Development of Benchmarking Standard for Document Clustering Algorithms based on Clustering Techniques for Data Mining

Developed By
Tehmina Amjad

Supervised By
Dr. Malik Sikander Hayat Khiyal

Department of Computer Science
Faculty of Applied Sciences
International Islamic University, Islamabad
(2004)

1. Data mining

10/07/10 ★

In the name of Allah,
The most Beneficent,
The most Merciful

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
MS in Computer Science

DEDICATION

Dedicated to my loving family especially my parents, whose affection has always been a source of encouragement for me, and whose prayers have always been the key to my success.

DECLARATION

I hereby declare that this neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this software and the accompanied report entirely on the basis of my personal efforts made under the guidance of my teachers and supervisor. 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.

Tehmina Amjad
95-CS-MS/02

DECLARATION

I hereby declare that this neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this software and the accompanied report entirely on the basis of my personal efforts made under the guidance of my teachers and supervisor. 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.

Tehmina Amjad
95-CS-MS/02

ACKNOWLEDGEMENT

All praise, acclamation and appreciation to Almighty ALLAH, the most merciful and compassionate, the most gracious and beneficent, whose bounteous blessings enabled me to complete this thesis.

I would like to express my cordial gratitude to **Dr. Malik Sikander Hayat Khiyal**, Head, Department of Computer Science, **Dr. S Tauseef ur Rehman** Head, Department of Telecommunication, and **Prof. Dr. Khalid Rashid,** Dean, Faculty of Applied sciences, International Islamic University Islamabad, who kept my moral high by their suggestions and appreciation. And last but not least I would like to acknowledge the support of my family and friends who helped me in every step of this thesis.

Tehmina Amjad
95-CS-MS/02

PROJECT IN BRIEF

Project Title:	C Miner, Development of Benchmarking Standard for Document Clustering Algorithms based on Clustering Techniques for Data Mining
Objective:	To compare different clustering algorithms under same endo and exo factors
Undertaken By:	Tehmina Amjad
Supervised By:	Dr. Malik Sikander Hayat Khiyal Head of Department Department of Computer Science, International Islamic University, Islamabad
Starting Date:	September 2003.
Completion Date:	June 2004.
Technologies Used:	Microsoft Visual C++ 6.0
Operating System	Windows 2000 Professional
System Used	Intel Pentium IV

ABSTRACT

Cluster analysis divides data into meaningful or useful groups (clusters). If meaningful clusters are the goal, then the resulting clusters should capture the "natural" structure of the data. For example, cluster analysis has been used to group related documents for browsing, to find genes and proteins that have similar functionality, and to provide a grouping of spatial locations prone to earthquakes. However, in other cases, cluster analysis is only a useful starting point for other purposes, e.g., data compression or efficiently finding the nearest neighbors of points. Whether for understanding or utility, cluster analysis has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining.

In this thesis focus is on specific algorithms of clustering technique and their review in a comparative manner. It describes a system designed to choose an appropriate algorithm of the Clustering technique, for a given dataset and its comparison with another algorithm using same data set. The design will provide a platform for such comparisons under same endo and exo factors.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1 INTRODUCTION

Knowledge discovery and data mining are the rapidly growing interdisciplinary fields 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. In this chapter the adopted definitions of these terms are;

> *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.

1.1 Data Mining

Databases today can range in size into the terabytes — more than 1,000,000,000,000 bytes of data. Within these masses of data lies hidden information of strategic importance. But when there are so many trees, how do you draw meaningful conclusions about the forest? The newest answer is data mining, which is being used both to increase revenues and to reduce costs. The potential returns are enormous. Innovative organizations worldwide are already using data mining to locate and appeal to higher-value customers, to reconfigure their product offerings to increase sales, and to minimize losses due to error or fraud. Data mining is a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions. The first and simplest analytical step in data mining is to describe the data — summarize its statistical attributes (such as means and standard deviations), visually review it using charts and graphs, and look for potentially meaningful links among variables (such as values that often occur together). As emphasized in the section on The Data Mining Process, collecting, exploring and selecting the right data are critically important.

But data description alone cannot provide an action plan. You must build a predictive model based on patterns determined from known results, and then test that model on results outside the original sample. A good model should never be confused with reality (you know a road map isn't a perfect representation of the actual road), but it can be a useful guide to understanding your business. The final step is to empirically verify the model. For example, from a database of customers who have already responded to a particular offer, you've built a model predicting which prospects are likeliest to respond to the same offer. Can you rely on this prediction? Send a mailing to a portion of the new list and see what results you get.

More precisely we can define Data mining as, *the extraction of hidden predictive information from large databases*, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to questions such as, "Which clients are most likely to respond to my next promotional mailing, and why?"

1.2 The Foundations of Data Mining

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time. Data mining takes this evolutionary process beyond retrospective data access and navigation to prospective and proactive information delivery. Data mining is ready for application in the business community because it is supported by three technologies that are now sufficiently mature:

- Massive data collection
- Powerful multiprocessor computers
- Data mining algorithms

Commercial databases are growing at unprecedented rates. A recent META Group survey of data warehouse projects found that 19% of respondents are beyond the 50 gigabyte level, while 59% expect to be there by second quarter of 1996. In some industries, such as retail, these numbers can be much larger. The accompanying need for improved computational engines can now be met in a cost-effective manner with parallel multiprocessor computer technology. Data mining algorithms embody techniques that have existed for at least 10 years, but have only recently been implemented as mature, reliable, understandable tools that consistently outperform older statistical methods.

In the evolution from business data to business information, each new step has built upon the previous one. For example, dynamic data access is critical for drill-through in data navigation applications, and the ability to store large databases is critical to data mining.

1.3 The Scope 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. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

1.3.1 Automated prediction of trends and behaviors.

Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

1.3.2 Automated discovery of previously unknown patterns.

Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

Data mining techniques can yield the benefits of automation on existing software and hardware platforms, and can be implemented on new systems as existing platforms are upgraded and new products developed. When data mining tools are implemented on high performance parallel processing systems, they can analyze massive databases in minutes.

Faster processing means that users can automatically experiment with more models to understand complex data. High speed makes it practical for users to analyze huge quantities of data. Larger databases, in turn, yield improved predictions.

Databases can be larger in both depth and breadth:

More columns. Analysts must often limit the number of variables they examine when doing hands-on analysis due to time constraints. Yet variables that are discarded because they seem unimportant may carry information about unknown patterns. High performance data mining allows users to explore the full depth of a database, without pre selecting a subset of variables.

More rows. Larger samples yield lower estimation errors and variance, and allow users to make inferences about small but important segments of a population.

1.4 How Data Mining Works

How exactly is data mining able to tell you important things that you didn't know or what is going to happen next? The technique that is used to perform these feats in data mining is called modeling. Modeling is simply the act of building a model in one situation where you know the answer and then applying it to another situation that you don't. For instance, if you were looking for a sunken Spanish galleon on the high seas the first thing you might do is to research the times when Spanish treasure had been found by others in the past. You might note that these ships often tend to be found off the coast of Bermuda and that there are certain characteristics to the ocean currents, and certain routes that have likely been taken by the ship's captains in that era. You note these similarities and build a model that includes the characteristics that are common to the locations of these sunken treasures. With these models in hand you sail off looking for treasure where your model indicates it most likely might be given a similar situation in the past. Hopefully, if you've got a good model, you find your treasure.

This act of model building is thus something that people have been doing for a long time, certainly before the advent of computers or data mining technology. What happens on computers, however, is not much different than the way people build models. Computers are loaded up with lots of information about a variety of situations where an answer is known and then the data mining software on the computer must run through that data and distill the characteristics of the data that should go into the model. Once the model is built it can then be used in similar situations where you don't know the answer. For example, say that you are the director of marketing for a telecommunications company and you'd like to acquire some new long distance phone customers. You could just randomly go out and mail coupons to the general population - just as you could randomly sail the seas looking for sunken treasure. In neither case would you achieve the results you desired and of course you have the opportunity to do much better than random - you could use your business experience stored in your database to build a model.

1.5 Data mining: What it can't do

Data mining is a tool, not a magic wand. It won't sit in your database watching what happens and send you e-mail to get your attention when it sees an interesting pattern. It doesn't eliminate the need to know your business, to understand your data, or to understand analytical methods. Data mining assists business analysts with finding patterns and relationships in the data — it does not tell you the value of the patterns to the organization. Furthermore, the patterns uncovered by data mining must be verified in the real world. Remember that the predictive relationships found via data mining are not necessarily *causes* of an action or behavior. For example, data mining might determine that males with incomes between $50,000 and $65,000 who subscribe to certain magazines are likely purchasers of a product you want to sell. While you can take advantage of this pattern, say by aiming your marketing at people who fit the pattern, you should not assume that any of these factors *cause* them to buy your product.

To ensure meaningful results, it's vital that you understand your data. The quality of your output will often be sensitive to outliers (data values that are very different from the typical values in your database), irrelevant columns or columns that vary together (such as age and date of birth), the way you encode your data, and the data you leave in and the data you exclude. Algorithms vary in their sensitivity to such data issues, but it is unwise to depend on a data mining product to make all the right decisions on its own.

Data mining will not automatically discover solutions without guidance. Rather than setting the vague goal, "Help improve the response to my direct mail solicitation," you might use data mining to find the characteristics of people who (1) respond to your solicitation, or (2) respond AND make a large purchase. The patterns data mining finds for those two goals may be very different.

Although a good data mining tool shelters you from the intricacies of statistical techniques, it requires you to understand the workings of the tools you choose and the algorithms on which they are based. The choices you make in setting up your data mining tool and the optimizations you choose will affect the accuracy and speed of your models.

Data mining does not replace skilled business analysts or managers, but rather gives them a powerful new tool to improve the job they are doing. Any company that knows its business and its customers is already aware of many important, high-payoff patterns that its employees have observed over the years. What data mining can do is confirm such empirical observations and find new, subtle patterns that yield steady incremental improvement (plus the occasional breakthrough insight).

1.6 Data mining and data warehousing

Frequently, the data to be mined is first extracted from an enterprise data warehouse into a data mining database or data mart (Figure 1.1). There is some real benefit if your data is already part of a data warehouse. As we shall see later on, the problems of cleansing data for a data warehouse and for data mining are very similar. If the data has already been cleansed for a data warehouse, then it most likely will not need further cleaning in order to be mined. Furthermore, you will have already addressed many of the problems of data consolidation and put in place maintenance procedures.

The data mining database may be a logical rather than a physical subset of your data warehouse, provided that the data warehouse DBMS can support the additional resource demands of data mining. If it cannot, then you will be better off with a separate data mining database.

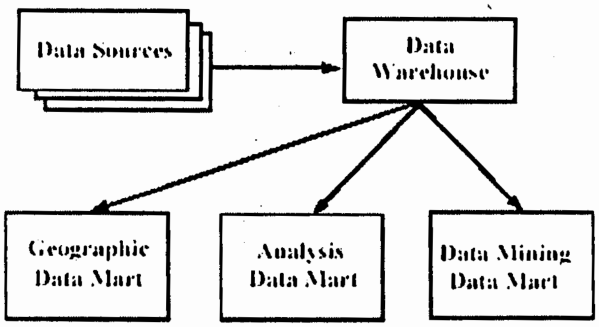


Figure 1.1: Data is extracted from a data warehouse to a data mining data mart

A data warehouse is not a requirement for data mining. Setting up a large data warehouse that consolidates data from multiple sources, resolves data integrity problems, and loads the data into a query database can be an enormous task, sometimes taking years and costing millions of dollars. You could, however, mine data from one or more operational or transactional databases by simply extracting it into a read-only database (Figure 1.2). This new database functions as a type of data mart.

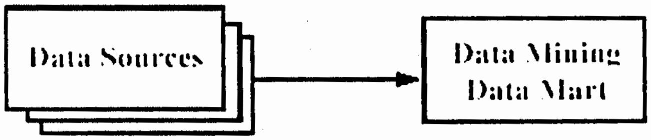


Figure 1.2: Data is extracted directly from data source to a data mining data mart

1.7 Data mining and OLAP

One of the most common questions from data processing professionals is about the difference between data mining and OLAP (On-Line Analytical Processing). As we shall see, they are very different tools that can complement each other.

OLAP is part of the spectrum of decision support tools. Traditional query and report tools describe *what* is in a database. OLAP goes further; it's used to answer *why* certain things are true. The user forms a hypothesis about a relationship and verifies it with a series of queries against the data. For example, an analyst might want to determine the factors that lead to loan defaults. He or she might initially hypothesize that people with low incomes are bad credit risks and analyze the database with OLAP to verify (or disprove) this assumption. If that hypothesis were not borne out by the data, the analyst might then look at high debt as the determinant of risk. If the data did not support this guess either, he or she might then try debt and income together as the best predictor of bad credit risks.

