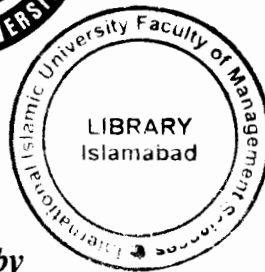


# K-MT, A CLUSTERING TECHNIQUE FOR DATA MINING

T00768



*Developed by*

**Muhammad Ali Mohsin**

*Supervised by*

**S. Tauseef-ur-Rehman**

**Department of Computer Science  
Faculty of Applied Sciences,  
International Islamic University, Islamabad  
(2003)**

**In the name of ALMIGHTY ALLAH,  
The most Beneficent,  
The most Merciful.**

**Department of Computer Science  
International Islamic University Islamabad**

**FINAL APPROVAL** 19 September, 2003

It is certificate that we have read the thesis submitted by Mr. Muhammad Ali Mohsin Reg. No. 04-CS/MS/01 and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the M.S Degree in Computer Science.

**Committee**

**External Examiner**

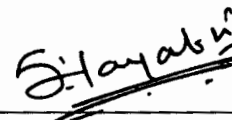
Dr. Nazir A Sangi  
Head of Department,  
Department of Computer Science,  
Allama Iqbal Open University,  
Islamabad



---

**Internal Examiner**

Dr. Sikander Hayat  
Head of Department  
Department of Computer Science,  
International Islamic University,  
Islamabad



---

**Supervisor**

S. Tauseef-ur-Rehman  
Assistant Professor  
Faculty of Applied Sciences,  
International Islamic University,  
Islamabad



---

19/9/2003

**A thesis submitted to the Department of Computer Science,  
International Islamic University, Islamabad as a partial  
fulfillment of the requirements for the award of the  
degree of M.S in Computer Science**

---

# **DEDICATION**

**Dedicated to Prophet Muhammad (Peace be upon Him)  
and my loving Family**

## **DECLARARTION**

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that we have developed this thesis and software entirely on the basis of our personal efforts made under the sincere guidance of our teachers. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Ali Mohsin  
04-CS/MS/01

## **ACKNOWLEDGEMENT**

All praise to Almighty Allah, the Most merciful and beneficial Who enabled me to complete this thesis. I pray to Almighty Allah that may this research work be used for the betterment of humanity and Muslim Ummah.

I like to express my highest gratitude and sincere thanks to my supervisor Mr. S.Tauseef-ur-Rehman and Dean Dr. Khalid Rashid who kept my moral high by there suggestions and appreciation. And last but not the least I like to acknowledge the support of my family and friends who helped me in every step of this thesis.

Muhammad Ali Mohsin  
04-CS/MS-01

## **PROJECT IN BRIEF**

Project Title:	<b>K-MT, a Clustering Technique for Data Mining</b>
Objective:	<b>To build an efficient Clustering Technique for Data Mining</b>
Undertaken By:	<b>Muhammad Ali Mohsin</b>
Supervised By:	<b>Mr. S. Tauseef-ur-Rehman Assistant Professor Department of Computer Science International Islamic University, Islamabad</b>
Technologies Used:	<b>Oracle 8.1.7 Oracle Developer 6i</b>
System Used:	<b>Celeron PIII 667 MHz</b>
Operating System Used:	<b>Windows 2000</b>
Date Started:	<b>October, 2002</b>
Date Completed	<b>September, 2003</b>



## **ABSTRACT**

Mining for information from databases has several important applications. Three of the most common methods to mine data are association rules, classification and clustering. Association rules derive patterns from grouped data attributes that co-occur with high frequency. Classification methods produce hierarchical decision models for input data that is sub-divided into classes. Finally, clustering methods group together the records of a data set into disjoint sets that are similar in some respect. Clustering also attempts to place dissimilar records in different partitions. Partitioning a large set of objects into homogeneous clusters is a fundamental operation in data mining. The K-Means technique is best suited for implementing this operation because of its efficiency in clustering large data sets. This thesis is a result of research done on different data mining algorithms (mainly clustering techniques). A new technique K-MT has been suggested which provides a better option for data mining. K-MT is a modifier of K-Means technique that was established back in 1975 by John Hartigen. Both these techniques are compared and implemented in Oracle 8i/ Developer and the results shows that K-MT is better option to choose while making clusters of data.

# TABLE OF CONTENTS

<i>Chapter No.</i>	<i>Contents</i>	<i>Page No.</i>
<b>1.</b>	<b>INTRODUCTION.....</b>	<b>2</b>
1.1	THE PROCESS OF KNOWLEDGE DISCOVERY .....	3
1.2	NEED OF KDD.....	5
1.3	APPLICATIONS OF KDD.....	7
1.4	KDD AND RELEATED FIELDS .....	7
1.5	DATA WAREHOUSE .....	8
1.6	ON LINE TRANSACTION PROCESS .....	8
1.8	DATA MINING .....	8
1.9	WHY DATA MINING? .....	9
1.10	USE OF DATA MINING.....	10
1.11	REASONS FOR GROWIG POPULARITY OF DATA MINING .....	11
1.11.1	Growing Data Volume .....	11
1.11.2	Limitations of Human Analysis .....	12
1.11.3	Low Cost Machine Learning.....	12
1.12	HOW DOES DATA MINING WORK? .....	12
1.13	APPLICATIONS OF DATA MINING .....	13
<b>2.</b>	<b>LITERATURE REVIEW.....</b>	<b>15</b>
2.1	EMERGENCE OF DATA MINING .....	16
2.2	CLUSTERING AT GALANCE .....	16
2.3	PARTITIONING CLUSTERING .....	16
2.3.1	K-Medioid Methods .....	17
2.3.2	K-Means Methods.....	17
2.3.3	Clustering and Continous K-Means Algorithm.....	18
2.3.4	Efficient Classificatino Algorithm for MultiSpectral Satellite Images .....	18
2.3.5	Spatial Data Mining Method .....	18
2.3.6	Clustering Catagorial Data. A Aproach Based on dynamical Systems .....	19
2.3.7	Refining Initial Poinits for K-Means Clustering .....	19
2.3.8	K-Modes Method .....	19
2.3.9	K-harmonic Means.....	19
2.3.10	Information Management and Process Improvemnet using Data Mining Techniques.....	19
2.4	DENSITY BASED .....	19
2.5	GRID BASED.....	20
2.5.1	BRICH .....	20
2.6	APRIORI ALGORITHM .....	20
2.7	BAYESIAN STRAUCTURAL EM ALGORITHM .....	20
<b>3.</b>	<b>DATA MINING TECHNIQUES.....</b>	<b>22</b>
3.1	ASSOCIATION RULE .....	22
3.1.1	The Basic Process Of Mining Assosiation Rule .....	22
3.1.2	When is Association Rule Analysis Useful? .....	22
3.1.3	Working of Association Rule .....	24
3.1.4	Algorithms Used in Association Rules .....	24
3.1.4.1	Apriori Algorithm.....	25
3.1.4.2	Distributed/Parallel Algorithm .....	25
3.2	SEQUENCE ANALYSIS.....	26

3.2.1 Working of Sequence Analysis .....	27
3.2.2 Algorithms Used in Sequence Analysis .....	27
3.2.2.1 AprioriAll Algorithm .....	28
3.2.2.2 AprioriSome Algorithm.....	28
3.3 CLASSIFICATION .....	29
3.3.1 Classification Methods.....	29
3.3.2 Classification Techniques .....	30
3.3.3 Understanding and Prediction .....	31
3.3.4 Algorithms Used in Classification .....	32
3.3.4.1 ID3 Algorithm .....	32
3.3.4.2 C4.5 Algorithm.....	32
3.3.4.3 SLIQ Algorithm .....	32
3.4 CLUSTERING .....	33
3.4.1 Working of Cluster Analysis.....	33
3.4.2 Uses of Cluster Analysis .....	34
3.4.3 Further Classification of Cluster Technique .....	34
3.4.3.1 Partitioning Methods .....	34
3.4.3.2 Hierarchical Methods .....	35
3.4.3.3 Density Based Methods.....	35
3.4.3.4 Grid Based Methods.....	35
<b>4. PROBLEM and ELUCIDATION .....</b>	<b>37</b>
4.1 K-MEANS.....	37
4.1.1 K-Means Clustering Method.....	38
4.1.2 Example .....	38
4.1.3 Strengths and Weaknesses .....	39
4.1.4 Variations of K-Means.....	39
4.2 K-MT.....	39
4.2.1 Variation in K-MT .....	39
4.2.2 K-MT Method.....	40
4.3 COMPARISON EXAMPLE .....	40
4.3.1 Problem .....	40
4.3.2 Solution by K-Means .....	40
4.3.3 Solution by K-MT .....	41
4.3.4 Conclusion .....	41
4.4 BISECTING STEP .....	41
4.5 ARCHITECTURE OF K-MT .....	42
4.6 FUTURE RESEARCH AREAS.....	43
<b>5. SOFTWARE TESTING AND RESULTS.....</b>	<b>45</b>
5.1 DATA SETS.....	45
5.2 MAIN.....	46
5.3 WHAT IS DATA MINING .....	47
5.4 ABOUT MT MINER.....	48
5.5 ABOUT THESIS .....	49
5.6 K-MT.....	50
5.7 K-MEANS.....	51
5.8 CREATE TABLES.....	52
5.9 LOADING DATA.....	53
5.10 THREE CLUSTERS.....	54
5.11 FOUR/FIVE/SIX/SEVEN CLUSTERS.....	55
5.12 MENUS .....	55
5.13 WORKING BAR.....	56
5.14 REPORTS .....	56

5.15 RESULTS ..... 57

**APPENDIX ..... 59**

**REFERENCES & BIBLIOGRAPHY ..... 60**

**ACCEPTANCE MAIL ..... 64**

**RESEARCH PAPER ..... 67**

## *List of Figures*

<i>Figure No.</i>	<i>Caption</i>	<i>Page No.</i>
1.1	PROCESS OF KNOWLEDGE DISCOVERY .....	3
1.2	KDD PROCESS ILLUSTRATED .....	4
1.3	DATA MINING PROCESS .....	9
1.4	DATA REDUCTION .....	11
2.1	STEPS IN THE EVOLUTION OF DATA MINING .....	15
3.1	CLUSTERS .....	33
4.1	K-MEANS CLUSTERS .....	38
4.2	ARCHITECTURE OF K-MT .....	42
4.3	FUTURE RESEARCH AREAS .....	43
5.1	MAIN SCREEN SHOT .....	46
5.2	WHAT IS DATA MINING SCREEN SHOT .....	47
5.3	ABOUT MT MINER SCREEN SHOT .....	48
5.4	ABOUT THESIS SCREEN SHOT .....	49
5.5	K-MT ALGORITHM SCREEN SHOT .....	50
5.6	K-MEANS ALGORITHM SCREEN SHOT .....	51
5.7	CREATE TABLES SCREEN SHOT .....	52
5.8	LOADING DATA SCREEN SHOT .....	53
5.9	THREE CLUSTERS SCREEN SHOT .....	54
5.10	FILE MENU SCREEN SHOT .....	55
5.11	ALGORITHMS MENU SCREEN SHOT .....	55
5.12	HELP MENU SCREEN SHOT .....	55
5.13	WORKING BAR SCREEN SHOT .....	56
5.14	REPORTS SCREEN SHOT .....	56
5.15	DIFFERENT RESULTS FROM DIFFERENT DATA SETS .....	57

# *Chapter 1*

## *Introduction*

# 1. INTRODUCTION

In general, we often see data as a string of bits, or numbers and symbols, or “objects” which are meaningful when sent to a program in a given format (but still uninterpreted). We use bits to measure information, and see it as data stripped of redundancy, and reduced to the minimum necessary to make the binary decisions that essentially characterize the data (interpreted data). We can see knowledge as integrated information, including facts and their relations, which have been perceived, discovered, or learned as our “mental pictures”. In other words, knowledge can be considered data at a high level of abstraction and generalization.

Knowledge discovery and data mining (KDD) the rapidly growing interdisciplinary field which merges together database management, statistics, machine learning and related areas aims at extracting useful knowledge from large collections of data. There is a difference in understanding the terms “knowledge discovery” and “data mining” between people from different areas contributing to this new field. Knowledge discovery in databases is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns/models in data. Data mining is a step in the knowledge discovery process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, finds patterns or models in data. In other words, the goal of knowledge discovery and data mining is to find interesting patterns and/or models that exist in databases but are hidden among the volumes of data.

The roots of KDD can be traced as far as statistics, but KDD as it exists today really began to take shape in early 1990s when a number of factors came together. In addition to the advancements in processing power and storage capabilities, network infrastructure, databases tools and maturity of the machine learning techniques for discovering pattern led to the birth of KDD. KDD technology encompasses the entire process of selecting, preparing, extracting and reviewing patterns extracted from data, while data mining focuses on the extraction step of this process.

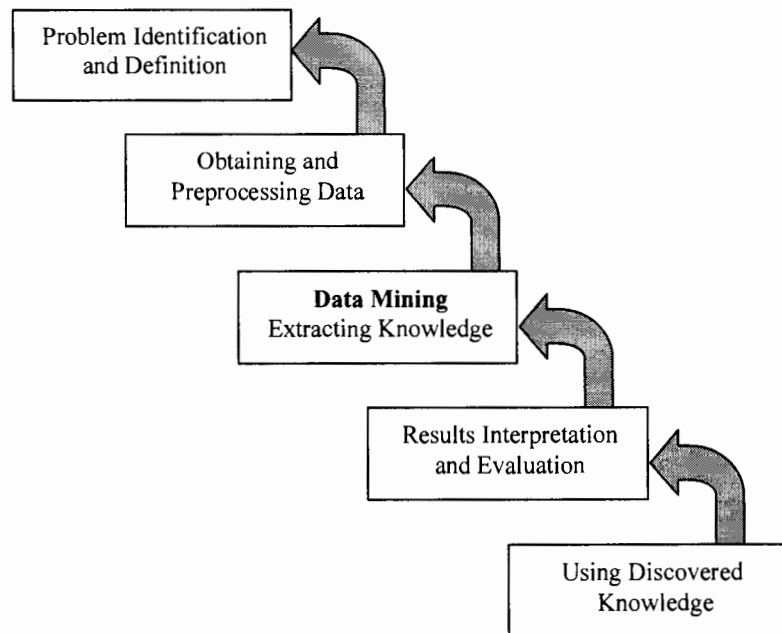
The Data Mining process can be completed with help of different algorithms. These algorithms are divided into 3 major classes. The main objective of this thesis is also these algorithms specially the clustering algorithms. K-MT algorithm is also of clustering type and is a modifier of K-Mean algorithm. These algorithms are discussed in the coming chapters of this thesis.

## 1.1 Problem Statement

The problems faced during the data mining processes vary from one another. Day by day as the data size grows, the techniques used for data mining purposes are becoming lesser effective. It is the need of the hour that we develop efficient techniques for data mining. These techniques should be more effective than the already present. They should take lesser time to execute while producing better results. During our research we made an efficient technique called as K-MT. Our research mainly evolved around the partitioning techniques used for clustering data.

## 1.2 The Process of Knowledge Discovery

The process of knowledge discovery inherently consists of several steps as shown in Figure 1.1.



**Fig 1.1 Process of Knowledge Discovery**

The **first step** is to *understand the application domain and to formulate the problem*. This step is clearly a prerequisite for extracting useful knowledge and for choosing appropriate data mining methods in the third step according to the application target and the nature of data.

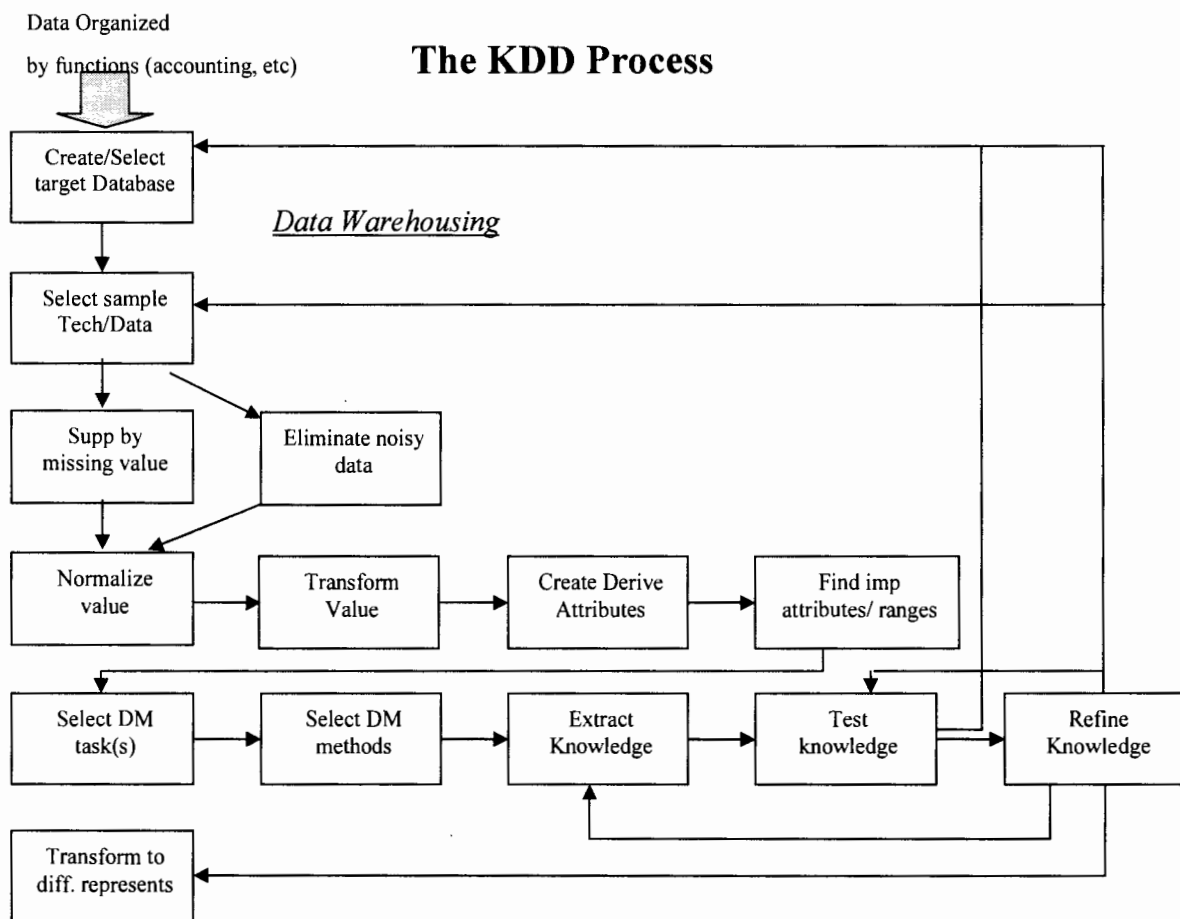
The **second step** is to *collect and preprocess the data*, including the selection of the data sources, the removal of noise or outliers, the treatment of missing data, the transformation (discrimination if necessary) and reduction of data, etc. This step usually takes the most time needed for the whole KDD process.

