(2) for (k = 2; $L_{k-1} \neq \emptyset$;k++) {
(3) C_k = apriori_gen(L_{k-1});
(4) for each transaction $t \in D$ { // scan D for counts
(5) C_t = subset(C_k, t); // get the subsets of t that are candidates
(6) for each candidate $c \in C_t$
(7) c.count++;
(8) }
(9) L_k={$c \in C_t$ | $c.count \geq min\ sup$}
(10) }
(11) return $L = \cup_k L_k$

procedure apriori gen(L_{k-1}:frequent (k-1)-itemsets)
(1) for each itemset $l_1 \in L_{k-1}$
(2) for each itemset $l_2 \in L_{k-1}$
(3) if (l_1[1] = l_2 [1])^(l_1[2] = l_2 [2])^:::^(l_1[k-2] = l_2 [k-2])^ l_1[k-1] = l_2 [k-1]then {
(4) $c = l_1 \bowtie l_{21}$; // join step: generate candidates
(5) if has infrequent subset(c, L_{k-1}) then
(6) delete c; // prune step: remove unfruitful candidate
(7) else add c to C_k;
(8) }
(9) return C_k;

procedure has infrequent subset(c: candidate k-itemset; L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
(1) for each (k-1)-subset s of c
(2) if $s \notin L_{k-1}$then
(3) return TRUE;
(4) return FALSE;

Figure(2) Apriori algorithm

3.2 Classification: Given a set of objects, each of which belongs to a known class, and each of which has a known vector of variables, classification techniques aim to construct a rule which will allow to assign future objects to

a class, given only the vectors of variables describing the future objects[Wu09].

Examples of classification tasks are:

- Determining whether a particular credit card transaction is fraudulent.
- Diagnosing whether a particular disease is present.

Bayesian algorithm is one of the algorithms in classification.

"What are Bayesian classifiers?" Bayesian classifiers (Figure 3) are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.

Bayesian classification is based on Bayes' theorem, described below. Studies comparing classification algorithms have found a simple Bayesian classifier known as the *naïve Bayesian classifier* to be comparable in performance with decision tree and selected neural network classifiers.

Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

Algorithm: Naïve Bayes
Input:
D: a data set containing *n* objects.
Output:
C: a selected class
Method:
For all values of the class attribute Ci
Calculate the prior probability $P(C_i)$
For all attribute values of tuple X (x_k):
Calculate the posterior probabilityof x_k condititioned on each class C_i $=P(x_k|C_i)$
For all values of the class attribute C_i:
Calculate the posterior probability of X given C_i $P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$
Calculate $P(X|C_i)P(C_i)$
For all values of the class attribute C_i:
return the largest value of $P(X|C_i)P(C_i)$

Figure(3) Baysian algorithm

3.3 clustering

This is also called unsupervised learning. Here, we are given a database of objects that are usually without any predefined categories or classes. We are required to partition the objects into subsets or groups such that elements of a group share a common set of properties. Moreover the partition should be such that the similarity between members of the same group is high and the similarity between members of different groups is low[Zaki03,Witt05]. Similarity can be determined using various distance functions[Garg06].Examples of clustering tasks include:

- gene expression clustering, where very large quantities of genes may exhibit similar behavior.

3.3.1 k-means Algorithm for clustering data:

The *k*-means algorithm takes the input parameter, *k*, and partitions a set of *n* objects into *k* clusters so that the resulting intracluster similarity is high but the intercluster similarity is low.

Cluster similarity is measured in regard to the *mean* value of the objects in a cluster, which can be viewed as the cluster's *centroid* or *center of gravity*.

"How does the k-means algorithm work?" The *k*-means algorithm proceeds as follows.

First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.

It then computes the new mean for each cluster. This process iterates until the criterion function converges.