In other words, the OLAP analyst generates a series of hypothetical patterns and relationships and uses queries against the database to verify them or disprove them. OLAP analysis is essentially a deductive process. But what happens when the number of variables being analyzed is in the dozens or even hundreds? It becomes much more difficult and time-consuming to find a good hypothesis (let alone be confident that there is not a better explanation than the one found), and analyze the database with OLAP to verify or disprove it.

Data mining is different from OLAP because rather than verify hypothetical patterns, it uses the data itself to uncover such patterns. It is essentially an inductive process. For example, suppose the analyst who wanted to identify the risk factors for loan default were to use a data mining tool. The data mining tool might discover that people with high debt and low incomes were bad credit risks (as above), but it might go further and also discover a pattern the analyst did not think to try, such as that age is also a determinant of risk.

Here is where data mining and OLAP can complement each other. Before acting on the pattern, the analyst needs to know what the financial implications would be of using the discovered pattern to govern who gets credit. The OLAP tool can allow the analyst to answer those kinds of questions.

Furthermore, OLAP is also complementary in the early stages of the knowledge discovery process because it can help you explore your data, for instance by focusing attention on important variables, identifying exceptions, or finding interactions. This is important because the better you understand your data, the more effective the knowledge discovery process will be.

1.8 Data mining, machine learning and statistics

Data mining takes advantage of advances in the fields of artificial intelligence (AI) and statistics. Both disciplines have been working on problems of pattern recognition and classification. Both communities have made great contributions to the understanding and application of neural nets and decision trees.

Data mining does not replace traditional statistical techniques. Rather, it is an extension of statistical methods that is in part the result of a major change in the statistics community. The development of most statistical techniques was, until recently, based on elegant theory and analytical methods that worked quite well on the modest amounts of data being analyzed. The increased power of computers and their lower cost, coupled with the need to analyze enormous data sets with millions of rows, have allowed the development of new techniques based on a brute-force exploration of possible solutions.

New techniques include relatively recent algorithms like neural nets and decision trees, and new approaches to older algorithms such as discriminant analysis. By virtue of bringing to bear the increased computer power on the huge volumes of available data, these techniques can approximate almost any functional form or interaction on their own. Traditional statistical techniques rely on the modeler to specify the functional form and interactions.

The key point is that data mining is the application of these and other AI and statistical techniques to common business problems in a fashion that makes these techniques available to the skilled knowledge worker as well as the trained statistics professional. Data mining is a tool for increasing the productivity of people trying to build predictive models.

1.9 Data mining and hardware/software trends

A key enabler of data mining is the major progress in hardware price and performance. The dramatic 99% drop in the price of computer disk storage in just the last few years has radically changed the economics of collecting and storing massive amounts of data. At $10/megabyte, one terabyte of data costs $10,000,000 to store. At 10¢/megabyte, one terabyte of data costs only $100,000 to store! This doesn't even include the savings in real estate from greater storage capacities.

The drop in the cost of computer processing has been equally dramatic. Each generation of chips greatly increases the power of the CPU, while allowing further drops on the cost curve. This is also reflected in the price of RAM (random access memory), where the cost of a megabyte has dropped from hundreds of dollars to around a dollar in just a few years. PCs routinely have 64 megabytes or more of RAM, and workstations may have 256 megabytes or more, while servers with gigabytes of main memory are not a rarity.

While the power of the individual CPU has greatly increased, the real advances in scalability stem from parallel computer architectures. Virtually all servers today support multiple CPUs using symmetric multi-processing, and clusters of these SMP servers can be created that allow hundreds of CPUs to work on finding patterns in the data.

Advances in database management systems to take advantage of this hardware parallelism also benefit data mining. If you have a large or complex data mining problem requiring a great deal of access to an existing database, native DBMS access provides the best possible performance.

The result of these trends is that many of the performance barriers to finding patterns in large amounts of data are being eliminated.

1.10 An Architecture for Data Mining

To best apply these advanced techniques, they must be fully integrated with a data warehouse as well as flexible interactive business analysis tools. Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining. The resulting analytic data warehouse can be applied to improve business processes throughout the organization, in areas such as promotional campaign management, fraud detection, new product rollout, and so on. Figure 1 illustrates architecture for advanced analysis in a large data warehouse.

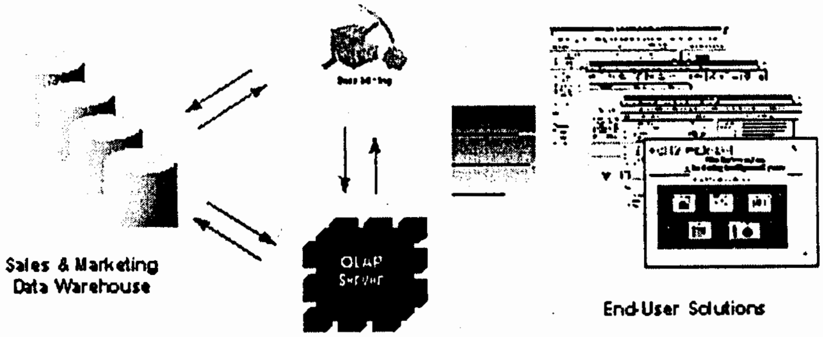


Figure 1.3 - Integrated Data Mining Architecture

The ideal starting point is a data warehouse containing a combination of internal data tracking all customer contact coupled with external market data about competitor activity. Background information on potential customers also provides an excellent basis for prospecting. This warehouse can be implemented in a variety of relational database systems: Sybase, Oracle, Redbrick, and so on, and should be optimized for flexible and fast data access.

An OLAP (On-Line Analytical Processing) server enables a more sophisticated end-user business model to be applied when navigating the data warehouse. The multidimensional structures allow the user to analyze the data as they want to view their business – summarizing by product line, region, and other key perspectives of their business. The Data Mining Server must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure. An advanced, process-centric metadata template defines the data mining objectives for specific business issues like campaign management, prospecting, and promotion optimization. Integration with the data warehouse enables operational decisions to be directly implemented and tracked. As the warehouse grows with new decisions and results, the organization can continually mine the best practices and apply them to future decisions.

This design represents a fundamental shift from conventional decision support systems. Rather than simply delivering data to the end user through query and reporting software, the Advanced Analysis Server applies users' business models directly to the warehouse and returns a proactive analysis of the most relevant information. These results enhance the metadata in the OLAP Server by providing a dynamic metadata layer that represents a distilled view of the data. Reporting, visualization, and other analysis tools can then be applied to plan future actions and confirm the impact of those plans.

Data mining is increasingly popular because of the substantial contribution it can make. It can be used to control costs as well as contribute to revenue increases.

1.11 Successful data mining

There are two keys to success in data mining. First is coming up with a precise formulation of the problem you are trying to solve. A focused statement usually results in the best payoff. The second key is using the right data. After choosing from the data available to you, or perhaps buying external data, you may need to transform and combine it in significant ways.

The more the model builder can "play" with the data, build models, evaluate results, and work with the data some more (in a given unit of time), the better the resulting model will be. Consequently, the degree to which a data mining tool supports this interactive data exploration is more important than the algorithms it uses.

Ideally, the data exploration tools (graphics/visualization, query/OLAP) are well-integrated with the analytics or algorithms that build the models.

1.12 Data mining applications

Many organizations are using data mining to help manage all phases of the customer life cycle, including acquiring new customers, increasing revenue from existing customers, and retaining good customers. By determining characteristics of good customers (profiling), a company can target prospects with similar characteristics. By profiling customers who have bought a particular product it can focus attention on similar customers who have not bought that product (cross-selling). By profiling customers who have left, a company can act to retain customers who are at risk for leaving (reducing churn or attrition), because it is usually far less expensive to retain a customer than acquire a new one.

Data mining offers value across a broad spectrum of industries. Telecommunications and credit card companies are two of the leaders in applying data mining to detect fraudulent use of their services. Insurance companies and stock exchanges are also interested in applying this technology to reduce fraud. Medical applications are another fruitful area: data mining can be used to predict the effectiveness of surgical procedures, medical tests or medications. Companies active in the financial markets use data mining to determine market and industry characteristics as well as to predict individual company and stock performance. Retailers are making more use of data mining to decide which products to stock in particular stores (and even how to place them within a store), as well as to assess the effectiveness of promotions and coupons. Pharmaceutical firms are mining large databases of chemical compounds and of genetic material to discover substances that might be candidates for development as agents for the treatments of disease.

1.12.1 Profitable Applications

A wide range of companies have deployed successful applications of data mining. While early adopters of this technology have tended to be in information-intensive industries such as financial services and direct mail marketing, the technology is applicable to any company looking to leverage a large data warehouse to better manage their customer relationships. Two critical factors for success with data mining are: a large, well-integrated data warehouse and a well-defined understanding of the business process within which data mining is to be applied (such as customer prospecting, retention, campaign management, and so on).

Some successful 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 via 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.

Each of these examples have a clear common ground. They leverage the knowledge about customers implicit in a data warehouse to reduce costs and improve the value of customer relationships. These organizations can now focus their efforts on the most important (profitable) customers and prospects, and design targeted marketing strategies to best reach them.

CHAPTER 2
LITERATURE REVIEW

2 LITERATURE REVIEW

By this point in time, you've probably heard a good deal about data mining -- the database industry's latest buzzword. What's this trend all about? To use a simple analogy, it's finding the proverbial needle in the haystack. In this case, the needle is that single piece of intelligence your business needs and the haystack is the large data warehouse you've built up over a long period of time.

2.1 Data Mining:

Through the use of automated statistical analysis (or "data mining") techniques, businesses are discovering new trends and patterns of behavior that previously went unnoticed. Once they've uncovered this vital intelligence, it can be used in a predictive manner for a variety of applications. Brian James, assistant coach of the Toronto Raptors, uses data mining techniques to rack and stack his team against the rest of the NBA. The Bank of Montreal's business intelligence and knowledge discovery program is used to gain insight into customer behavior. *CIO Magazine* provides a great executive overview of data mining for business-minded professionals.

The first step toward building a productive data mining program is, of course, to gather data! Most businesses already perform these data gathering tasks to some extent -- the key here is to locate the data critical to your business, refine it and prepare it for the data mining process. If you're currently tracking customer data in a modern DBMS, chances are you're almost done. Take a look at the article Mining Customer Data from *DB2 Magazine* for a great feature on preparing your data for the mining process.

The next step is to choose one or more data mining algorithms to apply to your problem. If you're just starting out, it's probably a good idea to experiment with several techniques to give yourself a feel for how they worked. Your choice of algorithm will depend upon the data you've gathered, the problem you're trying to solve and the computing tools you have available to you. Let's take a brief look at two of the more popular algorithms.

Regression is the oldest and most well-known statistical technique that the data mining community utilizes. Basically, regression takes a numerical dataset and develops a mathematical formula that fits the data. When you're ready to use the results to predict future behavior, you simply take your new data, plug it into the developed formula and you've got a prediction! The major limitation of this technique is that it only works well with continuous quantitative data (like weight, speed or age). If you're working with categorical data where order is not significant (like color, name or gender) you're better off choosing another technique. For a humorous look at regression, read Multiple Regression with Ren and Stimpy from New Mexico State University's Psychology Department.

Working with categorical data or a mixture of continuous numeric and categorical data? Classification analysis might suit your needs well. This technique is capable of processing a wider variety of data than regression and is growing in popularity. You'll also find output that is much easier to interpret. Instead of the complicated mathematical formula given by the regression technique you'll receive a decision tree that requires a series of binary decisions. Take a look at the Classification Trees chapter from the Electronic Statistics Textbook for in-depth coverage of this technique.

Regression and classification are two of the more popular classification techniques, but they only form the tip of the iceberg. For a detailed look at other data mining algorithms, look at this feature on Data Mining Techniques or the SPSS Data Mining page.

Data mining products are taking the industry by storm. The major database vendors have already taken steps to ensure that their platforms incorporate data mining techniques. Oracle's Data Mining Suite (Darwin) implements classification and regression trees, neural networks, k-nearest neighbors, regression analysis and clustering algorithms. Microsoft's SQL Server 2000 also offers data mining functionality through the use of classification trees and clustering algorithms. If you're already working in a statistics environment, you're probably familiar with the data mining algorithm implementations offered by the advanced statistical packages SPSS and S-Plus.

2.1.1 Few Valuable Papers on Clustering:

2.1.1.1 Knowledge Discovery in Databases Tools and Techniques

By Peggy Wright Crossroads. 1998. "The purpose of this paper is to present the results of a literature survey outlining the state-of-the-art in KDD techniques and tools. The paper is not intended to provide an in-depth introduction to each approach; rather, they intended it to acquaint the reader with some KDD approaches and potential uses."