The **third step** is data mining that extracts patterns and/or models hidden in data. A model can be viewed “a global representation of a structure that summarizes the systematic component underlying the data or that describes how the data may have arisen”. In contrast, “a pattern is a local structure, perhaps relating to just a handful of variables and a few cases”. The major classes of *data mining methods* are *predictive modeling such as classification and regression; segmentation (clustering); dependency modeling such as graphical models or density estimation; summarization such as finding the relations between fields, associations, visualization; and change and deviation detection/modeling* in data and knowledge.



The **fourth step** is to interpret (post-process) discovered knowledge, especially the interpretation in terms of description and prediction—the two primary goals of discovery systems in practice. Experiments show that discovered patterns or models from data are not always of interest or direct use, and the KDD process is necessarily iterative with the judgment of discovered knowledge. One standard way to evaluate induced rules is to divide the data into two sets, training on the first set and testing on the second. One can repeat this process a number of times with different splits, and then average the results to estimate the rules performance.

The **final step** is to put discovered knowledge in practical use. In some cases, one can use discovered knowledge without embedding it in a computer system. Otherwise, the user may expect that discovered knowledge can be put on computers and exploited by some programs. Putting the results into practical use is certainly the ultimate goal of knowledge discovery.



Alternative names used in the past: data mining, data archaeology, data dredging, functional dependency analysis, and data harvesting. We consider the KDD process shown in Fig 1.2 in more details with the following tasks:

- **Develop understanding of application domain:** relevant prior knowledge, goals of end user, etc.
- **Create target data set:** selecting a data set, or focusing on a subset of variables or data samples, on which discovery is to be performed.
- **Data cleaning preprocessing:** basic operations such as the removal of noise or outliers if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, accounting for time sequence information and known changes.
- **Data reduction and projection:** finding useful features to represent the data depending on the goal of the task. Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.
- **Choose data mining task:** deciding whether the goal of the KDD process is classification, regression, clustering, etc...
- **Choose data mining method(s):** selecting method(s) to be used for searching for patterns in the data. This includes deciding which models and parameters may be appropriate (e.g., models for categorical data are different than models on vectors over the real numbers) and matching a particular data mining method with the overall criteria of the KDD process (e.g., the end-user may be more interested in understanding the model than its predictive capabilities).
- **Data mining to extract patterns/models:** searching for patterns of interest in a particular representational form or a set of such representations: classification rules or trees, regression, clustering, and so forth. The user can significantly aid the data mining method by correctly performing the preceding steps.
- **Interpretation and evaluation of pattern/models**
- **Consolidating discovered knowledge:** incorporating this knowledge into the performance system, or simply documenting it and reporting it to interested parties. This also includes checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

### 1.3 Need of KDD

There are many reasons that explain the need of KDD, typically

- Many organizations gathered *so much data*, what do they do with it?
- People store data because they think *some valuable assets are implicitly coded within it*. In scientific endeavors, data represents observations carefully collected about some phenomenon under study.
- In business, data captures information about critical markets, competitors, and customers. In manufacturing, data captures performance and optimization opportunities, as well as the keys to improve processes and troubleshooting problems.

- Only a small portion (typically 5%-10%) of the collected data is ever analyzed.
- Data that may never be analyzed continues to be collected, at great expense, out of fear that something which may prove important in the future is missed
- Growth rate of data precludes traditional “manual intensive” approach if one is to keep up.
- *Data volume is too large* for classical analysis regime. We may never see them entirety or cannot hold all in memory.
  - high number of records too large ( $10^8$ - $10^{12}$  bytes)
  - high dimensional data (many database fields:  $10^2$ - $10^4$ )
  - “how do you explore millions of records, ten or hundreds of fields, and finds patterns?”
- Networking, *increased opportunity for access*
- Web navigation on-line product catalogs, travel and services information, ...
- End user is *not a statistician*
- Need to *quickly identify and respond* to emerging opportunities before the competition
- Special financial instruments, target marketing campaigns, etc.
- As databases grow, ability to support analysis and decision making using *traditional (SQL) queries infeasible*:
- Many queries of interest (to humans) are difficult to state in a query language
  - e.g. “find me all records indicating frauds”
  - e.g., “find me individuals likely to buy product x?”
  - e.g., “find all records that are similar to records in table X”
- The *query formulation problem*
  - It is not solvable via query optimization
  - Has not received much attention in the database field or in traditional statistical approaches
  - Natural solution is via train-by-example approach (e.g., in machine learning, pattern recognition)

## 1.4 Applications of KDD

KDD techniques can be applied in many domains, typically

- Business information
  - Marketing and sales data analysis
  - Investment analysis
  - Loan approval
  - Fraud detection
- Manufacturing information
  - Controlling and scheduling
  - Network management

- Experiment result analysis
- Scientific information
  - Sky survey cataloging
  - Biosequence Databases
  - Geosciences: Quake finder
- Personal information

## 1.5 KDD and Related Fields

KDD is an interdisciplinary field that relates to statistics, machine learning, databases, algorithmic, visualization, high-performance and parallel computation, knowledge acquisition for expert systems, and data visualization. These systems typically draw upon methods, algorithms, and techniques from these diverse fields. The unifying goal is extracting knowledge from data in the context of large databases.

The fields of machine learning and pattern recognition overlap with KDD in the study of theories and algorithms for systems that extract patterns and models from data (mainly data mining methods). It focuses on the extension of these theories and algorithms to the problem of finding special patterns (ones that may be interpreted as useful or interesting knowledge) in large sets of real-world data.

KDD also has much in common with statistics, particularly exploratory data analysis (EDA). KDD systems often embed particular statistical procedures for modeling data and handling noise within an overall knowledge discovery framework.

Another related area is data warehousing, which refers to the recently popular MIS trend for collecting and cleaning transactional data and making them available for on-line retrieval. A popular approach for analysis of data warehouses has been called OLAP (on-line analytical processing). OLAP tools focus on providing multi-dimensional data analysis, which is superior to SQL (standard query language) in computing summaries and breakdowns along many dimensions. We view both knowledge discovery and OLAP as related facets of a new generation of intelligent information extraction and management tools.

## 1.6 Data Warehouse

A data warehouse represents a large collection of data which in principal can provide views of the data that are not practical for individual transaction sources. For example, a supermarket chain may want to compare sales trends across regions at the level of the product, broken down by weeks, and buy class id store within a region. Such views are often recomputed and stored in special purpose data stores that provide a multi dimensional front end to the underlying relational database and are sometimes called multidimensional databases.

## 1.7 Online Transaction Processing

Online transaction processing systems (OLTP) (or *operational systems*) cover the activities, systems, and processes associated with entering data reliably into a database. An online transaction processing system maintains the data required to operate the business on a day-to-day basis. As such, the data is point-in-time and applications are specific to business functions such as Accounts Receivable, Accounts Payable, Order Entry, and Payroll.

There are two key issues with online transaction processing systems:

➤ *Good performance*

Response times of under a second have been achieved over the years with much fine tuning of the hardware and software

➤ *High availability*

This has been achieved with overall improvements in the stability of hardware.

## 1.8 Data Mining

Data Mining is the process of *extracting knowledge hidden from large volumes of raw data*. The importance of collecting data that reflect your business or scientific activities to achieve competitive advantage is widely recognized now. Powerful systems for collecting data and managing it in large databases are in place in all large and mid-range companies. However, the bottleneck of turning this data into your success is the difficulty of extracting knowledge about the system you study from the collected data.

Human analysts with no special tools can no longer make sense of enormous volumes of data that require processing in order to make informed business decisions. Data mining automates the process of finding relationships and patterns in raw data and delivers results that can be either utilized in an automated decision support system or assessed by a human analyst.

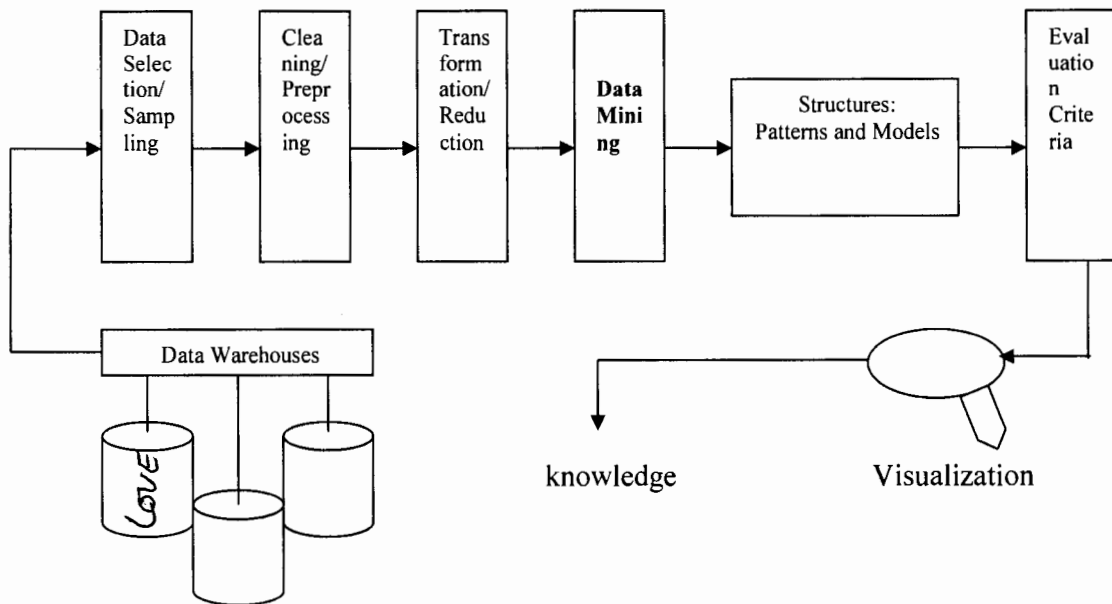
- What goods should be promoted to this customer?
- What is the probability that a certain customer will respond to a planned promotion?
- Can one predict the most profitable securities to buy/sell during the next trading session?
- Will this customer default on a loan or pay back on schedule?
- What medical diagnose should be assigned to this patient?
- How large the peak loads of a telephone or energy network are going to be?
- Why the facility suddenly starts to produce defective goods?

These are all the questions that can probably be answered if information hidden among megabytes of data in your database can be found explicitly and utilized. Modeling

the investigated system, discovering relations that connect variables in a database are the subject of data mining.

Modern computer data mining systems self learn from the previous history of the investigated system, formulating and testing hypotheses about the rules which this system obeys. When concise and valuable knowledge about the system of interest had been discovered, it can and should be incorporated into some decision support system which helps the manager to make wise and informed business decisions.

Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases



**Fig 1.3 Data Mining Process**

## 1.9 Why data mining?

Data might be one of the most valuable assets of a corporation - but only if we know how to reveal valuable knowledge hidden in raw data. Data mining allows us to extract diamonds of knowledge from historical data and predict outcomes of future situations. It will help us optimize business decisions, increase the value of each customer and communication, and improve satisfaction of customer with better services.

Data required for analysis may be of different types. Examples include:

- Sales and contacts histories
- Call support data

- Demographic data on your customers and prospects
- Patient diagnoses and prescribed drugs data
- Click stream and transactional data from your website

In all these cases data mining can help you reveal knowledge hidden in data and turn this knowledge into a crucial competitive advantage. Today increasingly more companies acknowledge the value of this new opportunity

## 1.10 Uses of Data Mining

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. It enables to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

For example, Blockbuster Entertainment mines its video rental history database to recommend rentals to individual customers. American Express can suggest products to its cardholders based on analysis of their monthly expenditures.

WalMart is pioneering massive data mining to transform its supplier relationships. WalMart captures point-of-sale transactions from over 2,900 stores in 6 countries and continuously transmits this data to its massive 7.5 terabyte Teradata data warehouse. WalMart allows more than 3,500 suppliers, to access data on their products and perform data analyses. These suppliers use this data to identify customer buying patterns at the store display level. They use this information to manage local store inventory and identify new merchandising opportunities. In 1995, WalMart computers processed over 1 million complex data queries.

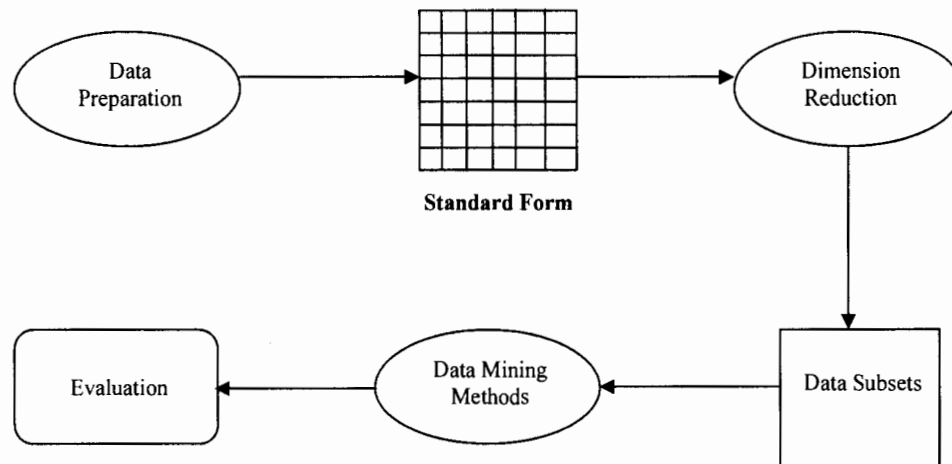
Data Mining can perform the following:

- **Identify best prospects and then retain them as customers.**  
By concentrating the marketing efforts only on best prospects we will save time and money, thus increasing effectiveness of marketing operation.
- **Predict cross-sell opportunities and make recommendations.**  
Whether we have a traditional or web-based operation, it can help customers quickly locate products of interest to them - and simultaneously increase the value of each communication with customers.
- **Learn parameters influencing trends in sales and margins.**  
One thinks one can do this with OLAP tools? True, OLAP can help us prove a hypothesis - but only if we know what questions to ask in the first place. In the

majority of cases we have no clue on what combination of parameters influences operation. In these situations data mining is only real option.

➤ **Segment markets and personalize communications.**

There might be distinct groups of customers, patients, or natural phenomena that require different approaches in their handling. If we have a broad customer range, would need to address teenagers in California and married homeowners in Minnesota with different products and messages in order to optimize your marketing campaign.



**Fig 1.4 Data Reduction**

## 1.11 Reasons for the growing popularity of Data Mining

Data mining derives its name from the similarities between searching for valuable business information in a large database for example, finding linked products in gigabytes of store scanner data and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. This can easily be done through data mining. There are many other reasons also for growing popularity of data mining such as:

### 1.11.1 Growing Data Volume

The main reason for necessity of automated computer systems for intelligent data analysis is the enormous volume of existing and newly appearing data that require processing. The amount of data accumulated each day by various business, scientific, and governmental organizations around the world is daunting. According to information from GTE research center, only scientific organizations store each day about 1 TB (terabyte!) of new information, and it is well known that academic world is by far not the leading supplier of new data. It becomes impossible for human analysts to cope with such overwhelming amounts of data.



### 1.11.2 Limitations of Human Analysis

Other problems that surface when human analysts process data are the inadequacy of the human brain when searching for complex multifactor dependencies in data, and the lack of objectiveness in such an analysis. A human expert is always a hostage of the previous experience of investigating other systems. Sometimes this helps, sometimes this hurts, but it is almost impossible to get rid of this fact.

### 1.11.3 Low Cost of Machine Learning

One additional benefit of using automated data mining systems is that this process has a much lower cost than hiring an army of highly trained (and payed) professional statisticians. While data mining does not eliminate human participation in solving the task completely, it significantly simplifies the job and allows an analyst who is not a professional in statistics and programming to manage the process of extracting knowledge from data.

## 1.12 How does data mining work?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

- **Classes:** Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters:** Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations:** Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns:** Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of five major elements:

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table.

### 1.13 Applications of Data Mining

A data mining application is an implementation of data mining technology that solves a specific business or research problem. Example application areas include:

- A pharmaceutical company can analyze its recent sales force activity and their results to improve targeting of high-value physicians and determine which marketing activities will have the greatest impact in the next few months. The data needs to include competitor market activity as well as information about the local health care systems. The results can be distributed to the sales force through a wide-area network that enables the representatives to review the recommendations from the perspective of the key attributes in the decision process. The ongoing, dynamic analysis of the data warehouse allows best practices from throughout the organization to be applied in specific sales situations.
- A credit card company can leverage its vast warehouse of customer transaction data to identify customers most likely to be interested in a new credit product. Using a small test mailing, the attributes of customers with an affinity for the product can be identified. Recent projects have indicated more than a 20-fold decrease in costs for targeted mailing campaigns over conventional approaches.
- A diversified transportation company with a large direct sales force can apply data mining to identify the best prospects for its services. Using data mining to analyze its own customer experience, this company can build a unique segmentation identifying the attributes of high-value prospects. Applying this segmentation to a general business database such as those provided by Dun & Bradstreet can yield a prioritized list of prospects by region.
- A large consumer package goods company can apply data mining to improve its sales process to retailers. Data from consumer panels, shipments, and competitor activity can be applied to understand the reasons for brand and store switching. Through this analysis, the manufacturer can select promotional strategies that best reach their target customer segments.

***Chapter 2***

***Literature Survey***

## 2. LITERATURE REVIEW

Data Mining is used to discover patterns and relationships in data, with an emphasis on large observational data bases. It sits at the common frontiers of several fields including Data Base Management, Artificial Intelligence, Machine Learning, Pattern Recognition, and Data Visualization. A statistical perspective is this that it can be viewed as computer automated exploratory data analysis of large complex data sets. In spite of the somewhat exaggerated hype, this field is having a major impact in business, industry and science. It also affords enormous research opportunities for new methodological developments. Despite the obvious connections between data mining and statistical data analysis<sup>[21]</sup>, most of the methodologies used in Data Mining have so far originated in fields other than Statistics. It is argued that Statistics can potentially have a major influence on Data Mining, but in order to do so some of our basic paradigms and operating principles may have to be modified.

Evolutionary Step	Business Question	Enabling Technologies	Product Providers	Characteristics
Data Collection (1960s)	"What was my total revenue in the last five years?"	Computers, tapes, disks	IBM, CDC	Retrospective, static data delivery
Data Access (1980s)	"What were unit sales in New England last March?"	Relational databases (RDBMS), Structured Query Language (SQL), ODBC	Oracle, Sybase, Informix, IBM, Microsoft	Retrospective, dynamic data delivery at record level
Data Warehousing & Decision Support (1990s)	"What were unit sales in New England last March? Drill down to Boston."	On-line analytic processing (OLAP), multidimensional databases, data warehouses	Pilot, Comshare, Arbor, Cognos, Microstrategy	Retrospective, dynamic data delivery at multiple levels
Data Mining (Emerging Today)	"What's likely to happen to Boston unit sales next month? Why?"	Advanced algorithms, multiprocessor computers, massive databases	Pilot, Lockheed, IBM, SGI, numerous startups (nascent industry)	Prospective, proactive information delivery

**Fig 2.1 Steps in the Evolution of Data Mining.**

## 2.1 Emergence of Data Mining

The idea of learning from data has been around for a long time. So it is reasonable to ask why the interest in data mining has suddenly become so intense. The principal reason is that the field of Data Base Management has recently become involved. Data, especially large amounts of it, reside in data base management systems (DBMS). Conventional DBMS are focused on On-Line Transaction Processing (OLTP); that is, the storage and fast retrieval of individual records for purposes of data organization. They are used to keep track of inventory, payroll records, billing records, invoices, etc.