X: a set of N data vectors Data set
C_I: initialized k cluster centroids Number of clusters,
C:the cluster centroids of k-clustering random initial centroids
$P = \{p(i) \mid i = 1, \ldots, N\}$ is the cluster label of X
KMEANS(X, C_I) _ (C, P)
REPEAT
$C_{previous}$ _ C_I;
FOR all $i \in [1, N]$ DO Generate new optimal paritions
$p(i)$ _ arg min $d(x_i, c_j)$;
l _ j _ k
FOR all $j \in [1, k]$ DO Generate optimal centroids
c_j _ Average of x_i, whose $p(i) = j$;
UNTIL $C = C_{previous}$ [18]

Figure(4) Kmean algorithm

4. The proposed automated data mining system

The proposed system process can be viewed as receiving a database as an input, and passing it in the different mining stages according to the database analysis results, where the analysis results depend mainly on the database metadata. Accordingly ,the software agent can choose the suitable mining task. In the work we used three most popular mining tasks , these are classification, clustering and association mining.

In order to implement this auto-miner system we used a separate metadata file approach, which is data about the data itself.

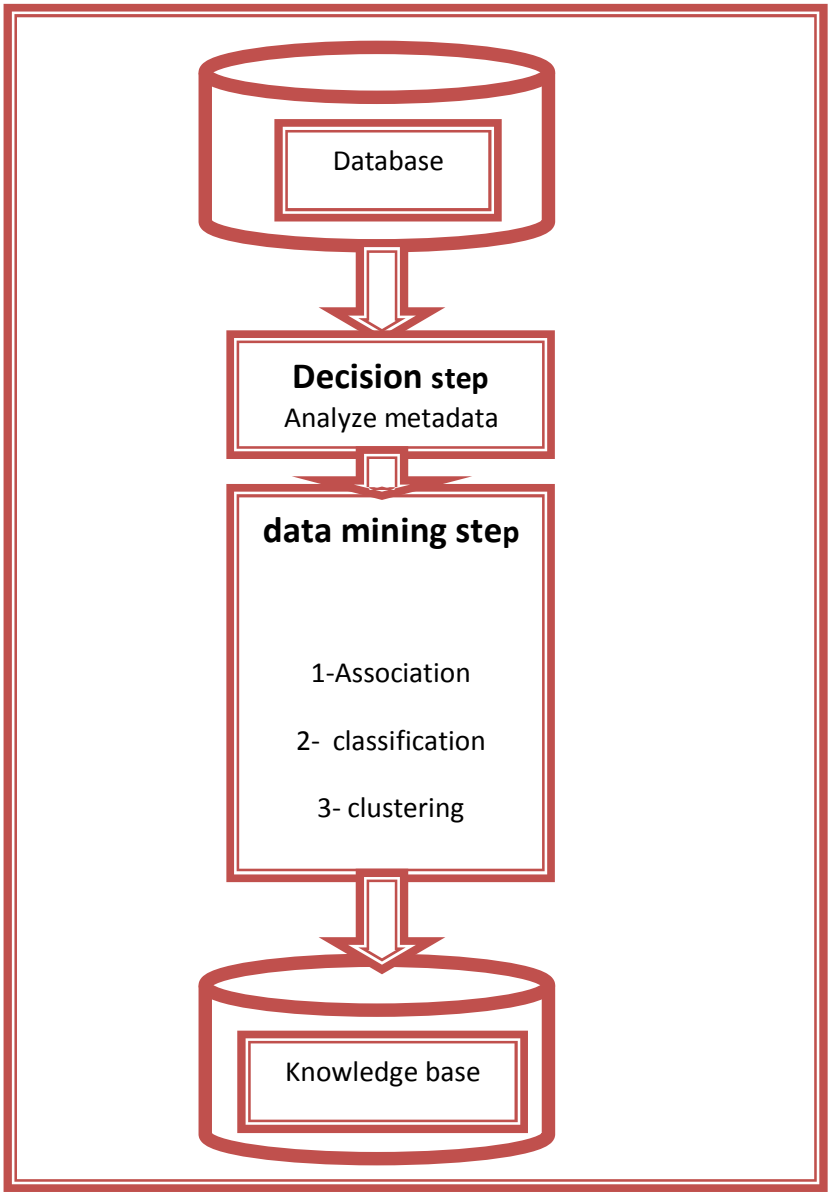
This metadata file includes:

- The name of the database attributes.
- Types of attributes.
- Values assigned for the categorical attributes.