2.1.1.2 Eureka! Knowledge Discovery

By Neena Buck. Software Magazine. December 2000/January 2001 cover story. "Knowledge discovery and data mining (KDD) is evolving from an esoteric art and a point solution, to a mainstream technology embedded in a variety of solutions, to help businesses turn information into insight."

2.1.1.3 Advanced Scout: Data Mining and Knowledge Discovery in NBA Data

A Brief Application Description by Inderpal Bhandari, et al. Data Mining and Knowledge Discovery 1, 121-125 (1997). Available from NEC ResearchIndex. "They describe Advanced Scout software from the perspective of data mining and knowledge discovery. This paper highlights the pre-processing of raw data that the program performs, describes the data mining aspects of the software and how the interpretation of patterns supports the process of knowledge discovery."

2.1.1.4 Data Mining Research: Opportunities and Challenges

A Report of three NSF Workshops on Mining Large, Massive, and Distributed Data. By Robert Grossman, Simon Kasif, Reagan Moore, David Rocke, and Jeff Ullman. January 1999. "Data mining is the semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data. That is, data mining attempts to extract knowledge from data. Data mining differs from traditional statistics in several ways:...." And be sure to read the impressive "Success Stories" in

Section 5! Made available by The National Center for Data Mining (NCDM) at the University of Illinois at Chicago (UIC).

2.1.1.5 Mining for trends at the help desk

By John Boyd. IBM Think Research (1999). "Ordinary data mining simply looks for keywords, but the text-mining system -- dubbed TAKMI (an abbreviation for Text Analysis and Knowledge Mining but also a Japanese word meaning 'skilled craftsman') -- spots grammatical relationships, as well. Knowing which word is the subject, which the verb, and which the object, TAKMI can categorize calls according to whether they are, say, complaints or questions and according to the product that is causing difficulty."

2.1.1.6 Knowledge Discovery & Data Mining Research at IBM.

The challenge of extracting knowledge from data draws upon research in statistics, databases, pattern recognition, machine learning, data visualization, optimization, and high-performance computing, to deliver advanced business intelligence and web discovery solutions

2.1.1.7 Financial Services data mining example: Identifying risky borrowers

From Salford Systems. "To introduce us to data mining with the CART decision tree software they have walked through a real world example drawn from the Financial Services industry. The database is an extract from a group of customers who selected a financial loan product, some of whom went 'BAD'. The information they used of comes from standard credit reports provided by all the major credit bureaus...."

2.1.1.8 Knowledge Discovery in Databases at Austrian Research Institute for Artificial Intelligence

In terms of research, the main emphasis of recent work has been on developing efficient and noise tolerant algorithms to discover the following types of knowledge: *Classification Rules* are still the most common type of knowledge that is inferred from databases. A classification rule predicts a dependent variable from other attributes in the database (e.g. Buyers of French cheese are usually working women with above average income.).... *Association Rules* represent dependencies between attributes in the database. A prototypical application area is the so-called basket analysis: Which items do customers often buy together in a supermarket?.... *Regression Rules* predict numeric values instead of symbolic values as common Machine Learning algorithms do (e.g. Beverage sales in June will be such and such.). For this purpose they have developed algorithms for the induction of first order regression trees...."

2.1.1.9 Data mining and Privacy in Public Sector using Intelligent Agents

By Max Voskob (max.voskob@paradise.net.nz) Release: 26-Nov-03. This paper discusses a technical solution for information sharing while addressing the privacy concerns with no need for reorganization of the existing public sector infrastructure . The solution is based on imposing an additional layer of Intelligent Software Agents and Knowledge Bases for data mining and analysis.

2.1.1.10 A Novel Neural Network for Data Mining

By: Tony Kai Yun Chan*, Eng Chong Tan and Neeraj Haralalka School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore. The algorithm proposed here for data mining deals with the standard Multilayer Perceptron using Temporal Back propagation algorithm with the concept of Bollinger Band Crossover from the concept of Trading Systems added to it, and is referred to as the Bollinger Band Crossover Supervised Network (BBCSN). To visualize the performances.of each of this algorithm, a portfolio management scheme was designed and translated into code using Visual C++. This enables a critical analysis of the algorithm, with respect to their financial performances. According to the results obtained in this project, the newly proposed and designed Bollinger Band Crossover gives much better results than generally obtained from a Multilayer Perceptron network.

2.1.1.11 Perfect Encoding: a Signature Method for Text Retrieval

By D. Dervos1; 2 P. Linardis1 Y. Manolopoulos1

A new methodology is introduced, where blocks of text are replaced by a compressed, fully reversible, signature pattern. Full reversibility implies zero information loss, thus the new method is termed Perfect Encoding. The method's analytical model is produced and, where applicable, contrasted with the current practice in signature file organizations. Analysis results indicate that it comprises a potential candidacy for information retrieval implementations. In particular, perfect encoding has the potential to develop into an alternative or complementary scheme to inverted or signature file based systems.

2.1.1.12 Research Issues in Web Data Mining

By Sanjay Madria, Sourav S Bhowmick, W. -K Ng, E. P. Lim

Center for Advanced Information Systems, School of Applied Science Nanyang Technological University, Singapore 639798. In this paper, they presented an overview of research issues in web mining. We discuss mining with respect to web data referred here as web data mining. In particular, our focus is on web data mining research in context of our web warehousing project called WHOWEDA (*W*are*ho*use of *We*b *Da*ta). We have categorized web data mining into threes areas; web content mining, web structure mining and web usage mining. We have highlighted and discussed various research issues involved in each of these web data mining category. We believe that web data mining will be the topic of exploratory research in near future.

2.1.13 Data Mining in Soft Computing Framework: A Survey

By: Sushmita Mitra, *Senior Member, IEEE*, Sankar K. Pal, *Fellow, IEEE*, and Pabitra Mitra

This article provides a survey of the available literature on data mining using soft computing. A categorization has been provided based on the different soft computing tools and their hybridizations used, the data mining function implemented, and the preference criterion selected by the model. The utility of the different soft computing methodologies is highlighted. Generally fuzzy sets are suitable for handling the issues related to understandability of patterns, incomplete/noisy data, mixed media information and human interaction, and can provide approximate solutions faster. Neural networks are nonparametric, robust, and exhibit good learning and generalization capabilities in data-rich environments. Genetic algorithms provide efficient search algorithms to select a model, from mixed media data, based on some preference criterion/objective function. Rough sets are suitable for handling different types of uncertainty in data. Some challenges to data mining and the application of soft computing methodologies are indicated.

2.2 Clustering:

Clustering is not a new term. The era of clustering began in 1983 and now a days it becomes more and more popular as a solution of different problems in computing networking environment. With Clustering Technology, multiple servers are connected to form a "cluster" of computers. Each computer in the "cluster" is connected to the Internet through a Load Balancing Router (LBR). When a request comes through to the LBR for information or services, the LBR checks to see which server is least busy and "routes" the request to that server. This provides optimal uptime and quick ease of use for our customers. Hence for Connecting two or more computers together in such a way that they behave like a single computer or need a parallel processing, for load balancing and for fault tolerance, they used clustering technology.

Clustering is a popular strategy for implementing parallel processing applications because it enables companies to leverage the investment already made in PCs and workstations. In addition, it's relatively easy to add new CPUs simply by adding a new PC to the network.

2.2.1 Innovation Clusters

There is evidence that the creation and growth of new, technology-based enterprises occurs most effectively in geographically-limited clusters.

A cluster, as defined by Michael Porter of the Harvard Business School, is a geographic concentration of competing and cooperating companies, suppliers, service providers, and associated institutions.

The most famous innovation cluster, of course, is Silicon Valley. In such clusters, there is a frequent and strong interaction between many individuals and organizations, on both formal and informal levels. Science parks are an attempt to create clusters of like-minded individuals and organizations - and to provide them with their basic infrastructure needs.

However, there is insufficient evidence to suggest that this 'artificial' building of clusters is truly effective. The 'real' clusters, like Silicon Valley grew up over significant periods of time and evolved into the structures they now are.

It is difficult to identify the precise building blocks necessary for an effective cluster but certainly many seem to bring together people and institutions involved in business development, finance (and especially venture capital), management, consulting, and research (many clusters are built around or near a University campus since this provides both research input and a source of new talent.

Clusters seem to have grown in particular parts of the world. This is often due to location-specific factors such as a history of clustering based around an available raw material, or an available transport infrastructure. Other parts of the world seem unable to establish effective clusters. For example, there are few in Latin America though there are areas with potential. There may be cultural barriers present here - or there may be 'maturity' issues, with the potential being realized as the area grows into being a cluster.

There is a belief abroad that in a networked world, the value of clustering is diminished - that physical co-location is less important. This seems to underplay the importance of the informal networking that goes on in clustered regions. Though it may be possible to simulate on the Web or via other technological processes a number of the processes involved in clustering, it is unlikely that (yet, at least) all such processes (which are only partially understood) can be simulated or re-created. Clustering is often now seen as a key means of driving regional development - of building private and public sector partnerships to mutual benefit - through government and regional investment in innovation incubators, science parks and cities, technology transfer offices, etc. It is seen that those regions around the world which have been able to achieve a clustering effect (whether by accident or design) do seem to be more able to achieve and sustain significant success in the global marketplace. Such regional agencies need to identify and understand the real success factors in building and maintaining an effective cluster - there is considerable research going on to identify such factors, but as yet, the jury is still out

2.2.2 HISTORY AND DEVELOPMENT OF CLUSTERING:

The era of clustering began in 1983, when Digital Equipment Corp. shipped VAXclusters to allow multiple VAX computers to share a common workload. A VAXcluster looked like a single computer to users, the network, developers, and even system administrators. Clustering technology for Unix servers became available in the late 1980s and early 1990s. Unfortunately, making Unix do everything VMS could do in a cluster would have meant rewriting much of the Unix operating system. With this in mind, Unix suppliers only attempted to do part of what Digital accomplished with its VMS. Unfortunately, for marketing purposes, these suppliers chose to use the term clustering to describe their software anyway. It's interesting to note that only now are Unix suppliers shipping software that can match what Digital did in the early 1980s.

Digital's VAXclusters (now OpenVMS) were the first clusters to market. In 1990, UNIX and NetWare clustering opened up this market, together capturing 50% of the sampled clusters installed between 1990 and 1992. Windows NT clustering appears in this sample in 1996, making great in-roads by 1997. Of the sampled clusters installed in 1997, over half are Windows NT-based. According to this sample, the interest in clustering is, in large part, driven by a desire to increase the availability of PC servers. As noted earlier, an increasing amount of critical data is being deployed on the distributed network, which was not designed to host mission critical applications. Clustering is allowing the MIS community to deploy business critical data on (relatively) inexpensive, user friendly PC servers, while still maintaining the levels of availability they required. Clustering has been around for a couple of decades and has been used in many shapes and forms by companies that run mission-critical applications and that can't afford to be without their all-important servers.

For years, Digital Equipment has been clustering VAX machines running OpenVMS. Several Unix players, including Digital, IBM, and Sun Microsystems, have offered clustered Unix systems for some time, too. And, Tandem has been well known for its Himalaya servers, which are based on massively parallel clusters of processors. (For more on the history of clustering and a closer look at this technology and standards Throughout its history, server clustering has shifted from mainframes to minicomputers to Unix- and Intel-based servers. Tandem (Cupertino, CA) was among the first, rolling out its Non Stop Himalaya servers nearly 20 years ago. During this era, Digital Equipment devised clustering of VAX systems running VMS. IBM also earned pioneer status, developing clustering hardware for its AIX and mainframe systems.

The term clustering has been given many different definitions. In this report , they referred to clustering as a technology in which two or more servers (or nodes) act as one, appearing to client applications and users as a single system. If the primary server goes down, failover to the secondary server ensures that the user or application continues to be serviced without interruption. As opposed to functioning solely as a backup, the secondary server is active, performing typical tasks until it's needed by the primary server.

2.2.3 Few Valuable Papers on Clustering:

2.2.3.1 Clustering Large Datasets in Arbitrary Metric Spaces

By Venkatesh Ganti Raghu Ramakrishnan Johannes Gehrke Computer Sciences Department, University of Wisconsin-Madison Allison Powell James French Department of Computer Science, University of Virginia, Charlottesville

They presented two scalable algorithms designed for clustering very large datasets in distance spaces. Our first algorithm BUBBLE is, to our knowledge, the first scalable clustering algorithm for data in a distance space. Our second algorithm BUBBLE-FM improves upon BUBBLE by reducing the number of calls to the distance function, which may be computationally very expensive. Both algorithms make only a single scan over the database while producing high clustering quality. In a detailed experimental evaluation, they studied both algorithms in terms of scalability and quality of clustering.