Recently the Data Base Management community has become interested in using DBMS for Decision Support. Such Decision Support systems (DSS) allow statistical queries from data collected for OLTP applications. A DSS requires the construction of a Data Warehouse. Data Warehouses unify the data scattered throughout the many departments of an organization into a single centralized (usually very large 100 GB) data base with a common format. Sometimes smaller sub data bases are also constructed for specialized analyses; these are called Data Marts. Decision Support systems are intended for on-line analytic processing (OLAP) and relational OLAP, called ROLAP. ROLAP is intended for multidimensional analysis. ROLAP data bases are organized by dimension, which is logical grouping by attributes). The conceptual framework is that of a data cube which can be viewed as a large high dimensional contingency table.

## 2.2 Clustering at Glance

General references regarding clustering include Hartigan 1975<sup>[29]</sup> Spath 1980<sup>[13]</sup> Jain & Dubes 1988<sup>[20]</sup> Kaufman & Rousseeuw 1990<sup>[25]</sup> Dubes & Everitt 1993<sup>[26]</sup> Mirkin 1996<sup>[27]</sup> Jain et al. 1999<sup>[28]</sup> Fasulo 1999<sup>[30]</sup> Kolatch 2001<sup>[31]</sup> Han et al. 2001<sup>[34]</sup> Ghosh 2002<sup>[38]</sup>. A very good introduction to contemporary data mining clustering techniques can be found in the textbook<sup>[39]</sup>. There is a close relationship between clustering techniques and many other disciplines. Clustering has always been used in statistics and science. The classic introduction into pattern recognition framework is given in<sup>[20]</sup>. Machine learning clustering algorithms were applied to *image segmentation* and *computer vision*. For statistical approaches to pattern recognition see<sup>[21]</sup>. Clustering can be viewed as a density estimation problem. This is the subject of traditional multivariate statistical estimation. Clustering is also widely used for data compression in image processing, which is also known as *vector quantization*. Data fitting in numerical analysis provides still another venue in data modeling<sup>[38]</sup>.

This survey's emphasis is on clustering in data mining. Such clustering is characterized by large datasets with many attributes of different types. Though we do not even try to review particular applications, many important ideas are related to the specific fields. Clustering in data mining was brought to life by intense developments in information retrieval and text mining, spatial database applications.

## 2.3 Partitioning Clustering

In this section we survey data partitioning algorithms, which divide data into several subsets. Because checking all possible subset systems is computationally infeasible, certain greedy heuristics are used in the form of *iterative optimization*. Specifically, this means different *relocation* schemes that iteratively reassign points between the  $k$  clusters. Unlike traditional hierarchical methods, in which clusters are not

revisited after being constructed, relocation algorithms gradually improve clusters. With appropriate data, this results in high quality clusters.

One approach to data partitioning is to take a *conceptual* point of view that identifies the cluster with a certain model whose unknown parameters have to be found. More specifically, *probabilistic* models assume that the data comes from a mixture of several populations whose distributions and priors we want to find. Corresponding algorithms are described in the sub-section *Probabilistic Clustering*. One clear advantage of probabilistic methods is the interpretability of the constructed clusters. Having concise cluster representation also allows inexpensive computation of intra-clusters measures of fit that give rise to a global *objective function*.

Another approach starts with the definition of *objective function* depending on a partition. As we have seen (sub-section *Linkage Metrics*), pair-wise distances or similarities can be used to compute measures of inter- and intra-cluster relations. In iterative improvements such pair-wise computations would be too expensive. Using unique cluster representatives resolves the problem: now computation of objective function becomes linear in  $N$  (and in a number of clusters). Depending on how representatives are constructed, iterative optimization partitioning algorithms are subdivided into *k-medoids* and *k-means* methods. *K-medoid* is the most appropriate data point within a cluster that represents it. Representation by *k-medoids* has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. In *k-means* case a cluster is represented by its centroid, which is a mean (usually weighted average) of points within a cluster. This works conveniently only with numerical attributes and can be negatively affected by a single outlier. On the other hand, centroids have the advantage of clear geometric and statistical meaning.

### 2.3.1 K-Medoids Methods

Two early versions of *k-medoid* methods are the algorithm PAM (Partitioning Around Medoids) and the algorithm CLARA (Clustering LARge Applications)<sup>[25]</sup>. PAM is iterative optimization that combines relocation of points between perspective clusters with re-nominating the points as potential medoids.

Further progress is associated with Ng & Han [1994] who introduced the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) in the context of clustering in *spatial* databases. Authors considered a graph whose nodes are the sets of *k* medoids and an edge connects two nodes if they differ by exactly one medoid. Our area of interest is *K-Means* that is described in next section.

### 2.3.2 K-Means Methods

The *k-means* algorithm<sup>[29]</sup> is by far the most popular clustering tool used in scientific and industrial applications. J. MacQueen in 1967 in his paper “*Some methods for classification and analysis of multivariate observations*” gave the basic *K-Means*. The name comes from representing each of *k* clusters  $C$  by the mean (or weighted average)  $c$  of its points, the so-called *centroid*. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. Two versions of *k-means* iterative optimization are known. The first version is

similar to EM algorithm and consists of two-step major iterations that (1) reassign all the points to their nearest centroids, and (2) recomputed centroids of newly assembled groups. Iterations continue until a stopping criterion is achieved (for example, no reassignments happen). This version is known as Forgy's algorithm<sup>[41]</sup>. The second (classic in iterative optimization) version of  $k$ -means iterative optimization reassigns points based on more detailed analysis of effects on the objective function caused by moving a point from its current cluster to a potentially new one. If a move has a positive effect, the point is relocated and the two centroids are recomputed. There is experimental evidence that compared with Forgy's algorithm, the second (classic) version frequently yields better results<sup>[23]</sup>. In particular, Dhillon et al. [2002] noticed that a Forgy's *spherical*  $k$ -means has a tendency to get stuck when applied to document collections. Pelleg & Moore<sup>[28]</sup> suggested how to directly (without any squashing) accelerate  $k$ -means iterative process by utilizing KD-trees [Moore 1999]. The algorithm  $X$ -means goes a step further: in addition to accelerating the iterative process it tries to incorporate a search for the best  $k$  in the process itself. The complexities are defined in the coming chapters.

The tremendous popularity of  $k$ -means algorithm has brought to life many other extensions and modifications. Mahalanobis distance can be used to cover hyperellipsoidal clusters. Maximum of intra-cluster variances, instead of the sum, can serve as an objective function. Generalizations that incorporate categorical attributes are known. Sometimes the term *k-prototypes* is used in this context. Modifications which constructs clusters of balanced size are discussed in the sub-section *Constrained-Based Clustering*.

### 2.3.3 Clustering and the Continuous $k$ -Means Algorithm

In 1994 Vance Faber worked on several widely used algorithms that consolidate data by clustering, or grouping, and then present a new method, the continuous  $k$ -means algorithm,\* developed at the Laboratory specifically for clustering large datasets.

### 2.3.4 Efficient Classification Algorithm for Multispectral Satellite Images

In 1995 SUNG-HEE PARK \*HWANG-SOO KIM SOO-JUN PARK MYUNG-GIL JANG proposed a clustering algorithm to partition a multispectral remotely sensed image data set into several clusters using a hash search algorithm. The processing time of our algorithm is compared with that of clustering algorithm using other speed-up concepts

### 2.3.5 Spatial Data Mining Method

In 1996 Eun-Jeong Son in-Soo Kang Tae-Wan Kim Ki-Joune Li of Pusan National University proposed a spatial data mining method to analyze the influence between clusters and spatial objects and consequently, to discover relevance among influences.

### 2.3.6 Clustering Categorical Data: An Approach Based on Dynamical Systems

In 1997 David Gibson, Jon Kleinberg and Prabhakar Raghavan gave a novel approach for clustering collections of sets, and its application to the analysis and mining of categorical data

### 2.3.7 Refining Initial Points for K-Means Clustering

In June 1998 P. S. Bradley and Usama M. Fayyad of Microsoft presented a procedure for computing a refined starting condition from a given initial one that is based on an efficient technique for estimating the modes of a distribution. The refined initial starting condition allows the iterative algorithm to converge to a “better” local minimum.

### 2.3.8 K-Modes Methods

The  $k$ -means algorithm is best suited for implementing this operation because of its efficiency in clustering large data sets. However, working only on numeric values limits its use in data mining because data sets in data mining often contain categorical values. In 1998 Zhexue Huang presented an algorithm, called  $k$ -modes, to extend the  $k$ -means paradigm to categorical domains

### 2.3.9 K-Harmonic Means A Data Clustering Algorithm

In 1999 Bin Zhang, Meichun Hsu, Umeshwar Dayal of Hewlett-Packard Research Laboratory proposed a new clustering method called the  $K$ -Harmonic Means algorithm ( $KHM$ ).  $KHM$  is a center-based clustering algorithm which uses the Harmonic Averages of the distances from each data point to the centers as components to its performance function

### 2.3.10 Information management and Process Improvement Using Data Mining Techniques

In 2000 Gibbons, Ranta, Scott and Mantyla described a computer component manufacturing scenario which concentrates on the application of data mining techniques to improve information management and process improvement within a manufacturing scenario

## 2.4 Density Based

DBSCAN relies on  $R^*$ -tree indexation given in 1990 by Kriegel et al. Two generalizations lead to the algorithm GDBSCAN Sander et al. in 1998, which uses the same two parameters as algorithm DBSCAN. Algorithm OPTICS (Ordering Points To Identify the Clustering Structure) given by Ankerst et al. in 1999 adjusts DBSCAN to this challenge. It builds an augmented ordering of data which is consistent with DBSCAN, but goes a step further:



## 2.5 Grid Based

The algorithm STING (STatistical INformation Grid-based method) was given by Wang et al. in 1997.

### 2.5.1 BIRCH: A New Data Clustering Algorithm and Its Applications

In 1997 Tian Zhang, Raghu Ramakrishnan, Miron Livny gave an efficient and scalable data clustering method, based on a new in-memory data structure called CF-tree, which serves as an in-memory summary of the data distribution. We have implemented it in a system called BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), and studied its performance extensively in terms of memory requirements, running time, clustering quality, stability and scalability.

## 2.6 APRIORI algorithm

In 1996 Agrawal, Mannila, Srikant, Toivonen, Verkamo gave an algorithm based on *Fast discovery of Association Rules*

## 2.7 An improved Bayesian Structural EM algorithm

J M Pefna, J A Lozano and P Larranaga proposed to perform the optimization of the Bayesian network parameters using an alternative approach to the EM algorithm the BC\_EM method.

## *Chapter 3*

# *Data Mining Techniques*

### 3. DATA MINING TECHNIQUES

There have been many advances on researches and developments of data mining, and many data mining techniques and systems have recently been developed. Different classification schemes can be used to categorize data mining methods and systems based on the kinds of databases to be studied, the kinds of knowledge to be discovered, and the kinds of techniques to be utilized,

The data mining categories can be classified into these categories:

- Association Rule
- Sequence Analysis
- Clustering
- Classification

#### 3.1 Association Rule

Association analysis is an unsupervised form of data mining that looks for links between records in a data set. Association analysis is sometimes referred to as 'market basket analyses, its most common application. [28]

##### 3.1.1 The basic process of mining association rule:

The basic process for association rule analysis consist of three important concerns

- Choosing the right set of items.
- Generating rules by deciphering the counts in the co-occurrence matrix.
- Overcoming the practical limits imposed by thousands or tens of thousands of items appearing in combinations large enough to be interesting.

##### 3.1.2 When is association rule analysis useful?

An appeal of market analysis comes from the clarity and utility of its results, which are in the form of *association rules*. There is an intuitive appeal to a market analysis because it expresses how tangible products and services relate to each other, how they tend to group together? A rule like, "if a customer purchases three way calling, then that customer will also purchase call waiting" is clear. Even better, it suggests a specific course of action, like bundling three-way calling with call waiting into a single service package. While association rules are easy to understand, they are not always useful. The following three rules are examples of real rules generated from real data:

- On Thursdays, grocery store consumers often purchase diapers and beer together.
- Customers who purchase maintenance agreements are very likely to purchase large appliances.

- When a new hardware store opens, one of the most commonly sold items is toilet rings.

These three examples illustrate the three common types of rules produced by association rule analysis: the *useful*, the *trivial*, and the *inexplicable*.

The *useful* rule contains high quality, actionable information. In fact, once the pattern is found, it is often not hard to justify. The rule about diapers and beer on Thursdays suggests that on Thursday evenings, young couples prepare for the weekend by stocking up on diapers for the infants and beer for dad (who, for the sake of argument, we stereotypically assume is watching football on Sunday with a six-pack). By locating their own brand of diapers near the aisle containing the beer, they can increase sales of a high-margin product. Because the rule is easily understood, it suggests plausible causes, leading to other interventions: placing other baby products within sight of the beer so customers do not “forget” anything and putting other leisure foods, like potato chips and pretzels, near the baby products.

*Trivial* results are already known by anyone at all familiar with the business. The second example “Customers who purchase maintenance agreements are very likely to purchase large appliances” is an example of a trivial rule. In fact, we already know that customers purchase maintenance agreements and large appliances at the same time. Why else would they purchase maintenance agreements? The maintenance agreements are advertised with large appliances and rarely sold separately. This rule, though, was based on analyzing hundreds of thousands of point-of-sale transactions from Sears. Although it is valid and well-supported in the data, it is still useless. Similar results abound: People who buy 2-by-4s also purchase nails; customers who purchase paint buy paint brushes; oil and oil filters are purchased together as are hamburgers and hamburger buns, and charcoal and lighter fluid.

A subtler problem falls into the same category. A seemingly interesting result—like the fact that people who buy the three-way calling option on their local telephone service almost always buy call waiting—may be the result of marketing programs and product bundles. In the case of telephone service options, three-way calling is typically bundled with call waiting, so it is difficult to order it separately. In this case, the analysis is not producing actionable results; it is producing already acted-upon results. Although a danger for any data mining technique, association rule analysis is particularly susceptible to reproducing the success of previous marketing campaigns because of its dependence on un-summarized point-of-sale data—exactly the same data that defines the success of the campaign. Results from association rule analysis may simply be measuring the success of previous marketing campaigns.

*Inexplicable* results seem to have no explanation and do not suggest a course of action. The third pattern (“When a new hardware store opens, one of the most commonly sold items is toilet rings”) is intriguing, tempting us with a new fact but providing information that does not give insight into consumer behavior or the merchandise, or suggest further actions. In this case, a large hardware company discovered the pattern for new store openings, but did not figure out how to profit from it. Many items are on sale during the store openings, but the toilet rings stand out. More investigation might give some explanation: Is the discount on toilet rings much larger than for other products? Are they consistently placed in a high-traffic area for store openings but hidden at other times? Is the result an anomaly from a handful of stores? Are they difficult to find at other

times? Whatever the cause, it is doubtful that further analysis of just the association rule data can give a credible explanation.

### 3.1.3 Working of Association Rule

This process is usually applied to transactional databases. These are databases where each record represents a transaction, typically some kind of sale. Formally, a transaction is defined as follows: given a set of items  $I$ , each transaction is a subset of the items. For example the items could be all the possible products of a store, and the transaction indicates which of these products were bought in a certain occasion by a particular customer. An association rule is defined as a relation:  $A \Rightarrow B$ , where  $A$  and  $B$  are both subset of the items. If the rule is a valid one, it indicates that all the records of the database which contain the items of  $A$ , also contain the items of  $B$ . In our example, this could be translated as: “all the customers who bought items #10 and #14 also bought items #19 and #75”.

The purpose of mining association rules is to discover all the possible rules which satisfy some condition of interest defined by the user. To formalize these conditions, we introduce two new values, the support of a rule and the confidence. The support is defined as the probability that the items of  $A$  and  $B$  are all present in a record, which is the number of records with  $A$  and  $B$ , divided the total number of records of the database.

$$\text{Support } (A \Rightarrow B) = \text{probability } (A \cup B) = |\{\text{records with } A \text{ and } B\}| / |\{\text{all records}\}|$$

The confidence is the probability that if a record contains the items of  $A$ , that same record contains also the items of  $B$ .

$$\text{Confidence } (A \Rightarrow B) = \text{probability } (B|A) = |\{\text{records with } A \text{ and } B\}| / |\{\text{records with } A\}|$$

For both these values we define a lower threshold,  $\text{min\_sup}$  and  $\text{min\_conf}$ . Every rule that has support and confidence superior to  $\text{min\_sup}$  and  $\text{min\_conf}$  is said to be a “strong” rule.

Our store example, a typical strong rule could be: “the customer who buys milk usually also buys biscuits”. More complex, and obscure, relationships could be discovered, giving us some insight on the expected behavior of our customer. This knowledge could be used to create special offers or to display the products in the right places. Usually, when applied to sales, the mining of association rules is called also Basket Market Analysis, but this is not the only possible application of the technique. Other fields of application include medicine: if we have a medical database large enough, we could find new relationships within symptoms and diseases, helping the diagnosis process.

### 3.1.4 Algorithms Used in Association Rules

The algorithms to discover all the strong rules of a database, given  $\text{min\_sup}$  and  $\text{min\_conf}$  as input, start looking for rules which have only one item in the left side, and then increasing the subset until the thresholds are broken. Other examples include.

### 3.1.4.1 Apriori Algorithm

An association rule mining algorithm, **Apriori** has been developed for rule mining in large transaction databases by IBM's Quest project team <sup>[24]</sup>. A *itemset* is a non-empty set of items.

They have decomposed the problem of mining association rules into two parts

- Find all combinations of items that have transaction support above minimum support. Call those combinations frequent itemsets.
- Use the frequent itemsets to generate the desired rules. The general idea is that if, say, ABCD and AB are frequent itemsets, then we can determine if the rule AB CD holds by computing the ratio  $r = \text{support}(ABCD)/\text{support}(AB)$ . The rule holds only if  $r \geq$  minimum confidence. Note that the rule will have minimum support because ABCD is frequent. The Apriori algorithm used in Quest for finding all frequent itemsets is given below

**procedure** AprioriAlg()

**begin**

```

L1 := {frequent 1-itemsets};
for ( k := 2; Lk-1 ≠ ∅; k++) do {
    Ck = apriori-gen(Lk-1); // new candidates
    for all transactions t in the dataset do {
        for all candidates c ∈ Ck contained in t do
            c.count++
    }
    Lk = { c ∈ Ck | c.count ≥ min-support }
}
Answer := ⋃k Lk

```

**end**

It makes multiple passes over the database. In the first pass, the algorithm simply counts item occurrences to determine the frequent 1-itemsets (itemsets with 1 item). A subsequent pass, say pass  $k$ , consists of two phases. First, the frequent itemsets  $L_{k-1}$  (the set of all frequent  $(k-1)$ -itemsets) found in the  $(k-1)$ th pass are used to generate the candidate itemsets  $C_k$ , using the `apriori-gen()` function. This function first joins  $L_{k-1}$  with  $L_{k-1}$ , the joining condition being that the lexicographically ordered first  $k-2$  items are the same. Next, it deletes all those itemsets from the join result that have some  $(k-1)$ -subset that is not in  $L_{k-1}$  yielding  $C_k$ .