The role of the system is to fetch data from the environment analyze the metadata file ,and decide which data mining task to choose, and then storing results in the knowledgebase file.

Borland Jbuilder9, a Java language development environment, was used for implementing the system.

The following figure present the design of the system:



Figure(5) The proposed system

4.1 Decision step

In this section, the database is ready to undergo the mining step, after it had been preprocessed.

The proposed system includes a suite of three data mining tasks among the most popular and important ones:

1. Classification
2. Clustering
3. Association Mining

For the current fetched database , the system chooses the best task automatically depending on a set of predefined rules. As it was mentioned in section 2, data mining tasks falls into two types descriptive and predictive. Classification is a descriptive method, while clustering and association mining are predictive ones. These facts can help perform the true decision.

4.2 Data mining step

After the suitable DM task has been chosen, the work now is to apply the corresponding algorithms.

The proposed system calls bayseian algorithm for classification, k-mean algorithm for clustering, and apriori algorithm for association mining.

4.2.1 Apriori Algorithm:

Input: database file

Output: frequent itemsets

In the implementation of Apriori, the values of minconf and minsup were prespecified.

Apriori algorithm generates frequent itemsets stored in arrays, then we implement genrules() function to generate rules from frequent itemsets.

The resulted frequent itemsets enter another procedure to generate association rules depending on minimum confidence.

Input: frequent itemsets

Output: Association Rules

The resulted association rules are stored in the knowledgebase file.

4.2.2 Naïve Bayes Algorithm

Input: Training database and test tuple

Output : Selected class.

In the proposed system, in the case where the agent select the classification task for a given database, it considers this database as the training one and ask for test datasets to give answers. The test data undergo set of calculations to estimate probabilities according to bayesian algorithm.

The test data tuple is represented by an *n*-dimensional attribute vector, $\boldsymbol{X}=(x_1, x_2, \dots, x_n)$ where n in the number of attributes.

The resulted class is stored absolutely in the knowledgebase file.

4.2.3 K-mean Algorithm

Input: Database file, k.

Output: Set of k clusters.

In the implementation of the proposed system, the value of k was suggested to be equal to three and the distance measure used is the city block distance:

$$d_1(x_i, x_j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}| \quad \dots\dots\dots\dots\dots\dots 2$$

The obtained k clusters are stored in the knowledge base file.

5. Experimental results

To accomplish the test of the designed system, Different types of databases with their metadata information were downloaded from different websites, some of which are professional in data mining research datasets .

5.1 Database 1:

The first database is the cereal database is available online with its attribute description at the following hyperlink http://lib.stat.cmu.edu/DASL/Stories/HealthyBreakfast.html.
This database illustrates different kinds of cereals and their nutritional composition.

The metadata information about the attributes and their types is shown in table (2).

Table(2) Database 1

Attribute Name	Attribute Type	Attribute values
Name	Discrete	-
Calories	Numeric	-
Protein	Numeric	-
Fat	Numeric	-
Sodium	Numeric	-
Fiber	Numeric	-
Carbo	Numeric	-
Sugars	Numeric	-
Vitamins	Numeric	-
Shelf	Numeric	-
Weight	Numeric	-
Cups	Numeric	-
Rating	Numeric	-

The system should decide which mining task or method to apply. For this reason it scanned the metadata file for attribute types, and it decided to choose the clustering task depending on the rules illustrated . The designed system stored the resulted clusters from the k-mean method in a knowledgebase file.

5.2 Database2:

The second database is car.mdb and is available online on the Machine Learning Repository UCI at the following hyperlink: http://archive.ics.uci.edu/ml/datasets/carevaluation. The metadata information about the attributes and their types is given in table (3).