2.2.3.2 A Comparison of Document Clustering Techniques

By Michael Steinbach George Karypis Vipin Kumar Department of Computer Science and Egineering, University of Minnesota

This paper presents the results of an experimental study of some common document clustering techniques. In particular, they compared the two main approaches to document clustering, agglomerative hierarchical clustering and K-means. (For K-means they used a "standard" K-means algorithm and a variant of K-means, "bisecting" K-means.) Hierarchical clustering is often portrayed as the better quality clustering approach, but is limited because of its quadratic time complexity. In contrast, K-means and its variants have a time complexity which is linear in the number of documents, but are thought to produce inferior clusters. Sometimes K-means and agglomerative hierarchical approaches arc combined so as to "get the best of both worlds." Howcver, their results indicatc that the bisecting K-means technique is better than the standard K-means approach and as good or better than the hierarchical approaches that they tested for a variety of clustcr evaluation metrics. They propose an explanation for these results that is based on an analysis of the specifics of the clustering algorithms and the nature of document data.

2.2.3.3 On Clustering Validation Techniques

by MARIA HALKIDI, YANNIS BATISTAKIS, MICHALIS VAZIRGIANNIS
Department of Informatics, Athens University of Economics & Business, Athens, Greece (Hellas)

This paper introduces the fundamental concepts of clustering while it surveys the widely known clustering algorithms in a comparative way. Moreover, it addresses an important issuc of clustering process regarding the quality assessment of the clustering results. This is also related to the inherent features of the data set under concern. A review of clustering validity measures and approaches available in the literature is presented. Furthermorc, thc papcr illustrates the issucs that are under-addresscd by thc rcccnt algorithms and givcs thc trcnds in clustcring process.

2.2.3.4 Mining Very Large Databases to Support Knowledge Exploration

Exploitation of data mining may be increased by moving away from dependence on statistically trained experts, and by making data mining, both in terms of its application and its results, more easily deployed within the business. Full volume data mining avoids the constraints of sample-based approaches. A data-mining framework, enabling software applications to be developed with simplified interfaces, increases the usability of these techniques. Encapsulating resultant predictive models as components enables easy deployment of the results within a business. A workshop style environment is appropriate, where business managers need to work quickly with a powerful data mining capability.

2.2.3.5 Evaluation and Comparison of Clustering Algorithms in Analyzing ES Cell Gene Expression Data

by Gengxin Chen, Saied A. Jaradat, Nila Banerjee, Tetsuya S. Tanaka, Minoru S.H. Ko, Michael Q. Zhang

Many clustering algorithms have been used to analyze micro array gene expression data.

Given embryonic stem cell gene expression data, they applied several indices to evaluate the performance of clustering algorithms, including hierarchical clustering, k-means, PAM and SOM. The indices were homogeneity and separation scores, silhouette width, redundant score (based on redundant genes), and WADP (testing the robustness of clustering results after small perturbation). The results showed that the ES cell dataset posed a challenge for cluster analysis in that the clusters generated by different methods were only partially consistent. Using this data set, they were able to evaluate the advantages and weaknesses of algorithms with respect to both internal and external quality measures. This study may provide a guideline on how to select suitable clustering algorithms and it may help raise relevant issues in the extraction of meaningful biological information from micro array expression data.

2.2.3.6 Document Clustering for Distributed Full text Search

By Jinyang, Robert Morris MIT LCS, 200 Technology Square, Cambridge MA, 02139 USA

Recent research efforts in peer-to-peer (P2P) systems concentrate on providing a "distributed hash table"-like primitive in the P2P system (Stoica et al., 2001). However, to make P2P systems useful, they need to build a keyword search engine to index the entire document collection in the distributed system. Doing keyword search in a distributed environment poses new challenges for traditional information retrieval techniques...

2.2.3.7 Scaling Clustering Algorithms to Large Databases

P. S. Bradley, Usama Fayyad, and Cory Reina Microsoft Research, Redmond, WA 98052, USA.

The fundamental clustering problem is that of grouping together (clustering) similar data items and has many applications in analysis, compression, prediction, and visualization. Practical data clustering algorithms require multiple data scans to achieve convergence. For large databases, these scans become prohibitively expensive. We present a scalable clustering framework applicable to a wide class of iterative clustering algorithms for which at most one scan of the database is required. In this work, the framework is instantiated and numerically justified with the popular K-Means clustering algorithm. The method is based on identifying regions of the data that are compressible, regions that must be maintained in memory, and operates within the confines of a limited memory buffer. Empirical results demonstrate that the scalable scheme outperforms a sampling-based approach, which is the straightforward method for "scaling" existing traditional in-memory implementations of clustering algorithms to large-scale databases. In our scheme, data resolution is preserved to the extent possible based upon the size of the allocated memory buffer and the fit of current clustering model to the data. This framework is naturally extended to update multiple clustering models simultaneously. The framework is extensively evaluated on synthetic and publicly available data sets.

2.2.3.8 Evaluation of Hierarchical Clustering Algorithms for Document Datasets

by Ying Zhao and George Karypis Department of Computer Science, University of Minnesota, Minneapolis

Fast and high-quality document clustering algorithms play an important role in providing intuitive navigation and browsing mechanisms by organizing large amounts of information into a small number of meaningful clusters. In particular, hierarchical clustering solutions provide a view of the data at different levels of granularity, making them ideal for people to visualize and interactively explore large document collections.

The focus of this paper is to evaluate different hierarchical clustering algorithms and toward this goal they compared various partitional and agglomerative approaches. Our experimental evaluation showed that partitional algorithms always lead to better clustering solutions than agglomerative algorithms, which suggests that partitional clustering algorithms are well-suited for clustering large document datasets due to not only their relatively low computational requirements, but also comparable or even better clustering performance. We also present a new class of clustering algorithms called ***constrained agglomerative algorithms*** that combine the features of both partitional and agglomerative algorithms. Our experimental results showed that they consistently lead to better hierarchical solutions than agglomerative or partitional algorithms alone.

2.2.3.9 Details of the Clustering Algorithms Supplement to the paper "Validating Clustering in Gene Expression Data" (to appear in Bioinformatics)

By Ka Yee Yeung, David R. Haynor, Walter L. Ruzzo October 16, 2000

They implemented three partitional clustering algorithms: the *Cluster Affinity Search Technique* (CAST) [Ben-Dor et al. 1999], an *iterative* partition algorithm and the *k-means* algorithm [Jain and Dubes 1988]. Three hierarchical clustering algorithms were also implemented: single-link, average-link and complete-link. For comparison, they also implemented random clustering. K-means, iterative and random are randomized algorithms; the others are deterministic.

2.2.3.10 Comparison of clustering algorithms in speaker Identification

By Tomi Kinnunen, Teemu Kilpeläinen And Pasi Fränti Department of Computer Science, University of Joensuu, Finland.

In speaker identification, they matched a given (unkown) speaker to the set of known speakers in a database. The database is constructed from the speech samples of each known speaker. Feature vectors are extracted from the samples by short-term spectral analysis, and processed further by vector quantization for locating the clusters in the feature space. We study the role of the vector quantization in the speaker identification system. We compare the performance of different clustering algorithms, and the influence of the codebook size. We want to find out, which method provides the best clustering result, and whether the difference in quality contribute to improvement in recognition accuracy of the system.

CHAPTER 3
CLUSTERING

3 CLUSTERING

Cluster analysis divides data into meaningful or useful groups (clusters). If meaningful clusters are the goal, then the resulting clusters should capture the "natural" structure of the data. For example, cluster analysis has been used to group related documents for browsing, to find genes and proteins that have similar functionality, and to provide a grouping of spatial locations prone to earthquakes. However, in other cases, cluster analysis is only a useful starting point for other purposes, e.g., data compression or efficiently finding the nearest neighbors of points. Whether for understanding or utility, cluster analysis has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining.

3.1 The Benefits of Clustering

The simple definition of clustering is a group of loosely coupled systems that work collectively as a single system to provide fast, uninterrupted service, higher performance, or both. Clustering architecture provides the means for significantly increasing system scalability while achieving the highest possible levels of availability.

What is cluster?

1. Cluster is a group of servers or nodes.
2. Operate independently.
3. Work collectively as a single system
4. Provide uninterrupted computing service or higher performance or both.
5. Appears to clients as a single server.
6. Now entering mainstream computing.

3.2 EXPLANATION

3.2.1Why we need clustering?

- To fulfill the enterprise need for high availability

- For a scalability for future growth of enterprise

- High guarantee network and access availability

- Provide value added services such as shared and dedicated commerce site hosting.

3.2.2 Types of clustering solutions

There are different categories of clustering technology available to meet the requirements. Broadly, there are "software-based" clustering solutions and "hardware-based" clustering solutions. With software clustering, users must login again after the backup computer takes over. The current transaction each user was working on (if it has yet been saved) is then re-entered. With the top-of-the line "hardware" clustering solutions, the users would probably not even see this little interruption at all. Hardware clustering solutions typically require special proprietary hardware as well as special software to interconnect the clustered computers.

3.2.3 Objective of Clustering

The objective of clustering is essentially the same as redundancy i.e. continuous availability and transparent switchover from the primary system to the backup system in the event of a system outage. However, instead of redundant components installed in one or two systems, we have to set up the power of multiple systems. While this may have been cost-prohibitive in the past when few solutions were available, technology advancements and lower price points from a multitude of manufacturers have made clustering a much more cost-effective strategy. Major manufacturers such as Compaq, Digital, IBM, HP and Dell are all recognizing the increasing need for clustering solutions.

3.2.4 Benefits of Clustering

Scalability by allowing multiple standards-based servers to work together. The cluster combines the processing power of multiple servers to run a single cluster-enabled application (such as a parallel database server). Furthermore, processing power can be increased by adding servers to the cluster. Availability by allowing servers to "back each other up" in the case of failure. When a server within the cluster fails, another server (or servers) picks up the workload of the failed server. To the user, the applications and data running on the failed server remain available. Manageability by providing a "single system image" to the user of the cluster. The user sees the cluster as the provider of services and applications. The user does not know (or care) which server within the cluster is actually providing services

3.2.5 Operation of Clustering

The two Holy Grails of server clustering are availability and scalability. In a high-availability system, if one server in a cluster goes down, another takes over its functions. Clustering goes beyond systems such as Vinca's (Orem, UT) StandbyServer mirroring approach, where the backup system is nonoperational until it's required to take over for a primary server that's gone down. Most systems of this mirroring type currently support only two servers, although many vendors of fault-tolerant solutions say they plan to incorporate scalability into future offerings.

In server clustering, backup servers remain operational and perform typical day-to-day functions in addition to taking over for failed servers. Also, server clustering protects software as well as hardware systems. This is becoming an increasingly important factor as software problems account for a larger and larger proportion of failures and as hardware continues to become more reliable, says Marty Miller, a product line manager for NetFrame (Milpitas, CA). The company's ClusterData product is designed to reduce downtime associated with operating system failures.

Clustering's scalability benefits stem from the fact that you can add, as needed, components such as CPUs and storage to bump up the servers' capacity. Server clustering avoids the memory contention that's characteristic of approaches such as SMP.

A scalable clustering environment with effective middleware and load-balancing capabilities is a welcome sight in the distributed-computing world. The question of just how scalable server clusters can be depends on a number of factors. In a perfectly scalable system you could add a processor or a new storage subsystem and always achieve a directly proportional performance increase for each hardware component installed. For reasons that we will discuss later, the scalability equation isn't often that precise.

3.2.6 Effectiveness of Clustering

Because interest in decreasing downtime is driving the clustering market, the most important question to ask is, "Are clusters improving availability?" Judging by this study, the answer is a resounding "Yes". The average application availability before clusters were implemented was only 90.1%, as depicted in Figure 3. Over the course of a year, this percentage translates into 36 days of downtime, unacceptable for almost all applications. After clusters were implemented the average availability of these applications increased to 98.6%. Thirty-two percent (32%) of the clustered applications have an availability rating of over 99.99%.

Although these sites implemented clusters to increase availability, they were not confident that clustering would have this effect. During the installation of clusters on their networks, the respondents were very skeptical. However, clustering has proven its effectiveness. Many respondents expressed surprise that clusters work as promised. It is real world performance that will win over the MIS community, not marketing promises.

3.2.7 Clustering Architectures

To thoroughly comprehend the concept of clustering, it's important to understand the primary architectures through which servers can be linked to provide enhanced levels of functionality in the areas of fault tolerance, high availability, and scalability.

There are multiple trends driving the clustering movement. One is the shift away from the 80/20 rule, which states that 80 percent of network traffic remains local and 20 percent traverses the backbone. With more and more data traveling further and further away from its point of origin, server farms, or groups of clustered servers, linked to high-speed backbones are proving more adept at handling these revised traffic patterns.