The algorithm now scans the database. For each transaction, it determines which of the candidates in  $C_k$  are contained in the transaction using a hash-tree data structure and increments the count of those candidates. At the end of the pass,  $C_k$  is examined to determine which of the candidates are frequent, yielding  $L_k$ . The algorithm terminates when  $L_k$  becomes empty.

### 3.1.4.2 Distributed/Parallel Algorithms

Databases or data warehouses may store a huge amount of data to be mined. Mining association rules in such databases may require substantial processing power. A possible solution to this problem can be a distributed system <sup>[29]</sup>. Moreover, many large

databases are distributed in nature which may make it more feasible to use distributed algorithms.

Major cost of mining association rules is the computation of the set of large itemsets in the database. Distributed computing of large itemsets encounters some new problems. One may compute locally large itemsets easily, but a locally large itemset may not be globally large. Since it is very expensive to broadcast the whole data set to other sites, one option is to broadcast all the counts of all the itemsets, no matter locally large or small, to other sites. However, a database may contain enormous combinations of itemsets, and it will involve passing a huge number of messages.

A distributed data mining algorithm FDM (Fast Distributed Mining of association rules) has been proposed by [5], which has the following distinct features.

- The generation of candidate sets is in the same spirit of Apriori. However, some relationships between locally large sets and globally large ones are explored to generate a smaller set of candidate sets at each iteration and thus reduce the number of messages to be passed.
- After the candidate sets have been generated, two pruning techniques, local pruning and global pruning, are developed to prune away some candidate sets at each individual sites.
- In order to determine whether a candidate set is large, this algorithm requires only  $O(n)$  messages for support count exchange, where  $n$  is the number of sites in the network. This is much less than a straight adaptation of Apriori, which requires  $O(n^2)$  messages.

### 3.2 Sequence Analysis

The input data is a set of sequences, called data-sequences. Each data sequence is a ordered list of transactions(or itemsets), where each transaction is a sets of items (literals). Typically there is a transaction-time associated with each transaction. A sequential pattern also consists of a list of sets of items. The problem is to find all sequential patterns with a user-specified minimum support, where the support of a sequential pattern is the percentage of data sequences that contain the pattern <sup>[5]</sup>.

An example of such a pattern is that customers typically rent "Star Wars", then "Empire Strikes Back", and then "Return of the Jedi". Note that these rentals need not be consecutive. Customers who rent some other videos in between also support this sequential pattern. Elements of a sequential pattern need not be simple items. "Fitted Sheet and flat sheet and pillow cases", followed by "comforter", followed by "drapes and ruffles" is an example of a sequential pattern in which the elements are sets of items. This problem was initially motivated by applications in the retailing industry, including attached mailing, add-on sales, and customer satisfaction. But the results apply to many scientific and business domains. For instance, in the medical domain, a data-sequence may correspond to the symptoms or diseases of a patient, with a transaction corresponding to the symptoms exhibited or diseases diagnosed during a visit to the doctor. The patterns discovered using this data could be used in disease research to help identify symptoms/diseases that precede certain diseases

### 3.2.1 Working of Sequence Analysis

We consider our database formed by a number of sequence  $s$ . Each sequence is an ordinate list of itemset  $s = \langle s_1, s_2, \dots, s_n \rangle$ . This could represent all the operations required by a single customer. Given the set of all possible items  $I$ , an itemset  $s_i$  is a subset of  $I$ . In the bank example the itemset represents all the operations made by the customer simultaneously. The purpose of mining sequential patterns is to discover sequences of itemset frequent enough. Again, the exact value of “enough” is defined as an input parameter by the user. We define the *support* of a certain sequence  $s$ , as the fraction of the sequences of the database which contains  $s$ . Finding sequential patterns means to discover all the sequences  $s$  which have a support superior to a certain threshold.

To give an exact definition of the problem it's necessary to introduce some temporal parameters. These are:

- *Sliding time window*: when we are looking for a certain itemset  $s_i$ , we allow its elements to be spread over different transactions, all included in the *sliding time window*.
- *Min\_gap*: this indicates the minimum time required for two transactions to be considered sequential. If their time indicators differ from a value inferior to *Min\_gap*, the two transactions are considered contemporaneous.
- *Max\_gap*: this indicates the maximum time allowed for different transactions to be considered connected and belonging to the same sequence. If two transactions have their time indicators that differ from a value superior to *Max\_gap*, it means that the two transactions have no relationship, and must be considered in two separate sequences.

One possible reason for a bank to look for sequential patterns in its database could be to anticipate the moves of its customers. For example, it could be discovered that unhappy customers, before closing their account, have similar behaviors. This could allow the bank to understand when a customer is not satisfied with the service and to take precautions not to lose him or her. Other examples of application are again in the field of medicine. Mining sequential patterns could help to connect the treatments with the course of the disease in a certain patient.

### 3.2.2 Algorithms Used in Sequence Analysis

Various groups working in this field have suggested algorithms for mining sequential patterns. Listed below are two algorithms proposed by IBM's Quest data team.

*Terminology*: The length of a sequence is the number of itemsets in the sequence. A sequence of length  $k$  is called a  $k$ -sequence. The sequence formed by the concatenation of two sequences  $x$  and  $y$  is denoted as  $x.y$ . The support for an itemset  $i$  is defined as the fraction of customers who bought the items in  $i$  in a single transaction. Thus the itemset  $i$  and the 1-sequence  $i$  have the same support. An itemset with minimum support is called a large itemset or *litemset*. Note that each itemset in a large sequence must have minimum support. Hence, any large sequence must be a list of litemsets. In the algorithms,  $L_k$  denotes the set of all large  $k$ -sequences, and  $C_k$  the set of candidate  $k$ -sequences.



There are two families of algorithms- *count-all* and *count-some*. The count-all algorithms count all the large sequences, including non-maximal sequences. The non-maximal sequences must then be pruned out (in the maximal phase). **AprioriAll** listed below is a count-all algorithm, based on the Apriori algorithm for finding large itemsets presented in chapter2. **Apriori-Some** is a count-some algorithm. The intuition behind these algorithms is that since we are only interested in maximal sequences, we can avoid counting sequences which are contained in a longer sequence if we first count longer sequences. However, we have to be careful not to count a lot of longer sequences that do not have minimum support. Otherwise, the time saved by not counting sequences contained in a longer sequence may be less than the time wasted counting sequences without minimum support that would never have been counted because their subsequences were not large.

### 3.2.2.1 AprioriAll Algorithm

The algorithm is given below <sup>[8]</sup>. In each pass, we use the large sequences from the previous pass to generate the candidate sequences and then measure their support by making a pass over the database. At the end of the pass, the support of the candidates is used to determine the large sequences. In the first pass, the output of the litemset phase is used to initialize the set of large 1-sequences. The candidates are stored in *hash-tree* to quickly find all candidates contained in a customer

```

L1 = large 1-sequences; // Result of litemset phase
for ( k = 2; Lk-1 0; k++) do
begin
    Ck = New Candidates generated from Lk-1 (see below)
    foreach customer-sequence c in the database do
        Increment the count of all candidates in Ck that are contained in c.
    Lk = Candidates in Ck with minimum support.
end

```

Answer = Maximal Sequences in  $\bigcup_k L_k$  ;

### 3.2.2.2 AprioriSome Algorithm

In this algorithm <sup>[8]</sup>, we only count sequences of certain lengths. For example, we might count sequences of length 1, 2, 4 and 6 in the forward phase and count sequences of length 3 and 5 in the backward phase. The function *next* takes as parameter the length of sequences counted in the last pass and returns the length of sequences to be counted in the next pass. Thus, this function determines exactly which sequences are counted, and balances the tradeoff between the time wasted in counting non-maximal sequences versus counting extensions of small candidate sequences. One extreme is  $\text{next}(k) = k + 1$  ( $k$  is the length for which candidates were counted last), when all non-maximal sequences are counted, but no extensions of small candidate sequences are counted. In this case, AprioriSome degenerates into AprioriAll. The other extreme is a function like  $\text{next}(k) = 100 * k$ , when almost no non-maximal large sequence is counted, but lots of extensions of small candidates are counted.

### 3.3 Classification

In Data classification one develops a description or model for each class in a database, based on the features present in a set of class-labeled training data. Classification identifies a specific group or class to which an item belongs. A prediction based on a classification model will, therefore, be a discrete outcome, identifying a customer as a responder or non-responder, or a patient as having a high or low risk of heart failure..

Classification is the operation most commonly supported by commercial data mining tools. It is an operation that enables organisations to discover patterns in large or complex data sets in order to solve specific business problems.

Classification is the process of sub-dividing a data set with regard to a number of specific outcomes. For example, we might want to classify our customers into 'high' and 'low' categories with regard to credit risk. The category or 'class' into which each customer is placed is the 'outcome' of our classification.

A crude method would be to classify customers by whether their income is above or below a certain amount. A slightly more subtle approach tries to find a linear relationship between two different factors - such as income and age -to divide a data set into two groups. Real-world classification problems usually involve many more dimensions and therefore require a much more complex delimitation between different classes.

#### 3.3.1 Classification Methods

There have been many data classification methods studied:

- **Statistical Algorithms** Statistical analysis systems such as SAS and SPSS have been used by analysts to detect unusual patterns and explain patterns using statistical models such as linear models. Such systems have their place and will continue to be used.
- **Neural Networks** Artificial neural networks mimic the pattern-finding capacity of the human brain and hence some researchers have suggested applying Neural Network algorithms to pattern-mapping. Neural networks have been applied successfully in a few applications that involve classification.
- **Genetic algorithms** Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Nearest neighbor** Nearest Neighbour (more precisely k-nearest neighbour, also k-NN) is a predictive technique suitable for classification models. Unlike other predictive algorithms, the training data is not scanned or processed to create the model. Instead, the training data is the model. When a new case or instance is presented to the model, the algorithm looks at all the data to find a subset of cases that are most similar to it and uses them to predict the outcome. As the term *nearest* implies, k-NN is based on a concept of distance, and this requires a metric to determine distances. All metrics must result in a specific number for comparison purposes. Whatever metric is used is both arbitrary and extremely

important. It is arbitrary because there is no preset definition of what constitutes a "good" metric. It is important because the choice of a metric greatly affects the predictions. Different metrics, used on the same training data, can result in completely different predictions. This means that a business expert is needed to help determine a good metric.

- **Rule induction** The extraction of useful *if-then* rules from data based on statistical significance.
- **Data visualization** The visual interpretation of complex relationships in multidimensional data.
- **Naïve-Bayes** Naïve-Bayes is a classification technique that is both predictive and descriptive. It analyses the relationship between each independent variable and the dependent variable to derive a conditional probability for each relationship. Naïve-Bayes requires only one pass through the training set to generate a classification model. This makes it the most efficient data mining technique. However, Naïve-Bayes does not handle continuous data, so any independent or dependent variables that contain continuous values must be binned or bracketed. For instance, if one of the independent variables is age, the values must be transformed from the specific value into ranges such as "less than 20 years," "21 to 30 years," "31 to 40 years" and so on.
- **Decision Trees** Decision trees are one of the most common data mining technique and are by far the most popular in tools aimed at the business user. They are easy to set up, their results are understandable by an end-user, they can address a wide range of classification problems, they are robust in the face of different data distributions and formats, and they are effective in analysing large numbers of fields. A decision tree algorithm works by splitting a data set in order to build a model that successfully classifies each record in terms of a target field or variable. An example is a decision tree which classifies a data set according to whether customers did or did not buy a particular product. The most common types of decision tree algorithm are CHAID, CART and C4.5. CHAID (Chi-square automatic interaction detection) and CART (Classification and Regression Trees) were developed by statisticians. CHAID can produce tree with multiple sub-nodes for each split. CART requires less data preparation than CHAID, but produces only two-way splits. C4.5 comes from the world of machine Learning, and is based on information theory. In order to generate a decision tree from the training set of data we need to split the data into progressively smaller subsets. Each iteration considers the data in only one node. The first iteration considers the root node that contains all the data. Subsequent iterations work on derivative nodes that will contain subsets of the data.

### 3.3.2 Classification Techniques

How the data mining tool analyses this data, and the type of information it provides, depends on the techniques it uses. The most common techniques for classification are decision trees and neural networks. If a decision tree is used, it will provide a set of branching conditions that successively split the customers into groups defined by the values in the independent variables. The aim is to be able to produce a set of rules or a model of some sort that can identify a high percentage of responders. A decision tree may formulate a condition such as:

*customers who are male and married and have incomes over £50,000 and who are home-owners responded to our offer*

The condition selects a much high percentage of responders than if you took a random selection of customers.

In contrast, a neural network identifies which class a customer belongs to, but cannot tell you why. The factors that determine its classifications are not available for analysis, but remain implicit in the network itself. Another set of techniques used for classification are k-nearest neighbour algorithms.

### 3.3.3 Understanding and prediction

Sophisticated classification techniques enable us to discover new patterns in large and complex data sets. Classification is therefore a powerful aid to understanding a particular problem, whether it is response rates to a direct mailing campaign or the influence of various factors on the likelihood of a patient recovering from cancer.

In some instances, improved understanding is sufficient. It may suggest new initiatives and provide information that improves future decision making. However, in many instances the reason for developing an accurate classification model is to improve our capability for prediction.

For example, we know that historically 60 per cent of customers who are male, married and have incomes over £60,000 responded to a promotion (compared with only three per cent of all targeted customers). There is therefore a better than average chance that new customers who fit this profile will also be interested in our product. In practice, data mining algorithms can find much more complex relationships involving numerous predictor variables, thus providing a much finer segmentation of customers.

A classification model is said to be ‘trained’ on historical data, for which the outcome is known for each record. It is then applied to a new, unclassified data set in order to predict the outcome for each record.

There are important differences between classifying data in order to understand the behaviour of existing customers and using that classification to predict future behaviour. For a historical data set, it is often possible to produce a set of rules or a mathematical function that classifies every record accurately. For example, if you keep refining your rules you end up with a rule for each individual of the form:

*100 per cent of customers called Smith who live at 28 Arcadia Street responded to our offer.*

Such a rule is of little use for predicting the classification of a new customer. In this case, the model is said to be ‘over-trained’ or ‘overfitted’ to the historical data set. Building a good predictive model requires that overfitting be avoided by testing and tuning a model to ensure that it can be ‘generalised’ to new data.

### 3.3.4 Algorithms used in Classification

There are many different algorithms used for the classification. Some of them are described in brief below.

#### 3.3.4.1 ID3 Algorithm

The ID3 algorithm (Quinlan86)<sup>[7]</sup> is a decision tree building algorithm which determines the classification of objects by testing the values of their properties. It builds the tree in a top down fashion, starting from a set of objects and a specification of properties. At each node of the tree, a property is tested and the results used to partition the object set. This process is recursively done till the set in a given subtree is homogeneous with respect to the classification criteria - in other words it contains objects belonging to the same category. This then becomes a leaf node. At each node, the property to test is chosen based on information theoretic criteria that seek to maximize information gain and minimize entropy. In simpler terms, that property is tested which divides the candidate set in the most homogeneous subsets.

#### 3.3.4.2 C4.5 Algorithm

This algorithm was proposed by Quinlan (1993)<sup>[9]</sup>. The C4.5 algorithm generates a classification-decision tree for the given data-set by recursive partitioning of data. The decision is grown using **Depth-first** strategy. The algorithm considers all the possible tests that can split the data set and selects a test that gives the best information gain. For each discrete attribute, one test with outcomes as many as the number of distinct values of the attribute is considered. For each continuous attribute, binary tests involving every distinct values of the attribute are considered. In order to gather the entropy gain of all these binary tests efficiently, the training data set belonging to the node in consideration is sorted for the values of the continuous attribute and the entropy gains of the binary cut based on each distinct values are calculated in one scan of the sorted data. This process is repeated for each continuous attributes.

#### 3.3.4.3 SLIQ Algorithm

SLIQ (Supervised Learning In Quest)<sup>[10]</sup> developed by IBM's Quest project team, is a decision tree classifier designed to classify large training data [1]. It uses a pre-sorting technique in the tree-growth phase. This helps avoid costly sorting at each node. SLIQ keeps a separate sorted list for each continuous attribute and a separate list called class list. An entry in the class list corresponds to a data item, and has a class label and name of the node it belongs in the decision tree. An entry in the sorted attribute list has an attribute value and the index of data item in the class list. SLIQ grows the decision tree in **breadth-first** manner. For each attribute, it scans the corresponding sorted list and calculate entropy values of each distinct values of all the nodes in the frontier of the decision tree simultaneously. After the entropy values have been calculated for each attribute, one attribute is chosen for a split for each nodes in the current frontier, and they are expanded to have a new frontier. Then one more scan of the sorted attribute list is performed to update the class list for the new nodes. While SLIQ handles disk-resident data that are too large to fit in memory, it still requires some information to stay memory-

resident which grows in direct proportion to the number of input records, putting a hard-limit on the size of training data. The Quest team has recently designed a new decision-tree-based classification algorithm, called SPRINT (Scalable PaRallelizable INduction of decision Trees) that for the removes all of the memory restrictions.

### 3.4 Clustering

Clustering is an unsupervised operation. It is used where you wish to find groupings of similar records in your data without any preconditions as to what that similarity may involve. Clustering is used to identify interesting groups in a customer base that may not have been recognised before. For example, it can be used to identify similarities in customers' telephone usage, in order to devise and market new call services.

Clustering is usually achieved using statistical methods, such as a k-means algorithm, or a special form of neural network called a Kohonen feature map network. Whichever method is used, the basic operation is the same. Each record is compared with a set of existing clusters, which are defined by their 'centre'. A record is assigned to the cluster it is nearest to, and this in turn changes the value that defines that cluster. Multiple passes are made through a data set to re-assign records and adjust the cluster centres until an optimum solution is found.

#### 3.4.Working of Cluster Analysis

Cluster analysis is the process of identifying the relationships that exist between items on the basis of their similarity and dissimilarity. Unlike classification, clustering does not require a target variable to be identified beforehand. A clustering algorithm takes an unbiased look at the potential groupings within a data set and attempts to derive an optimum delineation of items on the basis of those groups.

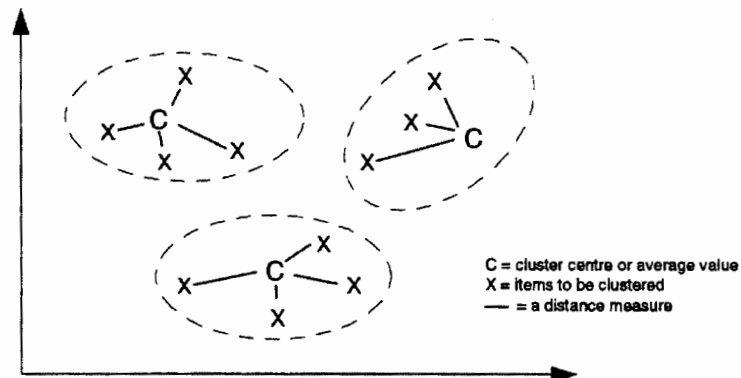


Fig 3.1 Clusters

Clusters are typically based around a "centre" or average value. How centres are initially defined and adjusted varies between algorithms. One method is to start with a random set of centers, which are then adjusted, added to and removed as the analysis progresses.