Table (3) Database 2

Attribute Name	Attribute Type	Attribute values
Buying(buying price)	Discrete	vhigh, high, med, low
Maint(maintainace price)	Discrete	vhigh, high, med, low
Doors(number of doors)	Discrete	1,2,3,4,5more
Persons(capacity to carry persons)	Discrete	2,4,more
Lugboot(size of luggage boot)	Discrete	small, med, big
Safety(car estimated safety)	Discrete	low, med, high
Class(car accepatability)	Discrete	unacc, acc, good, vgood

The same steps as depicted in the previous examples, then the system asks for test data to classify.Then it stored the resulted class value from the bayseian method in a knowledgebase file.

6. Conclusions

Designing and implementing the automated data mining system resulted in several conclusions:

a. The designed system facilitates the work of both experts and non-expert users:

Non-experts can do things that cannot do usually.
Experts can do things more easily and without effort.

b. The proposed system does not need user interference or help since it is designed to work on behalf of user completely.

c. Concept of automation in data mining can be developed using intelligent methods and algorithms.

References:

[Alba07] Kamal Ali Albashiri, Frans Coenen, and Paul Leng, **An investigation into the issues of Multi-Agent Data Mining**, Doctoral thesis, University of Liverpool, 2007.

[Camp05] Macros M. Campos, Peter J. Stengard, and Boriana L. Milenova, **Data-Centric Automated Data Mining**, International conference on machine learning and applications, 2005.

[Fayy96] Usama Fayyad, Gregory Piatetsky, and Padhraic Smyth, **From Data Mining to Knowledge Discovery in Databases**, AI-magazine, vol.17, no.3, p.p.37-54, 1996.

[Garg06] Sanjay Garg and Ramesh Ghandra Jain **, Variations of K-mean Algorithm:A Study for High Dimensional Large Data Sets,** Information Technology Journal, 2006.

[Han06] Jiawei Han, Micheline Kamber, **Data Mining: Concepts and Techniques**, Elsevier, 2006.

[Hipp00] Jochen Hipp, Ulrich G¨untzer and Gholamreza Nakhaeizadeh, **Algorithms for Association Rule Mining – A General Survey and Comparison,** SIGKDD Explorations, 2000.

[Kerd07] Nittaya Kerdprasop, Kittisak Kerdprasop, **Moving Data Mining Tools toward a Business Intelligence System**, World Academy of Science, Engineering and Technology 25, 2007.

[Nock02] Thomas Nocke, Heidrun Schumann, **Meta Data for Visual Data Mining**, In the conference on computer graphics and imaging CGIM, 2002.

[Rama12] Nancy.P1, Dr.R. Geetha Ramani, " Discovery of Patterns and evaluation of Clustering Algorithms in Social Network Data (Face book 100 Universities) through Data Mining Techniques and Methods", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, No.5, 2012.

[Shi03] Zhongzhi Shi, **General Data Mining Platform-MSMiner**, Intelligent Science Research Group, 2003.

[TCC05] Introduction to Data Mining and Knowledge Discovery, Two Crows Corporation, 2005.

[Witt05] Ian H.Witten, Eibe Frank, **Data Mining: Practical Machine Learning Tools and Techniques**, Elsevier, 2005.

[Wu08] X.Wu, V.Kumar, J.R.Quinlan, J.Ghosh, Q. Yang, H. Motoda, G.Mclachlan, A.Ng, B.Liu, P.Yu, Z.Zhou, M.Steinbach, D. Hand, D.Steinberg , **Top 10 algorithms in data mining**, Knowledge and Information System KAIS, vol.14, no.1, pp.1-37, Springer –Verlag London Limited, 2008.

[Wu09] Xindog Wu and Vipin Kumar**, The Top Ten Algorithms in Data Mining,** Taylor & Francis Group LLC, 2009.

[Zaki03] Mohammed J.Zaki and Limsoon Wong, **Data Mining Techniques**, Lecture notes series, 2003.

[Zhen01] Zijian Zheng, Ron Kohavi and Llew Mason, **Real World Performance of Association Rule Algorithms**, KDD, 2001