Another factor in the rise of clustering is the Internet, with its hefty bandwidth and performance requirements. This development has generated a need for more powerful hardware and software solutions. Also, as offerings such as Web servers and multimedia products become more feature rich and complex, servers have to work at an increasingly brisk clip to keep up with the pace.

There are three general categories of traditional clustering architectures, based on how each server in the cluster accesses memory and disks, and whether servers share a copy of the operating system and the I/O subsystem. These three categories are:

1. Shared-memory. In the shared-memory model, all servers in the cluster use the same primary memory, through which all traffic to the cluster is routed through the shared memory. The systems in a shared-memory configuration also share a single copy of the operating system and the I/O subsystem. Symmetric Multiprocessing (SMP) systems fall into this category
2. Shared-disk. In the shared-disk model, each server has its own memory but the cluster shares common disks i.e share storage resources.. Since every server can concurrently access every disk, a distributed lock manager is required. As each node in a cluster has its own memory, but the nodes all share storage resources. These nodes are usually linked via a high-speed connection that transmits a

system "heartbeat" which helps to detect faults and to ensure rapid failover. The nodes in this type of cluster arrangement access storage through a system bus.

3. Shared-nothing. In the shared-nothing model, every server has its own memory and its own disks. Systems based on disk mirroring often use the shared-nothing model. In shared -nothing model servers share neither memory nor devices. Every processor in the cluster has its own memory, as well as a copy of the NOS. Traffic traverses through a dedicated high- peed bus. In shared-nothing systems, only one system can access a specific resource at one time, although if the server that owns that resource fails, another server can take over the resource.

While all of these architectures can deliver significant advantages, true clustering exists in the shared-disk and shared-nothing environments.

In addition, there is also the hybrid common-disk shared-nothing model, which uses a shared-nothing architecture on top of shared-disk hardware. Only one server at a time can access a disk, but if that server fails, another can provide uninterrupted service.

There are other common attributes that help define how a cluster operates.

1. In an active/active cluster, each server runs its own workload, and can assume responsibility for another cluster member in the event of a failure. Commonly, this functionality means that cluster servers are paired, although it may work more generally.
2. In a cluster that provides fail over/fail back, the workload of a failed server is automatically transferred to another server until the first server recovers, at which time its workload is automatically transferred back.
3. A cluster may use IP switchover to allow clients to find the replacement for a failed server with a minimum of service disruption. IP switchover causes a replacement server to change its IP address to match that of a failed server; it requires support for DHCP (Dynamic Host Configuration Protocol) and ARP (Address Resolution Protocol) to dynamically register an IP address change and

then to update the physical network address translation caches of other systems attached to the cluster subnet.

4. Like switchover, IP impersonation allows clients to find a replacement server. Instead of dynamically assigning a new IP address, however, IP impersonation reroutes network traffic intended for the failed server to the replacement server.

3.2.8 Uses of Clustering

Although the word cluster is applied to many different types of hardware configuration, clustering is typically used to support the following functions:

1. Data availability
2. Application availability
3. Performance
4. Scalability
5. Single management domain creation

A given cluster may support more than one function!

3.2.8.1 Data availability

This use of clustering "virtualizes" the storage of a group of systems. Files, databases, and even entire storage volumes remain available even when the original host becomes unavailable. This type of clustering is often used in client-centric computing architectures. The client application can continue to run even though its original fileserver or database server is offline. An order-entry system supporting mobile sales professionals is a good example of this type of application. The sales professionals must be able to send in their orders without being concerned with routine maintenance on their fileserver.

3.2.8.2 Application availability

Clustering for application availability virtualizes one or more applications. This type of clustering is built on clustering for data availability. With this type of clustering, even if your Microsoft Exchange or Lotus Notes server became unavailable, your e-mail services would continue to function. In most cases, server-centric applications become virtual through the use of this software.

3.2.8.3 Performance

Clustering for performance is a different beast altogether. In this case, the point of clustering is to improve the computational performance of a single application. The application is designed to run separate processes in parallel on separate systems. Not all applications lend themselves well to this type of clustering.

An example of this type of clustering is media content creation. The computing required to develop special effects for a film is spread over many computers. Since each frame or sequence that makes up a film is a separate entity, it's possible to render them on different systems if necessary. Once the images have been rendered, they are collected back together in the correct order to produce the desired frames or sequences. Movies such as Titanic, Antz, and others have been developed using this technology.

3.2.8.4 Scalability

Scalability is the ability to vary the capacity of a system incrementally over as broad of a range as possible. The goal of clustering for scalability is to increase the number of users who can access a given application, not to shorten the time it takes to complete an individual transaction. Instances of the same application are run on multiple systems. Whether or not data is shared among the application instances depends on the application and its architecture. Multiple instances of a manufacturing automation application probably would share the same database. If the task at hand was to publish static data using Web technology, however, it's likely that each system would maintain its own copy of the data.

3.2.8.5 Single management domain

Lowering the cost of administration is the goal of this type of clustering. Each of the systems in the cluster may support the computing workload of different departments or divisions of a single organization. These applications need not share data or offer more availability than today's systems do right out of the box. A single individual can manage all of the systems in the cluster from a single console. This can lead to considerable cost savings.

3.2.9 When clustering is indicated

Clustering should be examined as an option for applications or data that have one or more of the following characteristics:

1. Must be available a greater percentage of the time than is possible when supported on a single system
2. Must run faster than is possible on any single system and be able to be designed to run in parallel
3. Must support more users than can be supported on any single system and require a shared database
4. Lowest cost of administration is a key goal

3.3 Basic Concepts and Techniques of Cluster Analysis

3.3.1 What Cluster Analysis Is

Cluster analysis groups objects (observations, events) based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group should be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better the clustering. The definition of what constitutes a cluster is not well defined, and in many applications, clusters are not well recognition, machine learning, and statistics (discriminant analysis and decision analysis). While cluster analysis can be very useful, either directly or as a preliminary means of finding classes, there is more to data analysis than cluster analysis. For example, the decision of what features to use when representing objects is a key activity of fields such as data mining, statistics, and pattern recognition. Cluster analysis typically takes the features as given and proceeds from there. Thus, cluster analysis, while a useful tool in many areas, is normally only part of a solution to a larger problem that typically involves other steps and techniques.

3.3.3 Some Working Definitions of a Cluster

As mentioned above, the term, cluster, does not have a precise definition. However, several working definitions of a cluster are commonly used and are given below. There are two aspects of clustering that should be mentioned in conjunction with these definitions. First, clustering is sometimes viewed as finding only the most "tightly" connected points while discarding "background" or noise points. Second, it is sometimes acceptable to produce a set of clusters where a true cluster is broken into several subclusters (which are often combined later, by another technique). The key requirement in this latter situation is that the subclusters are relatively "pure," i.e., most points in a subcluster are from the same "true" cluster.

3.3.3.1 Well-Separated Cluster Definition

A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster. Sometimes a threshold is used to specify that all the points in a cluster must be sufficiently close (or similar) to one another.

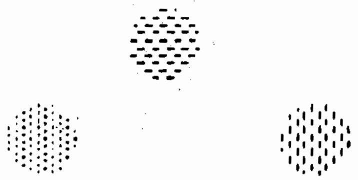


Figure 3.2: Three well-separated clusters of 2 dimensional points

However, in many sets of data, a point on the edge of a cluster may be closer (or more similar) to some objects in another cluster than to objects in its own cluster. Consequently, many clustering algorithms use the following criterion.

3.3.3.2 Center-based Cluster Definition

A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster. The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the "most representative" point of a cluster.

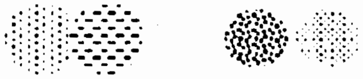


Figure 3.3: Four center-based clusters of 2 dimensional points

3.3.3.3 Contiguous Cluster Definition (Nearest Neighbor or Transitive Clustering)

A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

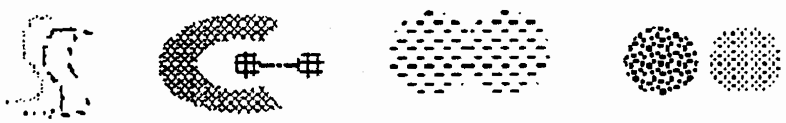


Figure 3.4: Eight contiguous clusters of 2 dimensional points

3.3.3.4 Density-based definition

A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density. This definition is more often used when the clusters are irregular or intertwined, and when noise and outliers are present. Notice that the contiguous definition would find only one cluster in Figure 6. Also note that the three curves don't form clusters since they fade into the noise, as does the bridge between the two small circular clusters.

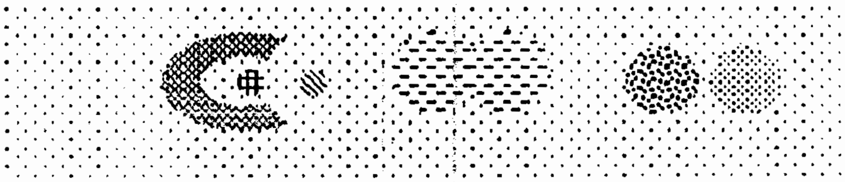


Figure 3.5: Six dense clusters of 2 dimensional points

3.3.3.5 Similarity-based Cluster definition

A cluster is a set of objects that are "similar", and objects in other clusters are not "similar." A variation on this is to define a cluster as a set of points that together create a region with a uniform local property, e.g., density or shape.

3.4 Selected Clustering Techniques

3.4.1 Cobweb

Cobweb is an incremental clustering algorithm, based on probabilistic categorization trees. The search for a good clustering is guided by a qualitative measure for partitions of data.

The algorithm reads **unclassified examples**, given in **attribute-value representation**. Pure COBWEB only supports nominal attributes, later extensions incorporate numerical values as well. An **incremental** clustering algorithm reads one example per iteration, modifying the actual result each time.

The aim is to achieve

1. High predictability of variable values, given a cluster.
2. High predictive-ness of a cluster, given variable values.

COBWEB's makes use of the **category (cluster) utility** and **partition utility** measures. Please follow the link above for details on these measures.

For each new example e COBWEB compares the following alternatives for the actual node (starting with the root node), as long as it is no leaf. The alternative with **highest partition utility** is chosen.

1. Insert e into the best successor node. Therefore e is inserted into every successor test wise, and the one where highest partition utility has been observed is chosen.
2. Create a new leaf for e and make it a successor of the actual node.
3. Generate a new node n, which is predecessor of the two best successors of the actual node (found in step 1) and insert n between the actual node and the two successors. Example e is inserted into n.
4. Choose the best successor node (found in step 1), delete it and make its successors direct successors of the actual node. Afterwards example e is inserted as in 1).

If a leaf has been reached, COBWEB creates two successors, one containing the new example, and one, containing the example of the old leaf. The old leaf becomes an inner node.

3.4.2 KMT

KMT is an algorithm that a modified version of the K Means algorithm. K Means is the basic algorithm that is used in portioning method of clustering. The KMT clustering algorithm classifies n points into k clusters by assigning a\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 not changes.

The variation in kmt which makes it different from kmeans is that in kmeans we select the initial mean or seed randomly but in kmt 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 kmean algorithm to the clusters. In this we also need to know the k number of clusters and n number of inputs.

3.4.2.1 KMT method

Given k, theKMT algorithm is implemented in five steps.

1. Consider all points as one big cluster
2. calculate mean of this cluster
3. divide the cluster into two clusters such that
    a) A cluster consists of all points less than mean
    b) B cluster consists of all points greater than mean
4. Repeat step three for ITER times and take the split that produces the clustering with the highest overall similarity.
5. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

3.4.3 k-Means

The K-means algorithm discovers K (non-overlapping) clusters by finding K centroids ("central" points) and then assigning each point to the cluster associated with its nearest centroid. (A cluster centroid is typically the mean or median of the points in its cluster and "nearness" is defined by a distance or similarity function.) Ideally the centroids are chosen to minimize the total "error," where the error for each point is given by a function that measures the discrepancy between a point and its cluster centroid, e.g., the squared distance. Note that a measure of cluster "goodness" is the error contributed by that cluster. For squared error and Euclidean distance, it can be shown [And73] that a gradient descent approach to minimizing the squared error yields the following basic Kmeans algorithm. (The previous discussion still holds if we use similarities instead of distances, but our optimization problem becomes a maximization problem.)

3.4.3.1 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 (or change very little).

K-means has a number of variations, depending on the method for selecting the initial centroids, the choice for the measure of similarity, and the way that the centroid is computed. The common practice, at least for Euclidean data, is to use the mean as the centroid and to select the initial centroids randomly.