To identify items that belong to a cluster, some measure must be used that gauges the similarity between items within a cluster and their dissimilarity to items in other

clusters. The similarity and dissimilarity between items is typically measured as their distance from each other and from the cluster centres within a multi-dimensional space, where each dimension represents one of the variables being compared.

### 3.4.2 Uses of Cluster Analysis

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape (not only spherical clusters)
- Minimal requirements for domain knowledge to determine input parameters (such as # of clusters)
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality (especially very sparse and highly skewed data)
- Incorporation of user-specified constraints
- Interpretability and usability (close to semantics)

### 3.4.3 Further Classification of Cluster Techniques

Clusters can be further divided into four categories.

- **Partitioning Method:** Construct various partitions and then evaluate them by some criterion
- **Hierarchical Method:** Create a hierarchical decomposition (agglomerative or divisive) of the set of data (or objects) using some criterion
- **Density Based Method:** based on connectivity and density functions
- **Grid Based Method:** based on a multiple-level granularity structure

#### 3.4.3.1 Partitioning Methods

Construct a partition of a database  $D$  of  $n$  objects into a set of  $k$  clusters  
 Given a  $k$ , find a partition of  $k$  clusters that optimizes the chosen partitioning criterion

It consist of these algorithms :

- K-means
- K-medoid
- K-Plane
- CLARANS (A Clustering Algorithm based on Randomized Search) (Ng and Han'94) it is improved version of k-mediod
- K-modes: extension of k-means
- Bisecting K-means: better results then k-mean
- PAM (Kaufman and Rousseeuw, 1987), :starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
- CLARA (Kaufmann and Rousseeuw in 1990): It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output

### 3.4.3.2 Hierarchical Methods

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

It consist of these algorithms :

- AGNES (Agglomerative Nesting): [7] Introduced in Kaufmann and Rousseeuw (1990) In the K-means approach to clustering, we start out with a fixed number of clusters and gather all records into them. There is another class of methods that work by agglomeration. In these methods, we start out with each data point forming its own cluster and gradually merge clusters until all points have been gathered together in one big cluster. Towards the beginning of the process, the clusters are very small and very pure – the members of each cluster are few, but very closely related. Towards the end of the process, the clusters are large and less well-defined. The entire history is preserved so you can choose the level of clustering that works best for your application.
- DIANA (Divisive Analysis) Introduced in Kaufmann and Rousseeuw (1990)
- BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
- CURE (1998): selects well-scattered (representative) points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
- CHAMELEON (1999): hierarchical clustering using dynamic modeling

### 3.4.3.3 Density Based Methods

The can discover clusters with arbitrary shapes and they do not need to present number of cluster.

It consist of these algorithms :

- DBScan
- TURN
- EM (Expectation Maximization)

### 3.4.3.4 Grid-Based Methods

It first quantize the clustering space into finite number of cells and then perform clustering on the girded cells.

It consist of these algorithms :

- WaveCluster
- Clique



## *Chapter 4*

### *Problem and Elucidation*

## 4. PROBLEM AND ELUCIDATION

Data Mining is mostly done on bulk data. The main problem that is encountered during data mining is that of *Time*. Usually, data mining queries takes hours or may be a day to get the results. The problem is of making efficient technique that should take less time and produce better results. In this Chapter we have given a new technique that takes lesser time while producing better results. This technique is called the K-MT technique, which is a modified version of K-Means technique.

### 4.1 K-Means

The K-Means clustering algorithm classifies  $n$  points into  $k$  clusters by assigning each point to the cluster whose average value on a set of  $p$  variables is nearest to it by some distance measure (usually Euclidean) on that set. The algorithm computes these assignments iteratively, until reassigning points and re computing averages (over all points in a cluster) produces no changes.

Another way to describe the K-Means algorithm is to define its goal geometrically. If  $n$  points are embedded in a  $p$ -dimensional space, then  $k$  clusters are summarized by their respective centroids (average of the cluster members' coordinates) in that space. Computing a K-Means clustering involves identifying a set of cluster centroids, which implicitly solves for the planes. There are many methods for finding a satisfactory set of centroids given a set of data. The simplest is to pick an initial set of centroid seeds randomly (assuming that how many clusters we want) and to assign each point to its closest seed. After each assignment, we need to update the assigned centroid by adding in the coordinates of the new point (a simple calculation). Assigning all the points to a set of successively updated centroids constitutes an iteration of the K-Means algorithm.

Each new iteration consists of a re-assignment of all points, until no point can be moved to a centroid closer than the one for the cluster it is already a member of. Every time a point is re-assigned, its old centroid must be down dated and its new centroid must be updated.

Starting with arbitrary random centroids is a relatively poor method. However, if there really are blobs of points, we do much better to begin with locations that are relatively close to the center of these blobs. John Hartigen (*Clustering Algorithms*, Wiley, 1975) suggests a stage wise method to approach this goal. We begin with two centroids. These are computed by splitting all points on the variable (dimension) with greatest range (or some other measure of spread). The split separates points above the mean (or some other measure of location) on this variable from those below the mean. Centroids are then computed for each group by averaging coordinates of its members. Then we do an entire K-Means cycle (assignments plus iterations). After convergence, we move to computing three centroids. Again, we split the cluster having the largest range on a variable into two clusters (one above and one below the mean on this variable). New centroids are computed and another K-Means cycle is performed. This process is continued until we reach the desired number ( $k$ ) of centroids.

If we do not know  $k$  in advance, how can we choose a value based on our data? There is circularity here: we need to know  $k$  to find clusters and we need to identify clusters to determine  $k$ . Hartigan's procedure gives us a lever. At each stage of Hartigan's method, we compute the sum-of-squares within groups over all variables (sum of squared deviations of each point from its centroid on every dimension). This sum of squares should decline as we add new clusters; indeed, it would be zero if we made every point a cluster. So we look for the reduction in sum of squares at each step and stop adding clusters when this reduction is negligible.

#### 4.1.1 K-Means Clustering Method

For a given  $k$ , the K-Means algorithm is implemented in 4 steps:

1. Partition objects into  $k$  nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
3. Assign each object to the cluster with the nearest seed point.
4. Go back to Step 2 and stop when no more new assignment

#### 4.1.2 Example

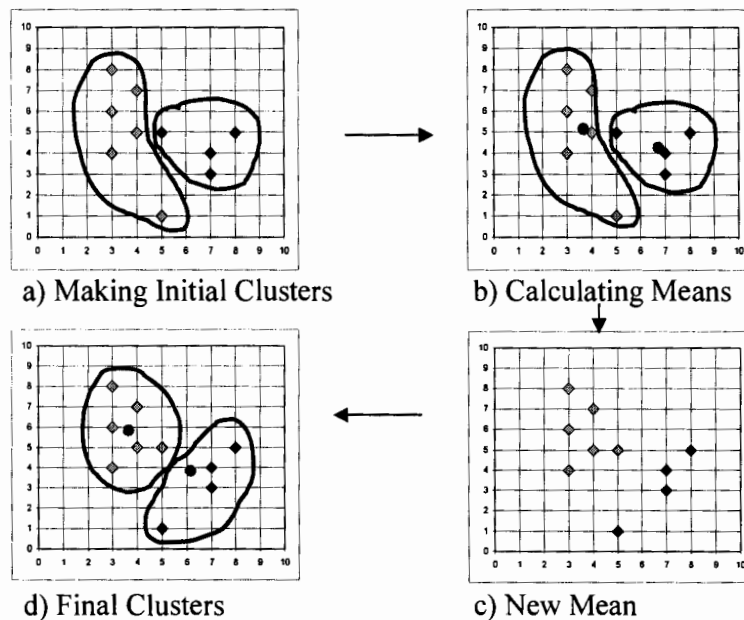


Fig 4.1 K-Means Clusters

In this figure we can see that first the clusters are made on randomly assigning the mean. Then the mean is recalculated and new clusters are made on the bases of new mean. The process goes on until finally the clusters are made.

### 4.1.3 Strengths and Weaknesses

#### ➤ Strengths

- Relatively efficient:  $O(tkn)$ , where  $n$  is # objects,  $k$  is # clusters, and  $t$  is # iterations. Normally,  $k, t \ll n$ .
- Often terminates at a local optimum. The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms.

#### ➤ Weaknesses

- Applicable only when mean is defined, then what about categorical data?
- Need to specify  $k$ , the number of clusters, in advance.
- Unable to handle noisy data.
- Not suitable to discover clusters with non-convex shapes.

### 4.1.4 Variations of K-Means

#### ➤ A few variants of the K-Means which differ in

- Selection of the initial  $k$  means
- Dissimilarity calculations
- Strategies to calculate cluster means

#### ➤ Handling categorical data: *k-modes* (Huang'98)

- Replacing means of clusters with modes
- Using new dissimilarity measures to deal with categorical objects
- Using a frequency-based method to update modes of clusters
- A mixture of categorical and numerical data: *k-prototype* method

## 4.2 K-MT

K-MT (K-MohsinTauseef) is a new algorithm that is a modified version of the K-Means algorithm. K-Means is the basic algorithm that is used in the partitioning method of clustering. The K-MT clustering algorithm classifies  $n$  points into  $k$  clusters by assigning each point to the cluster whose average value on a set of  $p$  variables is nearest to it by some distance measure on that set. The algorithm computes these assignments iteratively, until reassigning points and recompiling averages (over all points in a cluster) produces no changes.

### 4.2.1 Variation in K-MT.

The variation in K-MT, which makes it different from the K-Means, is that in K-Means we select the initial mean or seed randomly but in K-MT we calculate the initial mean our self and then do the calculations on that mean. When we have calculated the initial mean then we apply the K-Means algorithm to the clusters. In this, we also need to know the  $k$  number of clusters and  $n$  the number of inputs.

### 4.2.2 K-MT Method

Given  $k$ , the  $k$ -MT algorithm is implemented in 5 steps:

- I. Consider all points as one big cluster.
- II. Calculate mean of this cluster. (KMT Step)
- III. Divide the cluster into two clusters such that :
  - a) A cluster consist of all the points less then mean
  - b) B cluster consist of all the points greater then mean.
- IV. Repeat step 3, for ITER times and take the split that produces the clustering with the highest overall similarity.
- V. Repeat steps 1, 2 and 3 until the desired number of clusters is reached

### 4.3 Comparison Example

The difference will be clear by this example

#### 4.3.1 Problem

We are given a set  $G: \{2,4,10,12,3,20,30,11,25\}$  and we would like to make two clusters from it that is  $k=2$ .

#### 4.3.2 Solution by K-Means

If we apply K-Means on this given set, we would have to assign two means also randomly picked form the given set  $G$ . Lets us say the means selected are  $m_1 = 3$  and  $m_2 = 4$ , apply K-Means on the  $G$  set:

$$K_1 = \{2,3\}, K_2 = \{4,10,12,20,30,11,25\},$$

Recalculating the mean we get:

$$m_1 = 2.5, m_2 = 16$$

$$K_1 = \{2,3,4\}, K_2 = \{10,12,20,30,11,25\}$$

Recalculating the mean we get:

$$m_1 = 3, m_2 = 18$$

$$K_1 = \{2,3,4,10\}, K_2 = \{12,20,30,11,25\}$$

Recalculating the mean we get:

$$m_1 = 4.75, m_2 = 19.6$$

$$K_1 = \{2,3,4,10,11,12\}, K_2 = \{20,30,25\}$$

Recalculating the mean we get:

$$m_1 = 7, m_2 = 25$$

Now the clusters made are final and no further the iteration can be made on it. So the total number of iterations in this are 4.

### 4.3.3 Solution by K-MT

If we apply k-MT on this given set, we would have to calculate the mean our self which comes out to be 13. so  $m=13$ . We divide the given set in two clusters  $K_1$  and  $K_2$  such that  $K_1$  consist of elements less then or equal to the mean  $m$  and  $K_2$  consist of elements greater then the mean  $m$ .

$$K_1=\{2,3,4,10,11,12\}, K_2=\{20,30,25\}$$

Recalculate the mean and we get it as:

$$m_1=7, m_2=25$$

Now the clusters made are final and no further iteration can be made on it. Total 1 number of iteration is made.

### 4.3.4 Conclusion

The number of iterations has reduced from **4 to 1** immediately. This shows that K-MT approach is much better then the K-Means. Only because of calculating the mean instead of assigning it randomly, we have reduced the number of iterations from 4 to 1. We also calculated only one mean and manipulated that to make two clusters.

## 4.4 Bisecting Step

Bisecting Step is a technique in which we decide which cluster is to be further subdivided. If we had to make three clusters from it then what will we do? For that purpose it depends upon the algorithm which uses to select the third cluster. K-Means may select third cluster by selecting any random cluster and dividing it into further clusters. K-MT uses the cluster with bigger data set to be divided into further clusters. This is also an important issue that how we select the cluster to be further divided. Various techniques are available by which the clusters can to be selected. Bisecting algorithm is an example which helps us to selects which cluster is to be further divided. Bisecting method is itself an area in which research can be done.

## 4.5 Architecture of K-MT

The architecture of my research K-MT can easily be understood through this figure.

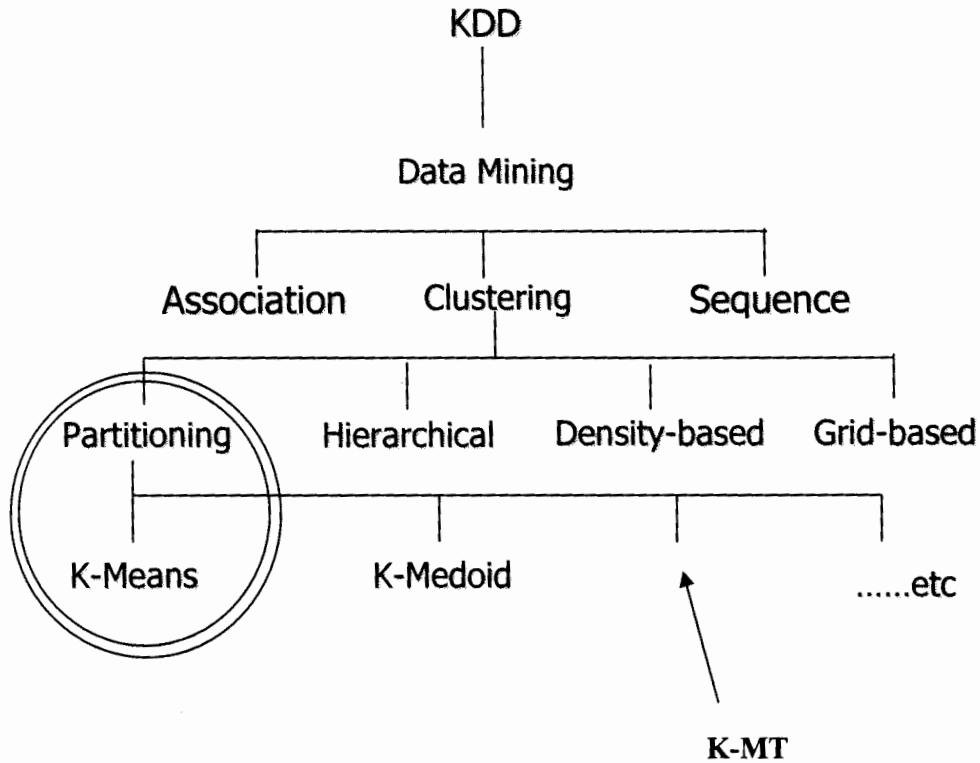


Fig 4.2 Architecture of K-MT

The basic starts from the Knowledge Discovery and Data Mining KDD field. Data Mining is further classification of it. Data Mining itself includes further three sub categories association, clustering and sequence.

The Clustering is divided into 4 sub categories partitioning, hierarchical, density based and grid-based clustering. The algorithms of my research are from the partitioning part. It has many different algorithms such as K-Means, K-Mode, K-Mediod etc. My algorithm K-MT also fits in this area. This diagram clearly states that what is my research area and what I have worked on.

## 4.6 Future Research Areas

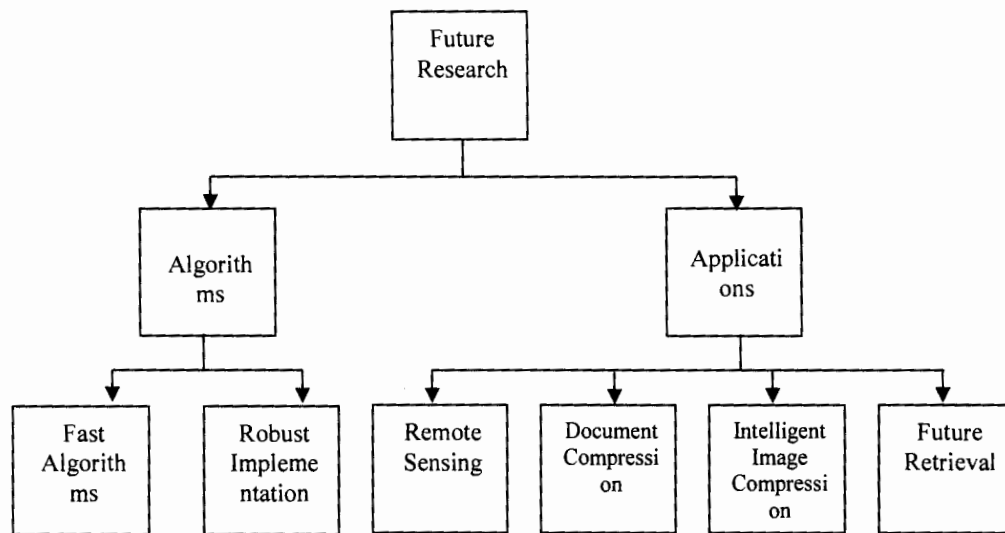


Fig 4.3 Future Research Areas



## *Chapter 5*

# *Software Testing and Results*

## 5. SOFTWARE TESTING AND RESULTS

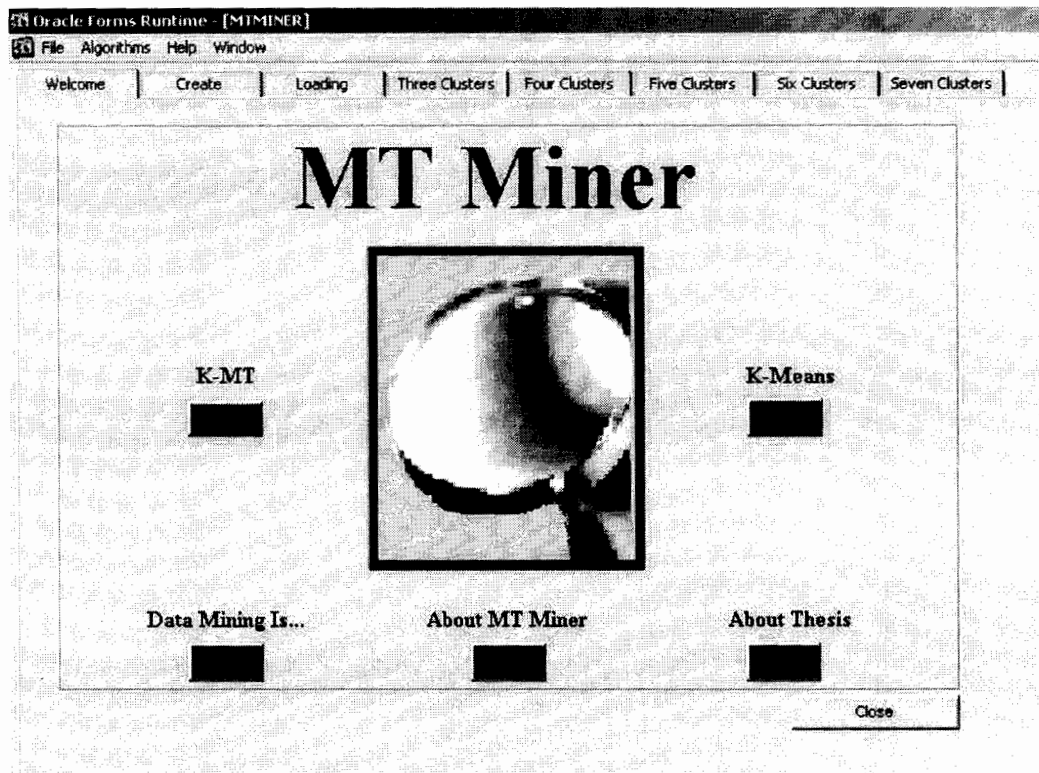
MT Miner is the software that is used for the demonstration of my algorithm K-MT and K-Mean. This software is tested on different number of data sets and the different results are obtained from it showing that K-MT is better than the K-Mean method.