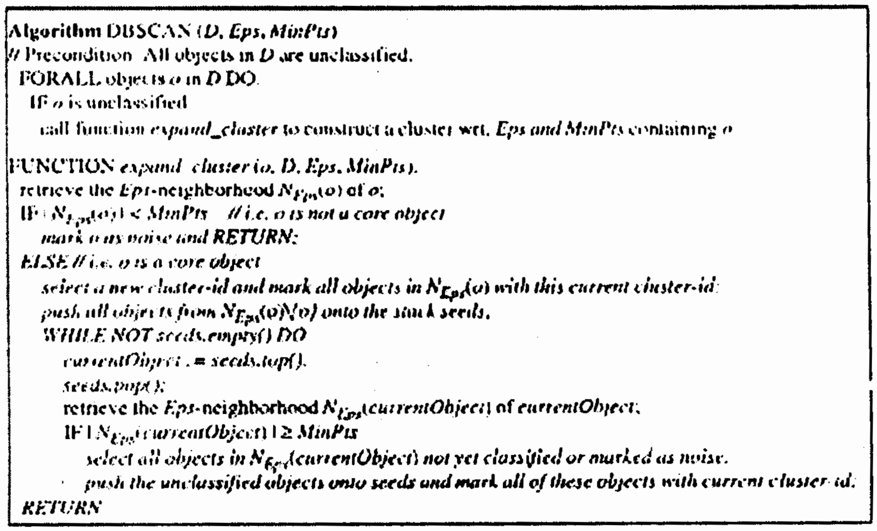
In the absence of numerical problems, this procedure converges to a solution, although the solution is typically a local minimum. Since only the vectors are stored, the space requirements are *O(m*n)*, where *m* is the number of points and *n* is the number of attributes. The time requirements are *O(I*K*m*n)*, where *I* is the number of iterations required for convergence. *I* is typically small and can be easily bounded as most changes occur in the first few iterations. Thus, the time required by K-means is efficient, as well as simple, as long as the number of clusters is significantly less than *m*.

Theoretically, the K-means clustering algorithm can be viewed either as a gradient descent approach which attempts to minimize the sum of the squared error of each point from cluster centroid or as procedure that results from trying to model the data as a mixture of Gaussian distributions with diagonal covariance matrices.

3.4.4 DBSCAN

DBSCAN is a clustering algorithm with two parameters, Eps and MinPts, utilizing the density notion that involves correlation between a data point and its neighbours .In order for data points to be grouped, there must be at least a minimum number of points calledMinP ts in Eps_neighbourhood, NEps(p), from a data point p, given a radius Eps. In DBSCAN, the density concept is introduced by the notations: Directly density-reachable, Density-reachable, and Density connected. These concepts define "Cluster" and "Noise".

The key idea of density-based clustering is that for each point of a cluster the neighborhood of a given radius (*Eps*) has to contain at least a minimum number of points (*MinPts*), i.e.the cardinality of the neighborhood has to exceed some threshold. We will first give a short introduction of DBSCAN including the definitions which are required for parallel clustering.



```
Algorithm DBSCAN (D, Eps, MinPts)
// Precondition: All objects in D are unclassified.
 FORALL objects o in D DO:
  IF o is unclassified
    call function expand_cluster to construct a cluster wrt. Eps and MinPts containing o

FUNCTION expand_cluster (o, D, Eps, MinPts):
 retrieve the Eps-neighborhood $N_{Eps}(o)$ of o;
 IF $|N_{Eps}(o)| < MinPts$   // i.e. o is not a core object
    mark o as noise and RETURN;
 ELSE // i.e. o is a core object
    select a new cluster-id and mark all objects in $N_{Eps}(o)$ with this current cluster-id;
    push all objects from $N_{Eps}(o)\backslash\{o\}$ onto the stack seeds;
    WHILE NOT seeds.empty() DO
      currentObject := seeds.top();
      seeds.pop();
      retrieve the Eps-neighborhood $N_{Eps}(currentObject)$ of currentObject;
      IF $|N_{Eps}(currentObject)| \geq MinPts$
        select all objects in $N_{Eps}(currentObject)$ not yet classified or marked as noise,
        push the unclassified objects onto seeds and mark all of these objects with current cluster-id;
 RETURN
```

Figure 3.1: The DBSCAN Algorithm

CHAPTER 4
ARCHITECTURE AND DESIGN

4 ARCHITECTURE AND DESIGN

4.1 Architecture

4.1.1Context Diagram

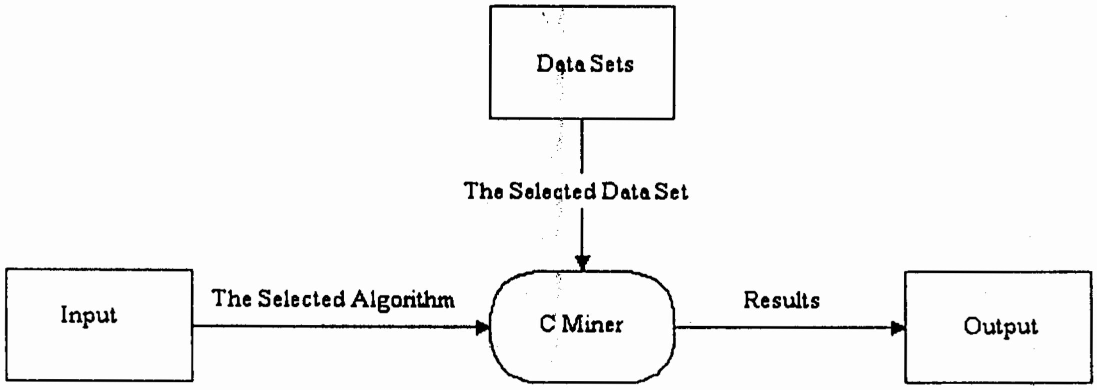


Figure 4.1: The context diagram

The user selects two different algorithms from the list of given algorithms. Then the user has to load the selected dataset. Both the selected algorithms will be applied to same data set and results will be displayed to user.

4.1.2 Module Diagram:

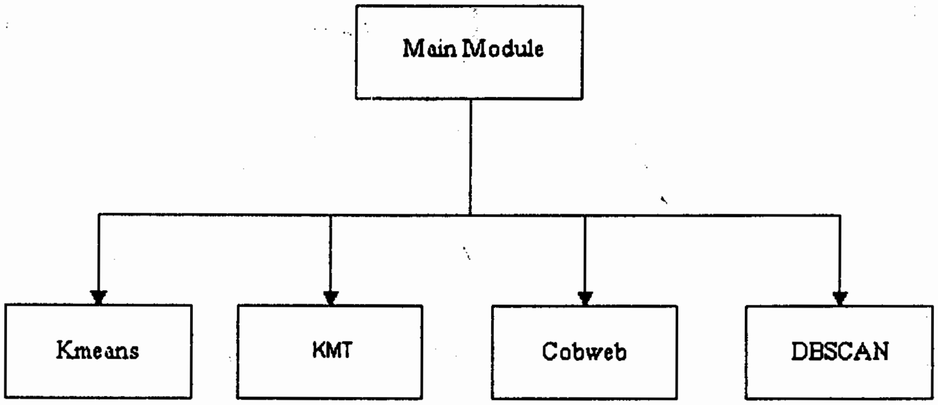


Figure 4.2: The Module Diagram

The main module consists of four different modules, each of which implements one clustering algorithm. Four clustering algorithms are implemented for purpose of comparison under same endo and exo factors. These are:

1. Kmeans
2. KMT
3. Cobweb
4. DBSCAN

4.2 Design

4.2.1 Class Diagram

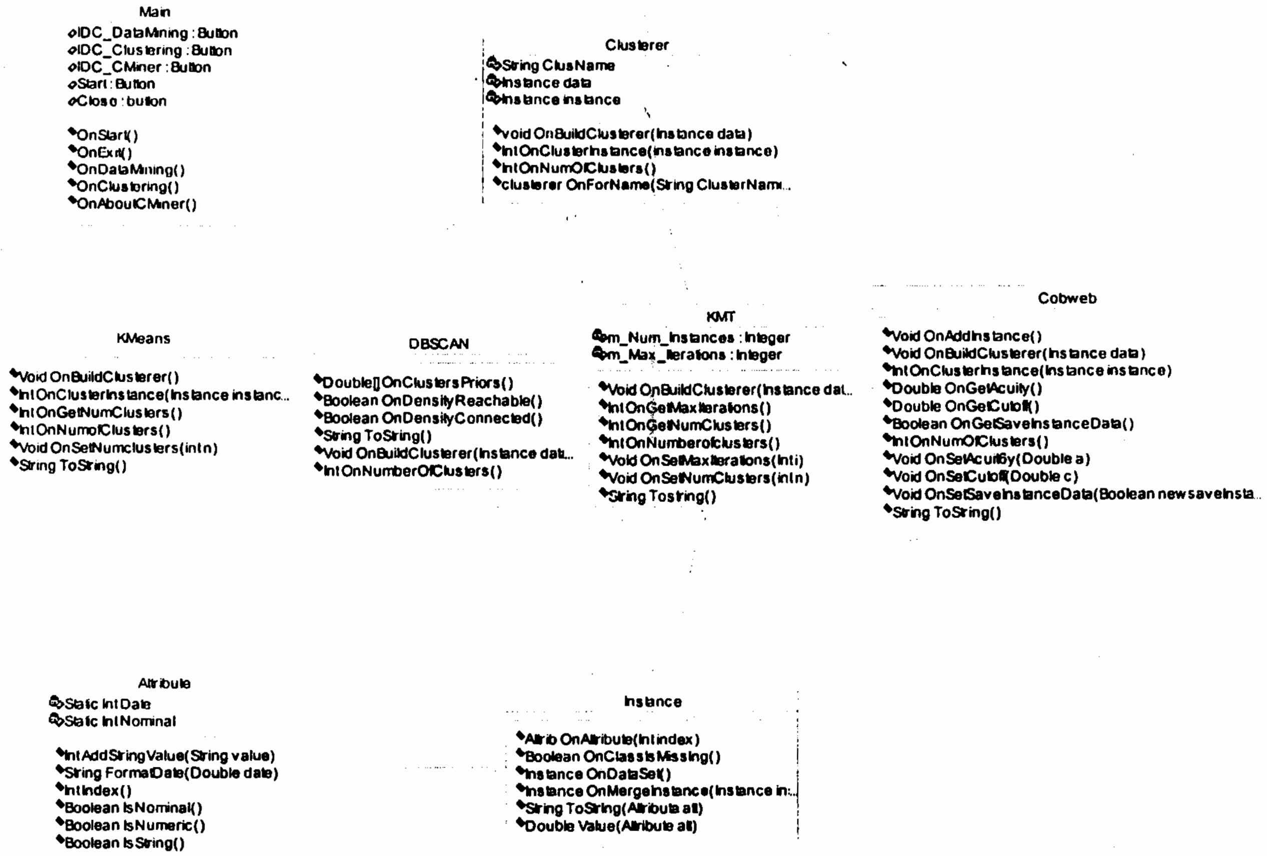

Main
IDC_DataMining : Button
IDC_Clustering : Button
IDC_CMiner : Button
Start : Button
Close : button
OnStart()
OnExit()
OnDataMining()
OnClustoring()
OnAboutCMiner()
Clusterer
String ClusName
Instance data
Instance instance
void OnBuildClusterer(Instance data)
IntOnClusterInstance(instance instance)
IntOnNumOfClusters()
clusterer OnForName(String ClusterNam...
KMeans
Void OnBuildClusterer()
IntOnClusterInstance(Instance instanc...
IntOnGetNumClusters()
IntOnNumofClusters()
Void OnSetNumclusters(intn)
String ToString()
DBSCAN
Double[] OnClustersPriors()
Boolean OnDensityReachable()
Boolean OnDensityConnected()
String ToString()
Void OnBuildClusterer(Instance dat...
IntOnNumberOfClusters()
KMT
m_Num_Instances : Integer
m_Max_Iterations : Integer
Void OnBuildClusterer(Instance dat...
IntOnGetMaxIterations()
IntOnGetNumClusters()
IntOnNumberofclusters()
Void OnSetMaxIterations(Inti)
Void OnSetNumClusters(intn)
String Tostring()
Cobweb
Void OnAddInstance()
Void OnBuildClusterer(Instance data)
IntOnClusterInstance(Instance instance)
Double OnGetAcuity()
Double OnGetCutoff()
Boolean OnGetSaveInstanceData()
IntOnNumOfClusters()
Void OnSetAcuit6y(Double a)
Void OnSetCutoff(Double c)
Void OnSetSaveInstanceData(Boolean newsaveInsta...
String ToString()
Attribute
Static IntDate
Static IntNominal
IntAddStringValue(String value)
String FormatDate(Double date)
IntIndex()
Boolean IsNominal()
Boolean IsNumeric()
Boolean IsString()
Instance
Attrib OnAttribute(Intindex)
Boolean OnClassIsMissing()
Instance OnDataSet()
Instance OnMergeInstance(Instance in...
String ToString(Attribute att)
Double Value(Attribute att)

Figure 4.3: Class Diagram

4.2.1.1 Class Main

OnStart

Starts the process of comparison

OnExit

Exits the software

OnDataMining

Leads to the introduction of data mining

OnClustering

Leads to the introduction of Clustering

OnAboutCMiner

Leads to the introduction of C Miner

4.2.1.2 Class Clusterer

OnBuildClusterer

public abstract void **OnbuildClusterer**(Instance data)
Generates a clusterer. Has to initialize all fields of the clusterer that are not being set via options.