MT Miner calculates the time taken by the algorithm when it is given number of input clusters to be made. It makes the clusters that are required by the user. It also then generates the reports of the clusters generated. The reports are run and could be printed if wanted. A brief introduction is also available of both the algorithms in this software.

### 5.1 Data Sets

The data set consists of different number of records. We have tested these algorithms on different number of records consisting from 20400 records to 500 records. Data mining algorithms are usually run on very high number of records data sets. The result is usually after hours or days. This result which we get is then used for the business use of the organization. Initially before running MT Miner there should be a set of records available in the data base.

## 5.2 Main



*Fig 5.1 Main Screen Shot*

This is the main form of our software. It has different buttons that can be used by the user for different purposes.

**Data Mining Is..** button will lead to a form that describes the brief definition of data mining and other processes.

**About MT Miner** will lead to the form that describes the function of MT Miner and its different uses.

**About Thesis** will open a form that tells us about the thesis and research work.

**K-MT** button will open the form that contains the K-MT algorithm.

**K-Means** button will open the form that contains the K-Means algorithm.

**Close** will exit the program.

### 5.3 What is Data Mining?

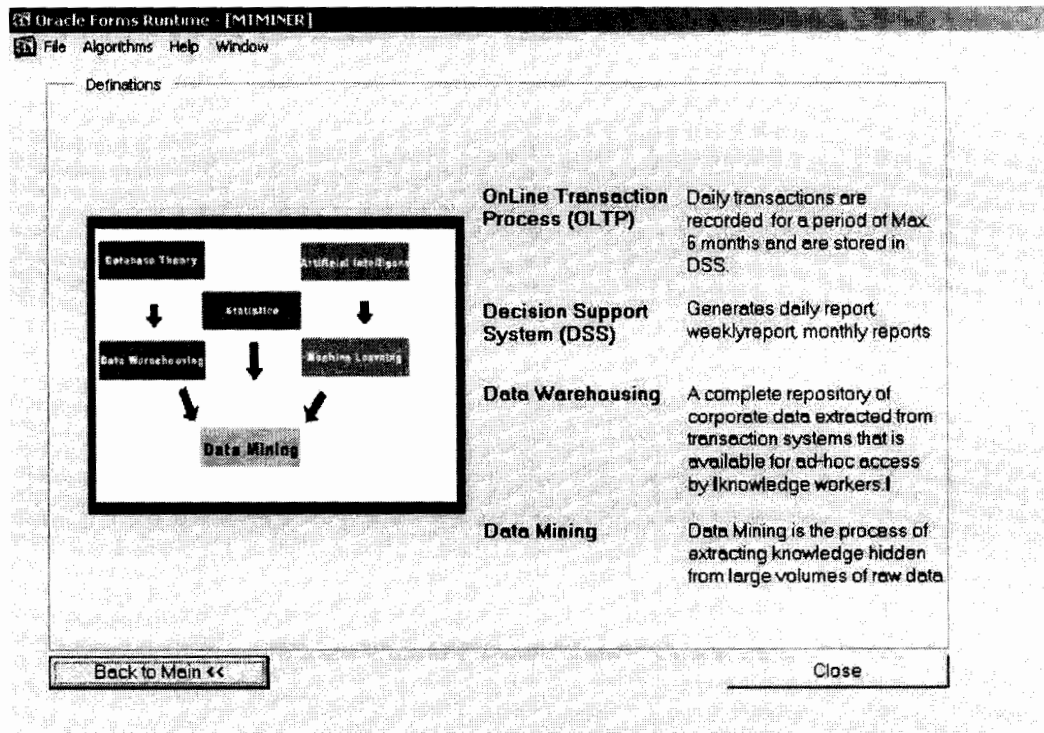
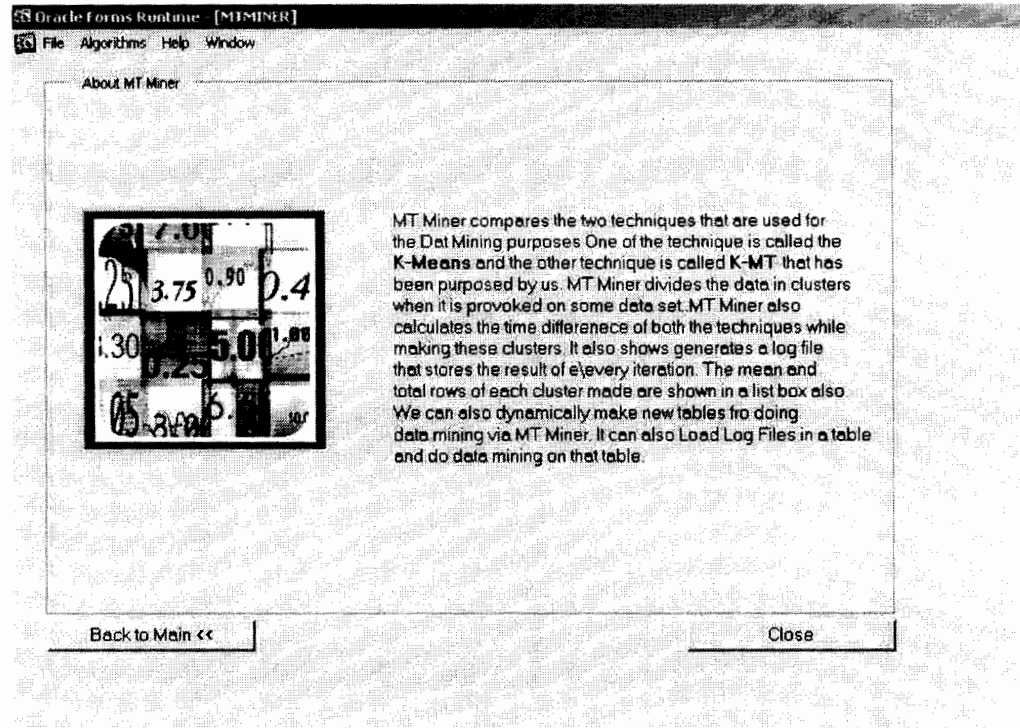


Fig 5.2 What is Data Mining Screen Shot

This is the form that is opened when we press the **What is Data Mining?** button from main form. This page simply displays the definitions of different concepts that are needed to understand the data mining concepts. OLTP, DSS and Data Warehousing are the processes that are to be completed before we can do the mining process.

**Back to Main <<** will lead back to the main form.  
**Close** will exit this form.

## 5.4 About MT Miner



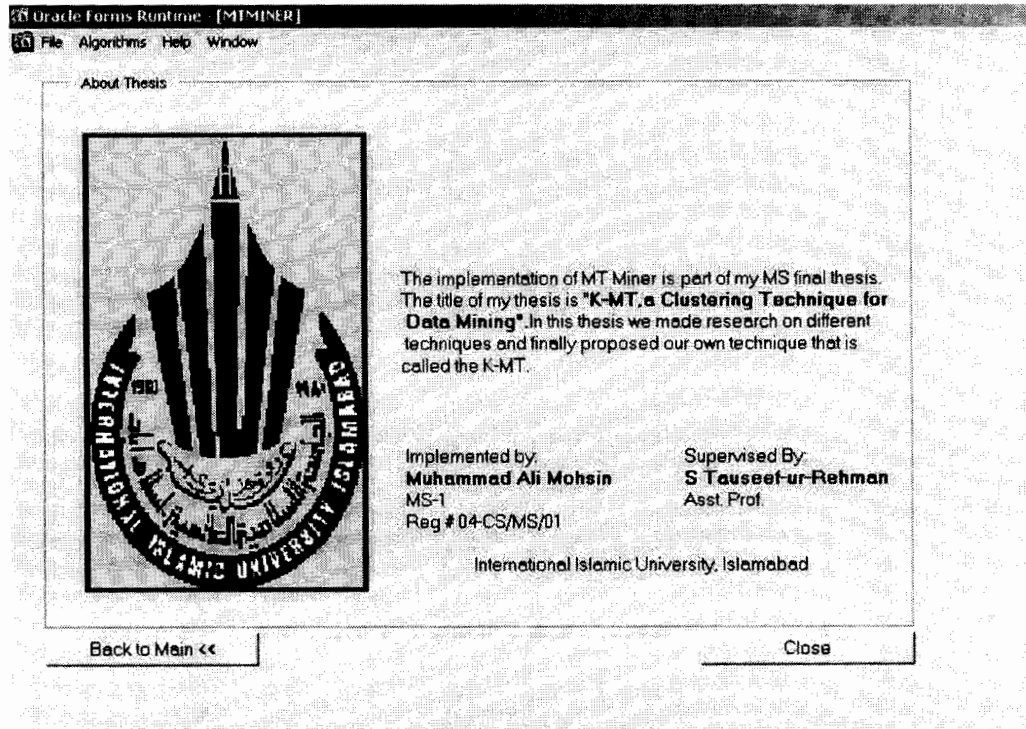
*Fig 5.3 About MT Miner Screen Shot*

This is the form that is achieved when **About MT Miner** is pressed. This form briefly tells about the MT Miner.

**Back to Main <<** will lead back to the main form.

**Close** will exit this form.

## 5.5 About Thesis



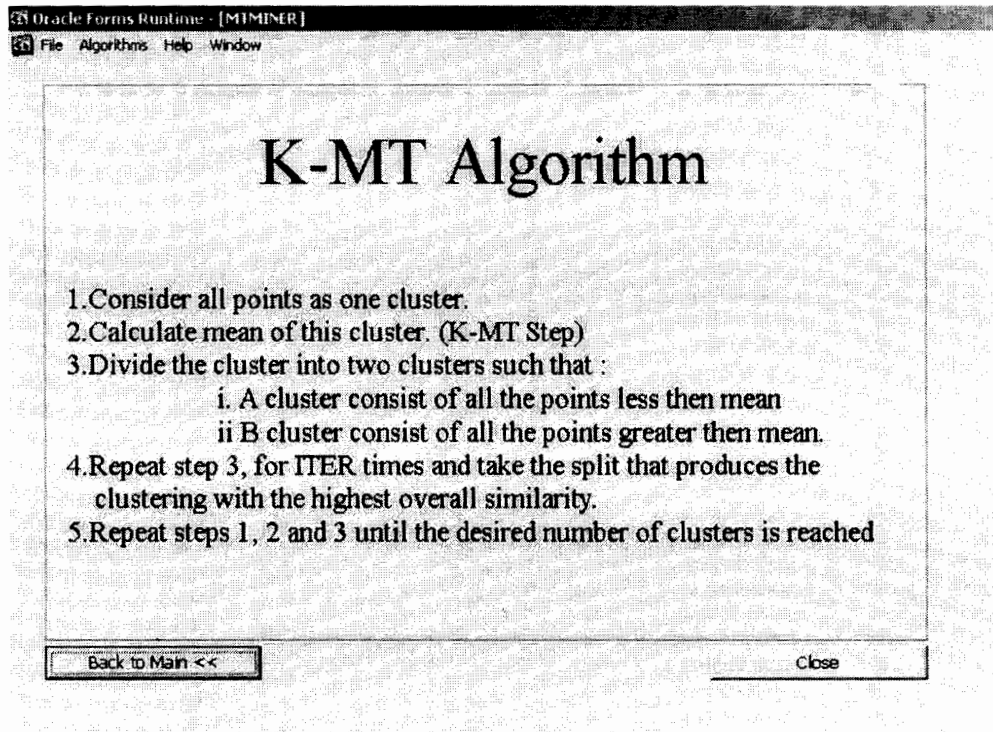
*Fig 5.4 About Thesis Screen Shot*

This form opens when **About Thesis** button is pressed. This form gives a brief view about our thesis.

**Back to Main** << will lead back to the main form.

**Close** will exit this form.

## 5.6 K-MT



*Fig 5.5 K-MT Algorithm Screen Shot*

This form opens when **K-MT** button is pressed. This form shows the K-MT algorithm.

**Back to Main <<** will lead back to the Main form

**Close** will exit this form

## 5.7 K-Means

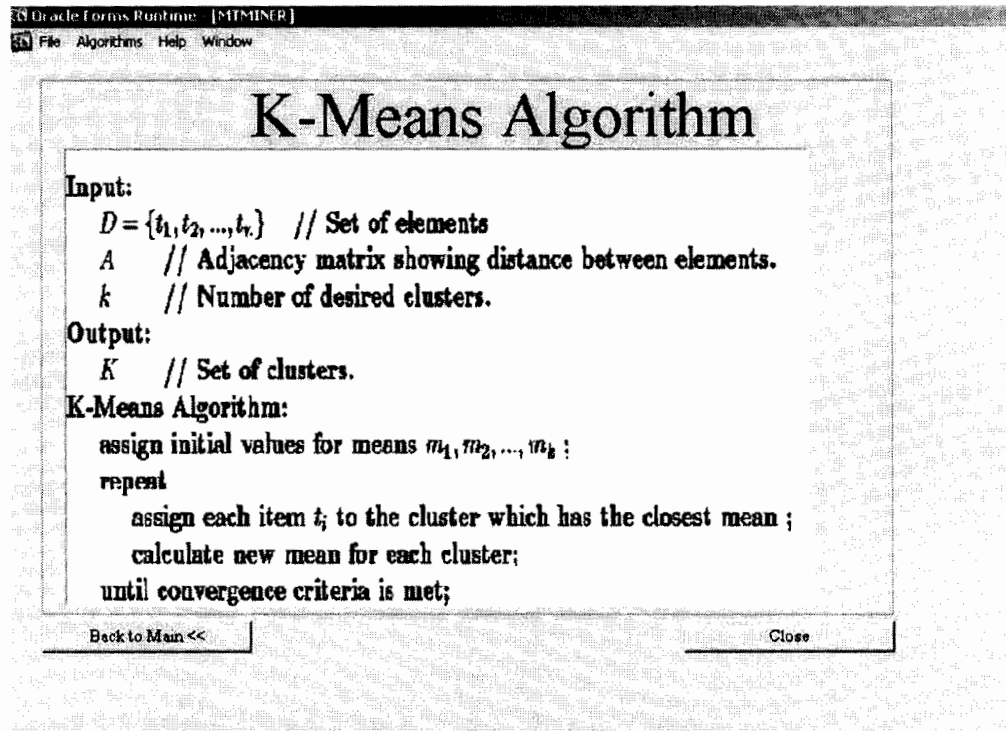


Fig 5.6 K-Means Algorithm Screen Shot

This form opens when **K-Means** button is pressed. This form shows the K-Means algorithm.

**Back to Main <<** will lead back to the main form

**Close** will exit this form



## 5.8 Create Table

Oracle Forms Runtime - [MTMINER]

File Algorithms Help Window

Welcome | Create | Loading | Three Clusters | Four Clusters | Five Clusters | Six Clusters | Seven Clusters

Table

Table Name

Column

Column Name	Type	Length	Constraint	Reference Table
			NONE	

Create | New Table | Cancel

*Fig 5.7 Create Table Screen Shot*

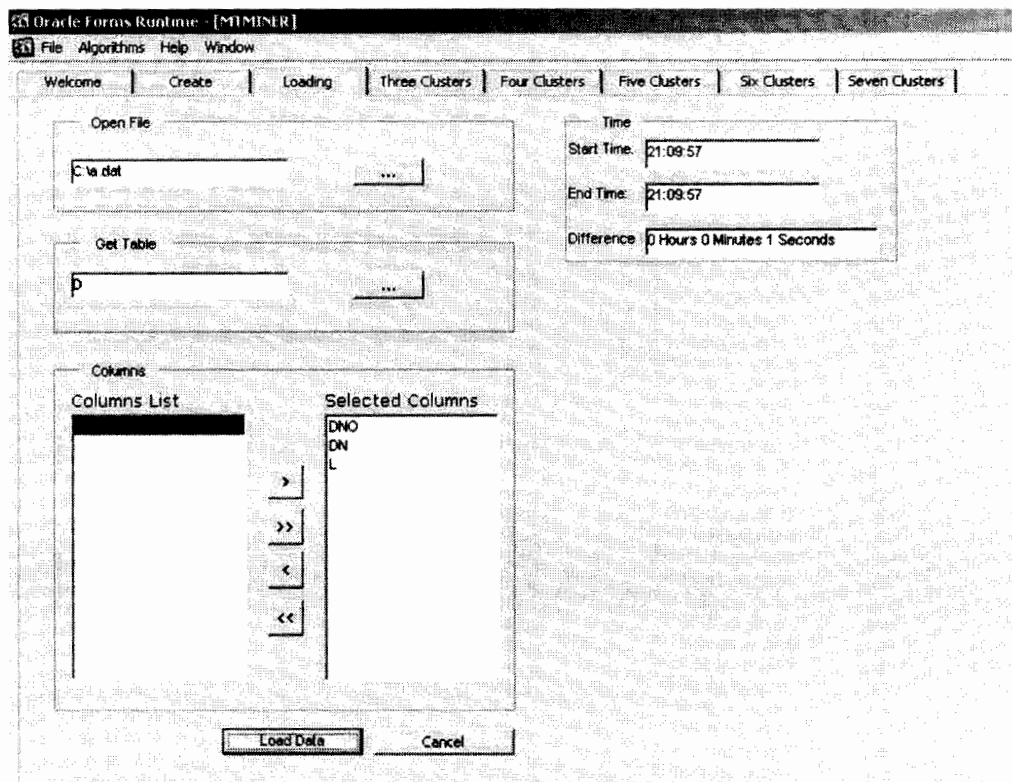
This form will create tables at runtime for the loading of the data if the data from warehouse is in flat files. One can create the table by giving table name and rest of parameters. All the constraints can also be implemented via this form.