Parameters:
data - set of instances serving as training data
Throws Exception
if the clusterer has not been generated successfully

OnClusterInstance

public int **OnClusterInstance**(Instance instance)

Classifies a given instance. Either this or distributionForInstance() needs to be implemented by subclasses.

Parameters:

instance - the instance to be assigned to a cluster

Returns:

the number of the assigned cluster as an integer

Throws Exception

if instance could not be clustered successfully

OnNnumberOfClusters

public abstract int **OnNumberOfClusters**()

Returns the number of clusters.

Returns:

The number of clusters generated for a training dataset.

Throws Exception:

If number of clusters could not be returned successfully

OnForName

public static Clusterer **OnForName**(String clustererName, String[] options)

Creates a new instance of a clusterer given it's class name and (optional) arguments to pass to it's setOptions method. If the clusterer implements OptionHandler and the options parameter is non-null, the clusterer will have it's options set.

Parameters:

options - an array of options suitable for passing to setOptions. May be null.

Returns:

the newly created search object, ready for use.

Throws Exception:

if the clusterer class name is invalid, or the options supplied are not acceptable to the clusterer.

4.2.1.3 Class Instance

public class **Instance**

Class for handling an instance. All values (numeric, nominal, or string) are internally stored as floating-point numbers. If an attribute is nominal (or a string), the stored value is the index of the corresponding nominal (or string) value in the attribute's definition. We have chosen this approach in favor of a more elegant object-oriented approach because it is much faster.

Typical usage (code from the main() method of this class):

```
...
// Create empty instance with three attribute values
Instance inst = new Instance(3);

// Set instance's values for the attributes "length", "weight", and
"position"
inst.setValue(length, 5.3);
inst.setValue(weight, 300);
inst.setValue(position, "first");

// Set instance's dataset to be the dataset "race"
inst.setDataset(race);

// Print the instance
System.out.println("The instance: " + inst);
...
```

All methods that change an instance are safe, ie. a change of an instance does not affect any other instances. All methods that change an instance's attribute values clone the attribute value vector before it is changed. If your application heavily modifies instance values, it may be faster to create a new instance from scratch.

OnAttribute

public Attribute **OnAttribute**(int index)

Parameters:

index - the attribute's index

Returns:

the attribute at the given position

Throws Exception:

UnassignedDatasetException - if instance doesn't have access to a dataset

OnClassIsMissing

public boolean **OnClassIsMissing**()

Tests if an instance's class is missing.

Returns:

true if the instance's class is missing

Throws Exception:

UnassignedClassException - if the class is not set or the instance doesn't have access to a dataset

OnDataset

public Instances **OnDataset**()

Returns the dataset this instance has access to. (ie. obtains information about attribute types from) Null if the instance doesn't have access to a dataset.

Returns:

the dataset the instance has accesss to

OnMergeInstance

```
public Instance OnMergeInstance(Instance inst)
```

Merges this instance with the given instance and returns the result. Dataset is set to null.

Parameters:

`inst` - the instance to be merged with this one

Returns:

the merged instances

OnToString

public final String **OnToString**(Attribute att)

Returns the description of one value of the instance as a string. If the instance doesn't have access to a dataset it returns the internal floating-point value. Quotes string values that contain whitespace characters, or if they are a question mark. The given attribute has to belong to a dataset.

Parameters:

att - the attribute

Returns:

the value's description as a string

OnValue

public double **value**(int attIndex)

Returns an instance's attribute value in internal format.

Parameters:

attIndex - the attribute's index

Returns:

the specified value as a double (If the corresponding attribute is nominal (or a string) then it returns the value's index as a double).

4.2.1.4 Class Attribute

public class **Attribute**

Class for handling an attribute. Once an attribute has been created, it can't be changed.

Three attribute types are supported:

- numeric:

  This type of attribute represents a floating-point number.

- nominal:

  This type of attribute represents a fixed set of nominal values.

- string:

  This type of attribute represents a dynamically expanding set of nominal values. String attributes are not used by the learning schemes in Weka. They can be used, for example, to store an identifier with each instance in a dataset.

Typical usage (code from the main() method of this class):

```
...
// Create numeric attributes "length" and "weight"
Attribute length = new Attribute("length");
Attribute weight = new Attribute("weight");

// Create vector to hold nominal values "first", "second", "third"
FastVector my_nominal_values = new FastVector(3);
my_nominal_values.addElement("first");
my_nominal_values.addElement("second");
my_nominal_values.addElement("third");

// Create nominal attribute "position"
Attribute position = new Attribute("position", my_nominal_values);
```

OnAddStringValue

public int **OnAddStringValue**(.String value)

Adds a string value to the list of valid strings for attributes of type STRING and returns the index of the string.

Parameters:

value - The string value to add

Returns:

the index assigned to the string, or -1 if the attribute is not of type Attribute.STRING

OnFormatDate

public String **OnFormatDate**(double date)

OnIsNominal

public final boolean **OnIsNominal**()

Test if the attribute is nominal.

Returns:

true if the attribute is nominal

OnIsNumeric

public final boolean **OnIsNumeric**()

Tests if the attribute is numeric.

Returns:

true if the attribute is numeric

OnIsString

public final boolean **OnIsString()**

Tests if the attribute is a string.

Returns:

true if the attribute is a string

OnIndex

public final int **OnIndex()**

Returns the index of this attribute.

Returns:

the index of this attribute

4.2.1.5 Class K-Means

OnBuildClusterer

public void **OnBuildClusterer**(Instances data)

throws java.lang.Exception

Generates a clusterer. Has to initialize all fields of the clusterer that are not being set via options.

Specified by:

buildClusterer in class Clusterer

Parameters:

data - set of instances serving as training data

Throws Exception:

if the clusterer has not been generated successfully

OnClusterInstance

public int **OnClusterInstance**(Instance instance)

Classifies a given instance.

Overrides:

clusterInstance in class Clusterer

Parameters:

instance - the instance to be assigned to a cluster

Returns:

the number of the assigned cluster as an interger if the class is enumerated,otherwise the predicted value

Throws Exception:

if instance could not be classified successfully

OnGetNumClusters

public int **OnGetNumClusters()**

gets the number of clusters to generate

Returns:

the number of clusters to generate

OnNumberOfClusters

public int **OnNumberOfClusters()**

Returns the number of clusters.

Specified by:

numberOfClusters in class Clusterer

Returns:

the number of clusters generated for a training dataset.

Throws Exception:

if number of clusters could not be returned successfully

4.2.1.6 Class Cobweb

public class **Cobweb**
extends Clusterer

Note: the application of node operators (merging, splitting etc.) in terms of ordering and priority differs (and is somewhat ambiguous) between the original Cobweb and Classit papers. This algorithm always compares the best host, adding a new leaf, merging the two best hosts, and splitting the best host when considering where to place a new instance.

OnAddInstance

public void **OnAddInstance**(Instance newInstance)
Adds an instance to the Cobweb tree.
Parameters:
newInstance - the instance to be added
Throws Exception:
if something goes wrong

OnBuildClusterer

public void **OnBuildClusterer**(Instances data)
Builds the clusterer.
Specified by:
buildClusterer in class Clusterer
Parameters:
data - the training instances.
Throws Exception:
if something goes wrong.

clusterInstance

public int **clusterInstance**(Instance instance)

Classifies a given instance.

Overrides:

clusterInstance in class Clusterer

Parameters:

instance - the instance to be assigned to a cluster

Returns:

the number of the assigned cluster as an interger if the class is enumerated, otherwise the predicted value

Throws Excception:

if instance could not be classified successfully

OnGetAcuity

public double **OnGetAcuity()**

get the acuity value

Returns:

the acuity

OnGetCutoff

public double **OnGetCutoff()**

Returns:

the cutoff

OnGetSaveInstanceData

public boolean **OnGetSaveInstanceData()**

Get the value of saveInstances.

Returns:

Value of saveInstances.

OnNumberOfClusters

public int **OnNumberOfClusters**()

Specified by:

numberOfClusters in class Clusterer

Returns:

the number of clusters generated for a training dataset.

Throws Exception:

if something goes wrong.

OnSetAcuity

public void **OnSetAcuity**(double a)

set the acuity.

Parameters:

a - the acuity value

OnSetCutoff

public void **OnSetCutoff**(double c)

Parameters:

c - the cutof

OnSetSaveInstanceData

public void **OnSetSaveInstanceData**(boolean newsaveInstances)

Set the value of saveInstances.

Parameters:

newsaveInstances - Value to assign to saveInstances.

OnToString

Public String **OnToString()**

Returns a description of the clusterer as a string.

Returns:

a string describing the clusterer.

4.2.1.7 Class DBSCAN

public class **DBSCAN**

Class for wrapping a Clusterer to make it return a distribution and density. Fits normal distributions and discrete distributions within each cluster produced by the wrapped clusterer.

OnBuildClusterer

public void **OnBuildClusterer(**Instances data)

Builds a clusterer for a set of instances.

Specified by:

buildClusterer in class Clusterer

Parameters:

`data` - set of instances serving as training data

Throws Exception:

if the clusterer hasn't been set or something goes wrong

OnClusterPriors

public double[] **OnClusterPriors()**

Returns the cluster priors.

Returns:

the prior probability for each cluster

OnNumberOfClusters

public int **numberOfClusters()**

Returns the number of clusters.

Returns:

the number of clusters generated for a training dataset.

Throws Exception:

if number of clusters could not be returned successfully

OnToString

public java.lang.String **toString()**

Returns a description of the clusterer.

Returns:

a string containing a description of the clusterer

4.2.1.8 Class KMT

OnBuildClusterer

public void **buildClusterer**(Instances data)

Generates a clusterer. Has to initialize all fields of the clusterer that are not being set via options.

Parameters:

`data` - set of instances serving as training data

Throws Exception:

if the clusterer has not been generated successfully

OnClusterPriors

public double[] **clusterPriors()**

Returns the cluster priors.

Returns:

the prior probability for each cluster

OnGetMaxIterations

public int **OnGetMaxIterations()**

Get the maximum number of iterations

Returns:

the number of iterations

OnGetMinStdDev

public double **OnGetMinStdDev()**

Get the minimum allowable standard deviation.

Returns:

the minumum allowable standard deviation

OnGetNumClusters

public int **getNumClusters()**

Get the number of clusters

Returns:

the number of clusters.

OnNumberOfClusters

public int **OnNumberOfClusters()**

Returns:

the number of clusters generated for a training dataset.

Throws Exception

if number of clusters could not be returned successfully

OnSetMaxIterations

public void **OnSetMaxIterations**(int i)

Set the maximum number of iterations to perform

Parameters:

i - the number of iterations

Throws Exception:

if i is less than 1

OnSetNumClusters

public void **OnSetNumClusters**(int n)

Set the number of clusters (-1 to select by CV).

Parameters:

n - the number of clusters

Throws Exception:

if n is 0

OnSetMinStdDev

public void **OnSetMinStdDev**(double m)

Set the minimum value for standard deviation when calculating normal density. Reducing this value can help prevent arithmetic overflow resulting from multiplying large densities (arising from small standard deviations) when there are many singleton or near singleton values.

Parameters:

m - minimum value for standard deviation

OnToString

Public String **OnToString()**

Outputs the generated clusters into a string.