**Create** will create the table in the data base and also check weather that table already exist in the data base or not

**New Table** is the option that is used when we want to create more then one table

**Cancel** will discard all the changes

## 5.9 Loading Data



*Fig 5.8 Loading Data Screen Shot*

The loading form is responsible for loading the data from log/flat/data files into the table already in data bases or created via the create form. In this form we have to first assign the file path and the table to load. One can also select the columns fro the loading purpose. The start time and end time is also taken along with the time taken to load the file in the table.

**Load Data** button will load the data from the file specified in the table field that are specified

**Cancel** will discard all the items

## 5.10 Three Clusters

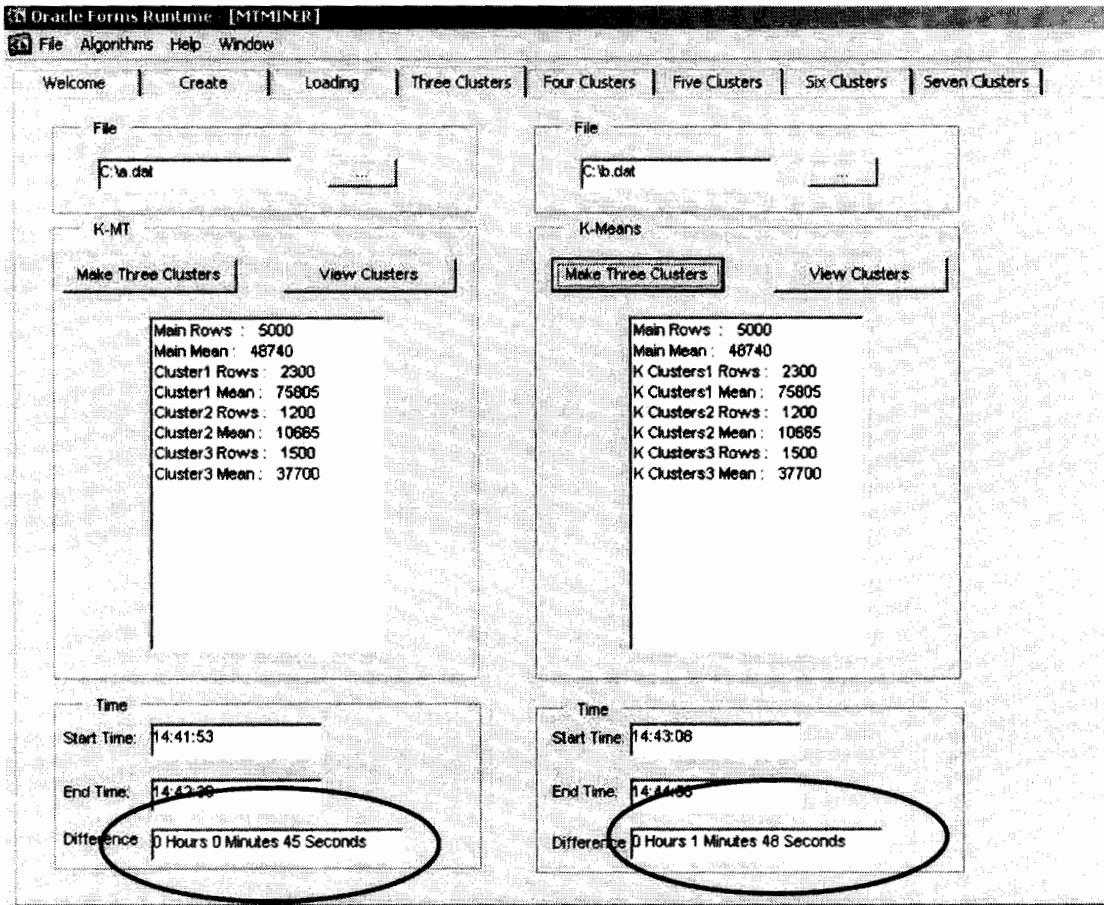


Fig 5.9 Three Clusters Screen Shot

This is the form that creates three clusters of the given set and shows its result. Here also the difference of time is taken that plays a key role in this form. The area marked in blue circle highlights the time taken by two algorithms to make same number of clusters from the same data set provided. The results are saved in a log file on the path specified.

**File** specify the path where to store the log/data file .

**Make Three Clusters** will make three clusters from the given data set and show their result

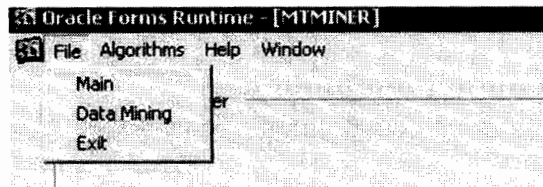
**View Clusters** will show the clusters physically from the data base in report form

## 5.11 Four/Five/Six/Seven Clusters

All these forms are similar to the previous form 5.10. The difference is only in the number of cluster that are to be made. Similarly the time is calculated and displayed at end of form.

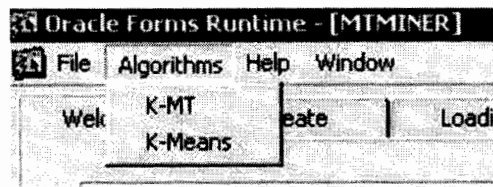
## 5.12 Menus

There is menu also available to manipulate the MT Miner.



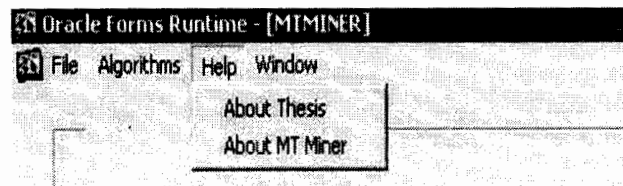
*Fig 5.10 File Menu Screen Shot*

This is the File menu that has three options, one of going to main page, other for going to data mining page and last for the exit.



*Fig 5.11 Algorithms Menu Screen Shot*

This is Algorithms menu that has two different options, K-MT will open the form that has K-MT algorithm and K-Means will open K-Means algorithms page.



*Fig 5.12 Help Menu Screen Shot*

This is help menu and it has two options. About thesis will open a form that will tell about the thesis and research, About MT Miner will show the description how MT Miner works.

### 5.13 Working Bar



*Fig 5.13 Working Bar Screen Shot*

This left lower side of the final form shows that the algorithm is running and how many percent the algorithm has been run.

### 5.14 Reports

The screenshot shows a window titled "Cluster Number 1" with a menu bar (File, View, Help) and a toolbar. The main content is a table with the following data:

Snu	Name	Sal	Iy
1	Mahesh	95000	99
4	Walia	98766	98
15	Walia	95555	98
16	Daasheal	89444	1
18	Dasha	55665	9
19	Dasha	66556	7
20	Dasha	99889	7
21	Dasha	55456	2
29	Dasha	65345	1
24	Dasha	764344	99
25	Dasha	58975	97
26	Dasha	79846	98
28	Dasha	94997	99
25	Dasha	65997	99
27	Dasha	66289	99
28	Dasha	88110	99
29	Dasha	70009	98
31	Dasha	99000	7
45	Dasha	55000	7
46	Dasha	99000	98
47	Dasha	60000	97
48	Dasha	64000	99
50	Dasha	88004	99

*Fig 5.14 Reports Screen Shot*

Reports for different clusters are generated in this form and the prints could easily be taken. Reports of all the three clusters are shown when View Clusters is pressed on the Three Clusters form. If the report is of more than one page then it will show page numbers also.

## 5.15 Results

Different results are obtained when MT Miner is executed on different data sets. Some of these results are as follows.

No. of clusters	Data Set	Time Taken by						Difference		
		K-MT			K-Mean			Hr	Mi	Sec
		Hr	Mi	Sec	Hr	Mi	Sec			
3	Ds1	0	0	01	0	0	01	0	0	00
4	Ds1	0	0	02	0	0	02	0	0	00
5	Ds1	0	0	03	0	0	05	0	0	02
6	Ds1	0	0	04	0	0	08	0	0	04
7	Ds1	0	0	04	0	0	08	0	0	04
3	Ds2	0	0	05	0	0	10	0	0	05
4	Ds2	0	0	06	0	0	11	0	0	05
5	Ds2	0	0	06	0	0	11	0	0	05
6	Ds2	0	0	06	0	0	12	0	0	04
7	Ds2	0	0	08	0	0	12	0	0	04
3	Ds3	0	0	46	0	1	51	0	1	05
4	Ds3	0	0	52	0	2	00	0	1	08
5	Ds3	0	0	58	0	2	05	0	1	09
6	Ds3	0	1	05	0	2	13	0	1	08
7	Ds3	0	1	11	0	2	16	0	1	05
3	Ds4	0	2	23	0	5	27	0	3	04
4	Ds4	0	3	00	0	8	30	0	5	30
5	Ds4	0	3	35	0	8	27	0	4	52
6	Ds4	0	3	53	0	8	46	0	4	53
7	Ds4	0	4	00	0	8	54	0	4	54
3	Ds5	0	9	44	0	28	25	0	18	41
4	Ds5	0	9	35	0	28	28	0	16	53
5	Ds5	0	9	39	0	28	35	0	16	56
6	Ds5	0	9	36	0	28	37	0	19	01
7	Ds5	0	10	01	0	29	04	0	19	03

*Fig 5.15 Different Results from Different Data Sets*

Note:

Ds1 is the data set consisting of 500 records.

Ds2 is the data set consisting of 1000 records.

Ds3 is the data set consisting of 5000 records.

Ds4 is the data set consisting of 10,000 records.

Ds5 is the data set consisting of 20,000 records.

## *Appendix*

**APPENDIX: ABBREVIATION**

<b>K-DD</b>	<b>Knowledge Discovery and Data Mining</b>
<b>K-MT</b>	<b>K-MohsinTauseef</b>
<b>OLTP</b>	<b>On-Line Transaction Process</b>
<b>DSS</b>	<b>Decision Support System</b>
<b>TB</b>	<b>Tera Byte</b>
<b>IBM</b>	<b>International Business Machine</b>
<b>OLAP</b>	<b>On-Line Analytical Process</b>
<b>FDM</b>	<b>Fast Distributed Mining of association rules</b>
<b>K-NN</b>	<b>K-Nearest Neighbor</b>
<b>CHAID</b>	<b>Chi-square Automatic Interaction Detection</b>
<b>CHART</b>	<b>Classification and Regression Tree</b>
<b>SLIQ</b>	<b>Supervised Learning in Quest</b>
<b>AGNES</b>	<b>Agglomerative Nesting</b>
<b>DIANA</b>	<b>Divisive Analysis</b>
<b>BIRCH</b>	<b>Balanced Iterative Reducing and Clustering</b>
<b>CLARANS</b>	<b>Clustering Large Applications based upon RANdomized Search.</b>
<b>CSAR</b>	<b>Clustered Spatial Association Rule</b>



## REFERENCE AND BIBLIOGRAPHY

1. [AGY99] Charu C. Aggarwal, Stephen C. Gates and Philip S. Yu, *on the merits of building categorization systems by supervised clustering*, Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 352 – 356, 1999.
2. [APR97] Javed Aslam, Katya Pelehov, and Daniela Rus, *A Practical Clustering Algorithm for Static and Dynamic Information Organization*, Proceedings of the 1998 ACM CIKM International Conference On Information and Knowledge Management, Bethesda, Maryland, USA, Pages 208-217, November 3-7, 1998.
3. [BF98] Paul Bradley and Usama Fayyad, *Refining Initial Points for K-Means Clustering*, Proceedings of the Fifteenth International Conference on Machine Learning ICML98, Pages 91-99. Morgan Kaufmann, San Francisco, 1998.
4. Hand, D. J. (1981) *Discrimination and Classification*, John Wiley & Sons.
5. [CCFW98] Moses Charikar, Chandra Chekuri, Tomas Feder, and Rajeev Motwani, *Incremental Clustering and Dynamic Information Retrieval*, STOC 1997, Pages 626-635, 1997.
6. [DJ88] Richard C. Dubes and Anil K. Jain, *Algorithms for Clustering Data*, Prentice Hall, 1988.
7. [EW89] A. El-Hamdouchi and P. Willet, *Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval*, The Computer Journal, Vol. 32, No. 3, 1989.
8. [GRS99] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, (1998), *ROCK: A Robust Clustering Algorithm for Categorical Attributes* In Proceedings of the 15th International Conference on Data Engineering, 1999.
9. [Kow97] Gerald Kowalski, *Information Retrieval Systems – Theory and Implementation*, Kluwer Academic Publishers, 1997.
10. [KR90] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, JohnWiley and Sons, 1990.
11. [LA99] Bjorn Larsen and Chinatsu Aone, *Fast and Effective Text Mining Using Linear-time Document Clustering*, KDD-99, San Diego, California, 1999.
12. [Rij79] C. J. van Rijsbergen, (1989), *Information Retrieval*, Butterworth, London, second edition.

13. SPATH H. 1980. *Cluster Analysis Algorithms*. Ellis Horwood, Chichester, England.
14. [ZEMK97] Oren Zamir, Oren Etzioni, Omid Madani, Richard M. Karp, *Fast and Intuitive Clustering of Web*
15. [Agrawal and Shafer1996] R. Agrawal and J. C. Shafer. Parallel mining of association rules: Design, implementation, and experience. *IEEE Trans. Knowledge Data Eng.*, 8(6):962–969, 1996.
16. [Alsabti et al.1998] K. Alsabti, S. Ranka, and V. Singh. An efficient  $k$ -means clustering algorithm. In *Proceedings of IPPS/SPDP Workshop on High performance Data Mining*, 1998.
17. [Bottou and Bengio1995] L. Bottou and Y. Bengio. Convergence properties of the  $k$ -means algorithms. In G. Tesauro and D. Touretzky, editors, *Advances in Neural Information Processing Systems 7*, pages 585–592. The MIT Press, Cambridge, MA, 1995.
18. [Chattratchat et al.1997] J. Chattratchat, J. Darlington, M. Ghanem, Y. Guo, H. Hünig, M. Köhler, J. Sutiwaraphun, H. W. To, and D. Yang. Large scale data mining: Challenges and responses. In D. Pregibon and R. Uthurusamy, editors, *Proceedings Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA*, pages 61–64. AAAI Press, 1997.
19. [Cheung and Xiao1999] D. W. Cheung and Y. Xiao. Effect of data distribution in parallel mining of associations. *Data Mining and Knowledge Discovery*, 1999.
20. JAIN, A. and DUBES, R. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ.
21. [Dhillon et al.1998] I. S. Dhillon, D. S. Modha, and W. S. Spangler. Visualizing class structure of Multidimensional data. In S. Weisberg, editor, *Proceedings of the 30th Symposium on the Interface Computing Science and Statistics, Minneapolis, MN*, May 13–16 1998.
22. [Fayyad et al.1996] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.
23. FORGY, E. 1965. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21, 768–780.
24. [Fukunaga and Narendra1975] K. Fukunaga and P. M. Narendra. A branch and bound algorithm for computing  $k$ -nearest neighbors. *IEEE Trans. Computer*, pages 750–753, 1975.
25. KAUFMAN, L. and ROUSSEEUW, P. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY.

26. EVERITT, B. 1993. *Cluster Analysis* (3rd ed.). Edward Arnold, London, UK.
27. MIRKIN, B. 1996. *Mathematic Classification and Clustering*. Kluwer Academic Publishers.
28. JAIN, A.K, MURTY, M.N., and FLYNN P.J. 1999. Data clustering: a review. *ACM Computing Surveys*, 31, 3, 264-323.
29. [Hartigan1975] J. A. Hartigan. *Clustering Algorithms*. Wiley, 1975.
30. FASULO, D. 1999. An analysis of recent work on clustering algorithms. *Technical Report UW-CSE01 -03-02, University of Washington*.
31. KOLATCH, E. 2001. Clustering Algorithms for Spatial Databases: A Survey. PDF is available on the Web..
32. [McLachlan and Krishnan1996] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1996.[Milligan1985] G. Milligan. An algorithm for creating artificial test clusters. *Psychometrika*, 50(1):123–127, 1985.
33. [Shafer *et al.*1996] J.C. Shafer, R. Agrawal, and M. Mehta. A scalable parallel classifier for data mining. In *Proc. 22nd International Conference on VLDB, Mumbai, India, 1996*.
34. HAN, J. and KAMBER, M. 2001. *Data Mining*. Morgan Kaufmann Publishers.
35. [Smyth *et al.*1997] P. Smyth, M. Ghil, K. Ide, J. Roden, and A. Fraser. Detecting atmospheric regimes using cross-validated clustering. In D. Pregibon and R Uthurusamy, editors, *Proceedings Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA*, pages 61–64. AAAI Press, 1997.
36. [AKT 03] A K Chaudhry, K Rashid, S Tauseef “*Data Profiling for transformation engine in VLDBs*”
37. [MT 2003] M A Mohsin, S Tauseef. “*K-MT: A Modified K-Means Algorithm for Data Mining*” *Data Warehousing Journal*, SE, USA
38. GHOSH, J., 2002. Scalable Clustering Methods for Data Mining. In Nong Ye (Ed.) *Handbook of Data Mining*, Lawrence Erlbaum, to appear.
39. “*Data Mining: Concepts and Techniques*” by J. Han and M. Kamber, 2001.
40. “*Pattern Classification*” by R. Duda, P. Hart and D. Stork, 2001 (1st ed. 1973).

*Acceptance Mail*

ADVERTISEMENT



Mail Addresses Calendar Notepad

stauseef@yahoo.com [Sign Out]

Check Mail

Compose

Mail Upgrades - Search Mail - Mail Options

Folders [Add]

Inbox (11)

Draft

Sent

Bulk (1) [Empty]

Trash [Empty]

My Folders [Hide]

Asynchrony

CD Requests

DownLoads

Jobs

Memberships

Netnotes

Personal

Update

mmail

How Bad is Your Credit?

Free: Access Your PC from Anywhere

FREE fax number for your Yahoo! Mail

Compare

Previous | Next | Back to Messages

Printable View - Full Headers



This message is not flagged. [ Flag Message - Mark as Unread ]

Date: Wed, 2 Apr 2004 09:54:12 GMT

To: stauseef@yahoo.com

Subject: Acceptance of Paper

From: "Maria Gonzalez" <m.gonzalez@dwj.com> | This is not spam | Add to Address Book

Dear Mr S.Tauseef Ur Rehman
Your Paper titled "K-MT: A Modified K-Mean Algorithm for Data Mining" has been accepted and will be published in September issue of DataWarehousing Journal. Five copies of the print will be sent to you free of charge, however you can order more copies by paying the \$8.00 per copy.

Once again, I congratulate you for the acceptance of letter.

Best Regards,

ADVERTISEMENT



Mail -  Addresses -  Calendar -  Notepad -

[stauseef@yahoo.com](mailto:stauseef@yahoo.com) [Sign Out]

[Check Mail](#)

[Compose](#)

[Mail Upgrades](#) - [Search Mail](#) - [Mail Options](#)

[Folders](#) [Add]

[Inbox \(11\)](#)

[Draft](#)

[Sent](#)

[Bulk \(1\)](#) [Empty]

[Trash](#) [Empty]

[My Folders](#) [Hide]

[Asynchrony](#)

[CD Requests](#)

[DownLoads](#)

[Jobs](#)


[Memberships](#)


[Netnotes](#)


[Personal](#)


[Update](#)

[mmail](#)

 [How Bad is Your Credit?](#)

 [Free: Access Your PC from Anywhere](#)

 [FREE fax number for your Yahoo! Mail](#)

 [Compare](#)

[Previous](#) | [Next](#) | [Back to Messages](#)

[Printable View](#) - [Full Headers](#)



This message is not flagged. [ [Flag Message](#) - [Mark as Unread](#) ]

Date: Wed, 2 Apr 2004 09:54:12 GMT

To: [stauseef@yahoo.com](mailto:stauseef@yahoo.com)

Subject: Acceptance of Paper

From: "Maria Gonzalez" <[m.gonzalez@dwj.com](mailto:m.gonzalez@dwj.com)> | [This is not spam](#) | [Add to Address Book](#)

Dear Mr S.Tauseef Ur Rehman  
Your Paper titled "K-MT: A Modified K-Mean Algorithm for Data Mining" has been accepted and will be published in September issue of DataWarehousing Journal. Five copies of the print will be sent to you free of charge, however you can order more copies by paying the \$8.00 per copy.

Once again, I congratulate you for the acceptance of letter.

Best Regards,

# *Research Paper*

# K-MT: A Modified K-Means Algorithm For Data Mining

M. Ali Mohsin, S.Tauseef-ur-Rehman  
[malimohsin@hotmail.com][stauseef@yahoo.com]  
Department of Computer Science  
Faculty of Applied Sciences  
International Islamic University, Islamabad

## Abstract

Partitioning a large set of objects into homogeneous clusters is a fundamental operation in data mining. The K-Means algorithm is best suited for implementing this operation because of its efficiency in clustering large data sets.

This paper presents the results of an experimental study of some common clustering techniques. In particular, we compare the two main approaches to clustering; K-Means and our proposed technique of clustering K-MT. K-Means and its variants have a time complexity which is linear in the number, but are thought to produce inferior clusters. However, our results indicate that the K-MT technique is better than the standard K-Means approach. We propose an explanation for these results that is based on an analysis of the specifics of the clustering algorithms. We also show that our approach K-MT is better than basic K-Means with help of an example.

Keywords: Clustering, K-Means , K-MT

## Introduction

Clustering, the purpose is to divide the records of a database in similar, homogeneous groups, but this time the user does not know the classes before the analysis. The clustering algorithm will have to discover the more natural way to group the records together, and then proceed with the grouping. Clustering involves dividing a set of data points into non-overlapping groups, or clusters, of points, where points in a cluster are "more similar" to one another than to points in other clusters. The term "more similar," when applied to clustered points, usually means closer by some measure of proximity. When a dataset is clustered, every point is assigned to some cluster, and every cluster can be characterized by a single reference point, usually an average of the points in the cluster. Any particular division of all points in a dataset into clusters is called a partitioning. Cluster analysis is the automatic identification of groups of similar objects. This analysis is achieved by maximizing inter-group similarity and minimizing intra-group similarity. Clustering is an unsupervised classification process that is fundamental to data mining. Many data mining queries are concerned either with how the data objects are grouped or which objects could be considered remote from natural groupings. There have been many works on cluster analysis, but we are now witnessing a significant resurgence of interest in new clustering techniques. Scalability and high dimensionality are not the only focus of the recent research in clustering analysis.



Indeed, it is getting difficult to keep track of all the new clustering strategies, their advantages and shortcomings.

### **K-Means**

K-Means is based on the idea that a center point can represent a cluster. In particular, for K-Means we use the notion of a centroid, which is the mean or median point of a group of points. Note that a centroid almost never corresponds to an actual data point.

The basic K-Means clustering technique is presented below. We elaborate on various issues in the following sections.

#### **Basic K-Means Algorithm for finding $K$ clusters**

1. Select  $K$  points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change.

Suppose that we are given a set of  $n$  data points  $X_1, X_2, \dots, X_n$  such that each data point is in  $R^d$ . The problem of finding the *minimum variance* clustering of this data set into  $k$  clusters is that of finding  $k$  points  $\{m_j\}_{j=1}^k$  in  $R^d$  such that:

$$\frac{1}{n} \sum_{i=1}^n \left( \min_j d^2(X_i, m_j) \right). \quad (1)$$

is minimized, where  $d(X_i, m_j)$  denotes the Euclidean distance between  $X_i$  and  $m_j$ . The points  $\{m_j\}_{j=1}^k$  are known as *cluster centroids* or as *cluster means*. Informally, the problem is that of finding  $k$  cluster centroids such that the average squared Euclidean distance (also known as the mean squared error or MSE, for short) between a data point and its nearest cluster centroid is minimized. The classical *K-Means* algorithm [Hartigan1975] provides an easy-to-implement approximate solution. Reasons for popularity of *K-Means* are ease of interpretation, simplicity of implementation, scalability, speed of convergence, adaptability to sparse data, and ease of out-of-core implementation [Zhang *et al.* 1996]. We present this algorithm in Figure 1, and intuitively explain it below:

1. **(Initialization)** Select a set of  $k$  starting points  $\{m_j\}_{j=1}^k$  in  $R^d$ . The selection may be done in a random manner. (line 5 in Figure 1).
2. **(Distance Calculation)** For each data point  $X_i$ ,  $1 \leq i \leq n$ , compute its Euclidean distance to each cluster centroid  $m_j$ ,  $1 \leq j \leq k$ , and then find the closest cluster centroid (lines 14-21 in Figure 1).

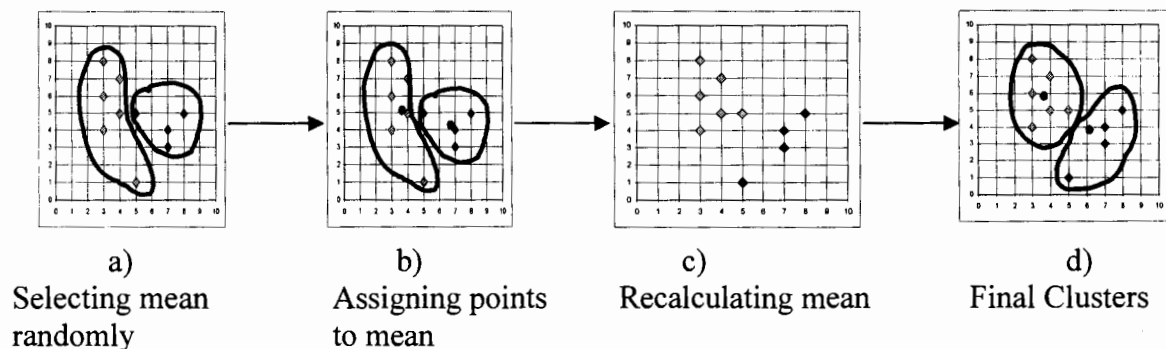
3. **(Centroid Recalculation)** For each  $1 \leq j \leq k$ , recompute cluster centroid  $m_j$  as the average of data points assigned to it (lines 22-26 in Figure 1).

4. **(Convergence Condition)** Repeat steps 2 and 3, until convergence (line 28 in Figure 1).

```

1:
2:
3: MSE = LargeNumber;
4:
5: Select  $k$  initial cluster centroids  $\{m_j\}_{j=1}^k$ ;
6:
7:
8: do {
9:   OldMSE = MSE;
10:  MSE' = 0;
11:  for  $j = 1$  to  $k$ 
12:     $m'_j = 0$ ;  $n'_j = 0$ ;
13:  endfor;
14:  for  $i = 1$  to  $n$ 
15:    for  $j = 1$  to  $k$ 
16:      compute squared Euclidean
        distance  $d^2(X_i, m_j)$ ;
17:    endfor;
18:    find the closest centroid  $m_\ell$  to  $X_i$ ;
19:     $m'_\ell = m'_\ell + X_i$ ;  $n'_\ell = n'_\ell + 1$ ;
20:     $MSE' = MSE' + d^2(X_i, m_\ell)$ ;
21:  endfor;
22:  for  $j = 1$  to  $k$ 
23:
24:
25:     $n_j = \max(n'_j, 1)$ ;  $m_j = m'_j / n_j$ ;
26:  endfor;
27:  MSE = MSE';
28: } while (MSE < OldMSE)
  
```

**Figure 1**



The above algorithm can be thought of as a gradient descent procedure which begins at the starting cluster centroids and iteratively updates these centroids to decrease the objective function. Furthermore, it is known that *K-Means* will always converge to a local minimum [Bottou and Bengio 1995]. The particular local minimum found depends on the

starting cluster centroids. As mentioned above, the problem of finding the global minimum is NP-complete. Before the above algorithm converges, steps 2 and 3 are executed a number of times, say I. The positive integer I is known as the *number of K-Means iterations*. The precise value of I can vary depending on the initial starting cluster centroids even on the same data set.

### K-MT

The K-MT is a variant of basic K-Means algorithm. In this approach we have used our own heuristics for calculating mean. We made one major modification to the K-Means algorithm i.e calculating the mean instead of assigning it randomly.

The main emphasize behind this approach is to reduce the size and time. The size is reduce as the number of iterations are minimized and the time is automatically reduced as less iterations are performed.

The K-MT algorithm is faster than the basic K-Means and thus extends the size of the datasets that can be clustered. It differs from the standard version in how the initial means are chosen.

This algorithm starts with a single cluster of all the points and works in the following way:

#### **Basic K-MT Algorithm for finding K clusters.**

- I. Consider all points as one big cluster.
- II. Calculate mean of this cluster. (KMT Step)
- III. Divide the cluster into two clusters such that :
  - a) A cluster consist of all the points less then mean
  - b) B cluster consist of all the points greater then mean.
- IV. Repeat step 3, for ITER times and take the split that produces the clustering with the highest overall similarity.
- V. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

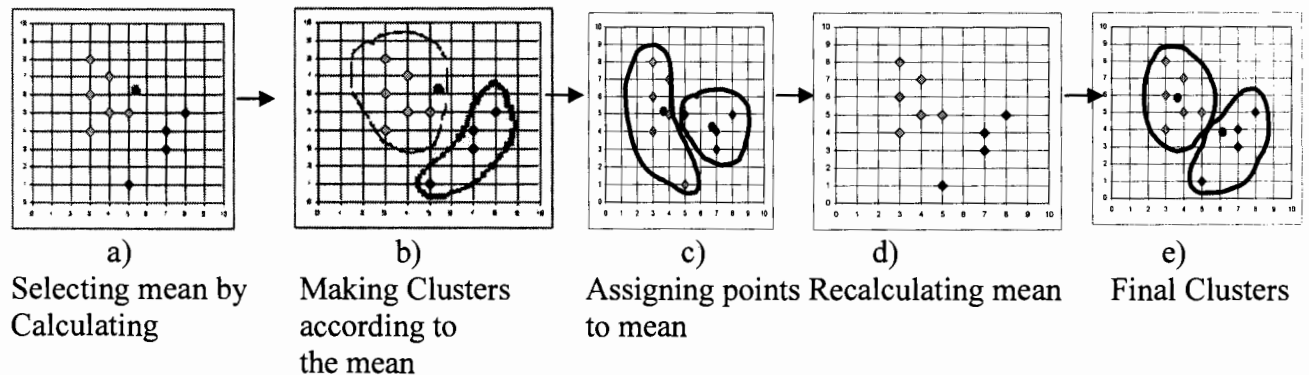
These modifications to the standard algorithm greatly accelerate the clustering process. The computer time can be further reduced by making the individual steps in the algorithm more efficient. A substantial fraction of the computation time required by any of these clustering algorithms is typically spent in finding the reference point closest to a particular data point.

Except for the first operation, the other three operations are repeatedly performed in the algorithm until the algorithm converges. The essence of the algorithm is to minimize the cost function

$$E = \sum_{l=1}^k \sum_{i=1}^n y_{i,l} d(X_i, Q_l) \quad (11)$$

where  $n$  is the number of objects in a data set  $X$ ,  $\bar{X}_l$  is the mean of cluster  $l$ , and  $y_{nxi}$  is an element of a *partition* matrix  $Y \times l \times n$  as in (Hand 1981).  $d$  is a dissimilarity measure usually defined by the squared Euclidean distance.

There is a question arises that if we are reducing the number of iteration and we are saying that it is efficient but we are putting an overhead here, that is of calculating mean by our self. This may be overhead but as long as it helps in reducing the size it is acceptable. Also one has to trade something to get efficient thing and in K-MT we are trading size with an extra calculation.



### Example

We can explain the difference between the K-Means and the K-MT by the help of this example in which there are randomly assigned number.

- Given: {2,4,10,12,3,20,30,11,25},  $k=2$   
We are given a random data set and we the initial  $k$  cluster is 2 .
- Randomly assign means:  $m_1=3, m_2=4$   
Randomly means are assigned to the  $k$  cluster.
- $K_1=\{2,3\}, K_2=\{4,10,12,20,30,11,25\}, m_1=2.5, m_2=16$
- $K_1=\{2,3,4\}, K_2=\{10,12,20,30,11,25\}, m_1=3, m_2=18$
- $K_1=\{2,3,4,10\}, K_2=\{12,20,30,11,25\}, m_1=4.75, m_2=19.6$
- $K_1=\{2,3,4,10,11,12\}, K_2=\{20,30,25\}, m_1=7, m_2=25$

Here we can see that after 4 number of iterations we have reached a point from further where we can not go. Now if we apply the same data set on K-MT and the K-MT approach is applied on it then we see what happens.

- Given: {2,4,10,12,3,20,30,11,25},  $k=2$   
We are given a random data set and we the initial  $k$  cluster is 2 .

- *Assign mean to it by calculating the mean and that comes to be  $m = 13$*
- Assign K1 less than mean i.e. all points less than  $m=13$  and K2 all the points greater than  $m=13$ .
- $K1=\{2,3,4,10,11,12\}, K2=\{20,30,25\}, m1=7, m2=25$

Hence we see that Straight away our number of iterations are reduced to 1 from 4. This shows that if we calculate the mean by our self and then assign the points, then the number of iterations are far less than basic K-Means iterations.

### **Conclusion**

“Data Mining is the process of extracting knowledge hidden from large volumes of raw data”. The importance of collecting data that reflect your business or scientific activities to achieve competitive advantage is widely recognized now. Powerful systems for collecting data and managing it in large databases are in place in all large and mid-range companies. However, the bottleneck of turning this data into your success is the difficulty of extracting knowledge about the system you study from the collected data. Data might be one of the most valuable assets of your corporation - but only if you know how to reveal valuable knowledge hidden in raw data. Data mining allows you to extract diamonds of knowledge from your historical data and predict outcomes of future situations. It will help you optimize your business decisions, increase the value of each customer and communication, and improve satisfaction of customer with your services.

Data mining can do:

- **Identify your best prospects and then retain them as customers.**  
By concentrating your marketing efforts only on your best prospects you will save time and money, thus increasing effectiveness of your marketing operation.
- **Predict cross-sell opportunities and make recommendations.**  
Whether you have a traditional or web-based operation, you can help your customers quickly locate products of interest to them - and simultaneously increase the value of each communication with your customers.
- **Learn parameters influencing trends in sales and margins.**  
You think you could do this with your OLAP tools? True, OLAP can help you prove a hypothesis - but only if you know what questions to ask in the first place. In the majority of cases you have no clue on what combination of parameters influences your operation. In these situations data mining is your only real option.
- **Segment markets and personalize communications.**  
There might be distinct groups of customers, patients, or natural phenomena that require different approaches in their handling. If you have a broad customer range, you would need to address teenagers in California and married homeowners in Minnesota with different products and messages in order to optimize your marketing campaign.

Cluster analysis is the automatic identification of groups of similar objects. This analysis is achieved by maximizing inter-group similarity and minimizing intra-group similarity. Clustering is an unsupervised classification process that is fundamental to data mining. Many data mining queries are concerned either with how the data objects are grouped or which objects could be considered remote from natural groupings. There have been many works on cluster analysis, but we are now witnessing a significant resurgence of interest in new clustering techniques. Scalability and high dimensionality are not the only focus of the recent research in clustering analysis. Indeed, it is getting difficult to keep track of all the new clustering strategies, their advantages and shortcomings.

### References

- [AGY99] Charu C. Aggarwal, Stephen C. Gates and Philip S. Yu, *On the merits of building categorization systems by supervised clustering*, Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 352 – 356, 1999.
- [APR97] Javed Aslam, Katya Pelekhov, and Daniela Rus, *A Practical Clustering Algorithm for Static and Dynamic Information Organization*, Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management, Bethesda, Maryland, USA, Pages 208-217, November 3-7, 1998.
- [BF98] Paul Bradley and Usama Fayyad, *Refining Initial Points for K-Means Clustering*, Proceedings of the Fifteenth International Conference on Machine Learning ICML98, Pages 91-99. Morgan Kaufmann, San Francisco, 1998.
- [BL85] Chris Buckley and Alan F. Lewit, *Optimizations of inverted vector searches*, SIGIR '85, Pages 97- 110, 1985.
- Hand, D. J. (1981) *Discrimination and Classification*, John Wiley & Sons.
- [CKPT92] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey, *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections*, SIGIR '92, Pages 318 – 329, 1992.
- [CCFW98] Moses Charikar, Chandra Chekuri, Tomas Feder, and Rajeev Motwani, *Incremental Clustering and Dynamic Information Retrieval*, STOC 1997, Pages 626-635, 1997.
- [DJ88] Richard C. Dubes and Anil K. Jain, *Algorithms for Clustering Data*, Prentice Hall, 1988.
- [EW89] A. El-Hamdouchi and P. Willet, *Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval*, The Computer Journal, Vol. 32, No. 3, 1989.

[GRS99] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, (1998), *ROCK: A Robust Clustering Algorithm for Categorical Attributes*, In Proceedings of the 15th International Conference on Data Engineering, 1999.

[Kow97] Gerald Kowalski, *Information Retrieval Systems – Theory and Implementation*, Kluwer Academic Publishers, 1997.

[KR90] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley and Sons, 1990.

[Rij79] C. J. van Rijsbergen, (1989), *Information Retrieval*, Butterworth, London, second edition.

[SS97] Hinrich Schutze and Craig Silverstein, *Projections for Efficient Document Clustering*, SIGIR '97, Philadelphia, PA, 1997.

[Sha48] Claude. E. Shannon, *A mathematical theory of communication*, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.

[ZEMK97] Oren Zamir, Oren Etzioni, Omid Madani, Richard M. Karp, *Fast and Intuitive Clustering of Web*

[Agrawal and Shafer1996] R. Agrawal and J. C. Shafer. Parallel mining of association rules: Design, implementation, and experience. *IEEE Trans. Knowledge and Data Eng.*, 8(6):962–969, 1996.

[Alsabti *et al.* 1998] K. Alsabti, S. Ranka, and V. Singh. An efficient *K-Means* clustering algorithm. In *Proceedings of IPDS/SPDP Workshop on High Performance Data Mining*, 1998.

Convergence properties of the *K-Means* algorithms. In G. Tesauro and D. Touretzky, editors, *Advances in Neural Information Processing Systems 7*, pages 585–592. The MIT Press, Cambridge, MA, 1995.

[Chattratchat *et al.* 1997] J. Chattratchat, J. Darlington, M. Ghanem, Y. Guo, H. H<sup>o</sup>uning, M. K<sup>o</sup>hler, J. Sutiwaraphun, H. W. To, and D. Yang. Large scale data mining: Challenges and responses.