4.2.2 Sequence Diagram

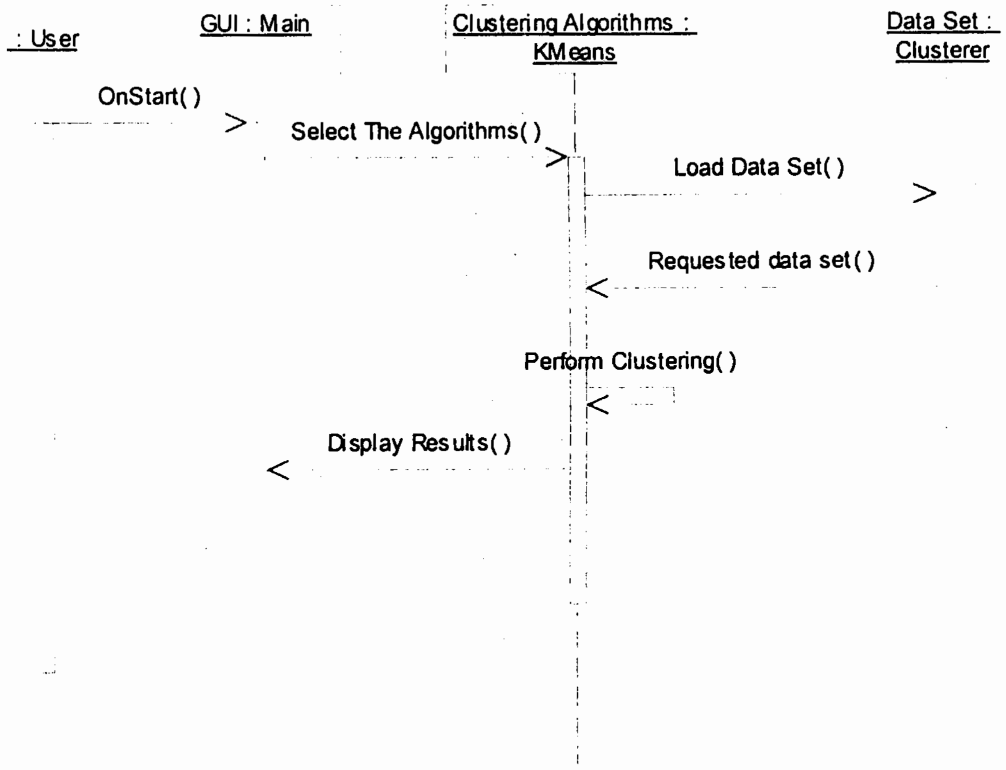


Figure 4.4: Sequence diagram

CHAPTER 5
RESULTS AND DISCUSSION

5 RESULTS AND DISCUSSION

5.1 General Criterions

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. The following are the typical requirements for a good clustering technique in data mining [10]:

1. Scalability: The cluster method should be applicable to huge databases and performance should decrease linearly with data size increase.
2. Versatility: Clustering objects could be of different types – numerical data, Boolean data or categorical data. Ideally a clustering method should be suitable for all different types of data objects.
3. Ability to discover clusters with different shapes: This is an important requirement for spatial data clustering. Many clustering algorithms can only discover clusters with spherical shapes.
4. Minimal input parameter: The method should require a minimum amount of domain knowledge for correct clustering. However, most current clustering algorithms have several key parameters and they are thus not practical for use in real world applications.

5. Robust with regard to noise: This is important because noise exists everywhere in practical problems. A good clustering algorithm should be able to perform successfully even in the presence of a great deal of noise.

5.2 The Data Sets Used

5.2.1 Contact Lens

Database for fitting contact lenses. This database is completing (all possible combinations of attribute-value pairs are represented). Each instance is complete and correct. The examples are complete and noise free. The examples highly simplified the problem.

5.2.1.1 Sources:

- Cendrowska, J. "PRISM: An algorithm for inducing modular rules",
- International Journal of Man-Machine Studies, 1987, 27, 349-370
- Donor: Benoit Julien (Julien@ce.cmu.edu)
- Date: 1 August 1990

5.2.1.2 Past Usage:

Witten, I. H. & MacDonald, B. A. (1988). Using concept learning for knowledge acquisition. International Journal of Man-Machine Studies, 27, (pp. 349-370).

5.2.2 CPU

As used by Kilpatrick, D. & Cameron-Jones, M. (1998). Numeric prediction using instance-based learning with encoding length selection. In Progress in Connectionist-Based Information Systems. Singapore: Springer-Verlag.

5.2.3 Iris

Iris Plants Database, This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

5.2.3.1 Sources

(a) Creator: R.A. Fisher

(b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)

5.2.3.2 Past Usage

- Publications: too many to mention!!! Here are a few.

(a) Fisher,R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

(b) Duda,R.O., & Hart,P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1. See page 218.

(c) Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine

(d) Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions on Information Theory, May 1972, 431-433.

5.3 COBWEB

COBWEB is an incremental clustering algorithm, based on probabilistic categorization trees. The search for a good clustering is guided by a qualitative measure for partitions of data. The algorithm reads **unclassified examples**, given in **attribute-value representation**. Pure COBWEB only supports nominal attributes; later extensions incorporate numerical values as well. That's why Cobweb performs well on the Contact Lens and Iris data cell but the performance decreases on CPU data set which shows that its scalability and versatility is affected.

5.4 K MEANS

K- Means is a least squares partitioning method allowing users to divide a collection of objects into k groups. During iterations it tries to minimize the distances of the objects to the respective group centroids. With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small). K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular. Since k-means is one of the so-called NP-Hard problems, so it can't guarantee that the absolute minimum of objective function has been reached. However it is scalable and has minimal input parameter. But the problem is that different initial partitions can result in different final clusters. In the experiments that I performed k-means worked well for the Contact Lens data set but its performance is reduced for CPU and Iris data set.

5.5 DBSCAN

The key idea of density-based clustering is that for each point of a cluster the neighborhood of a given radius (Eps) has to contain at least a minimum number of points (MinPts), i.e. the cardinality of the neighborhood has to exceed some threshold. It is significantly more affective in discovering clusters of the arbitrary shapes. The requirements of domain knowledge to determine input parameter are minimal. It is also affective for large databases.

5.6 KMT

KMT reduces the size and time. The size is reduced as the number of iterations is minimized and time is automatically reduced as the number of iterations is reduced. So it outperforms k-means and thus extends the size of the size of the data sets that can be clustered. It differs from the initial version in how the initial means are chosen. As a result is shows efficient result on all the three data sets.

5.7 Results

Different results are obtained when C Miner is executed on different data sets using different algorithms. The following table shows the time taken by each algorithm when executed on different data sets.

Data Set	Algorithm	Time taken
Contact Lens	Cobweb	00:00:03
	DBSCAN	00:00:04
	K- Means	00:00:06
	KMT	00:00:02
CPU	Cobweb	00:00:12
	DBSCAN	00:00:03
	K- Means	00:00:40
	KMT	00:00:03
IRIS	Cobweb	00:00:03
	DBSCAN	00:00:02
	K- Means	00:00:20
	KMT	00:00:02

CHAPTER 6
APPENDIX

6 Appendix

The main interface of C Miner is quite simple and easily understandable. It holds three buttons that lead to the introduction of Data Mining, Clustering and C Miner itself. The **Start** button leads towards the actual process. The **Close** button will close the C Miner. The following figure shows the main interface of C Miner.

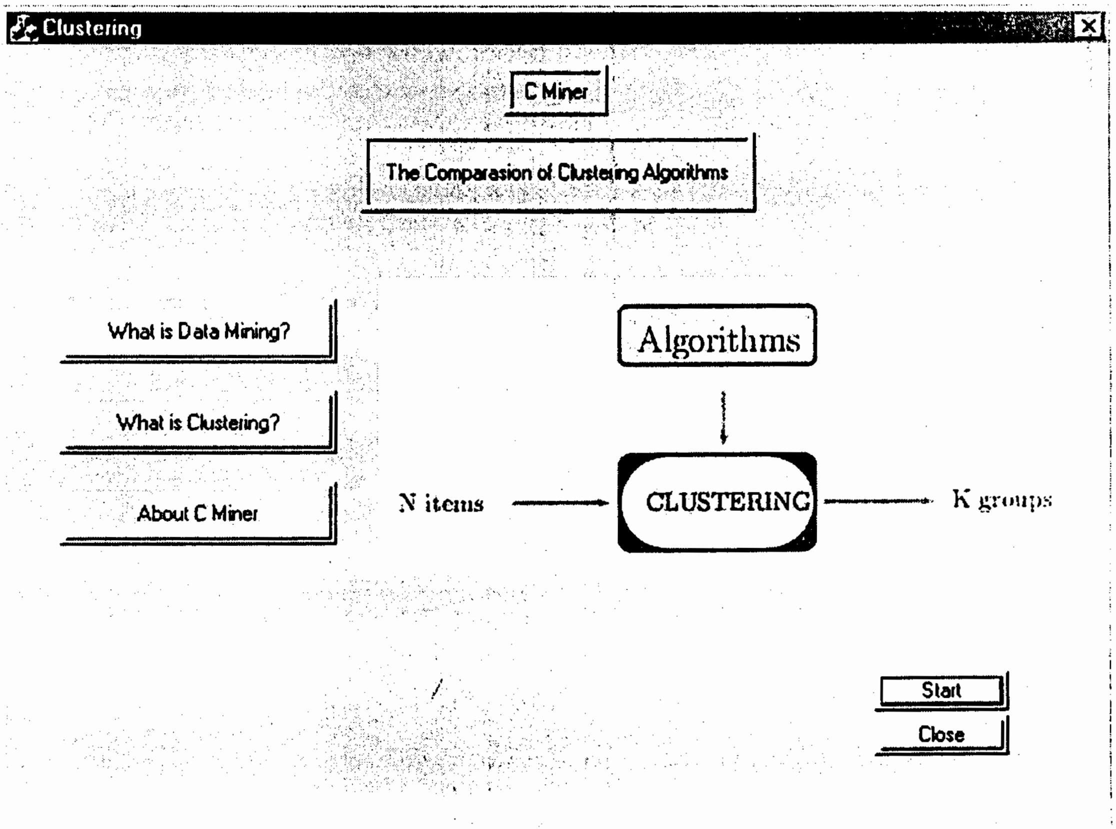
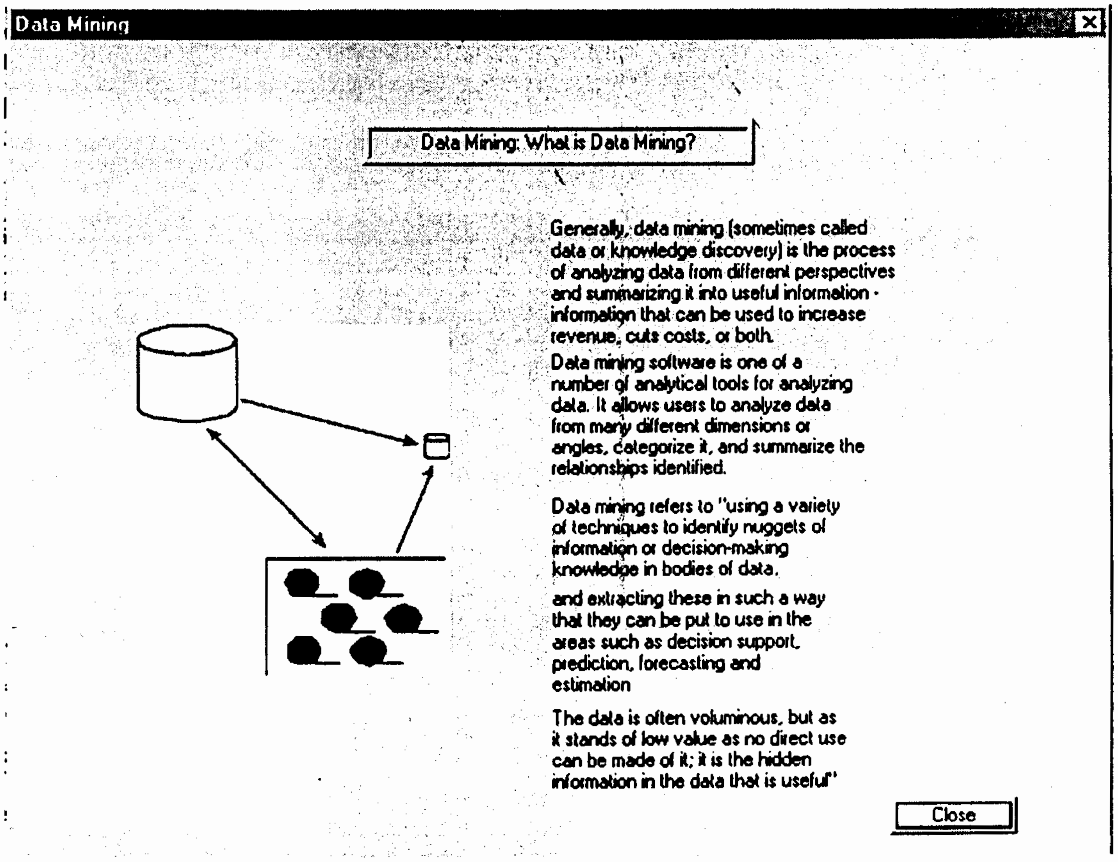


Figure 6.1: The main interface of C Miner

The **What is Data Mining?** button leads towards the following window. It displays a brief introduction of data mining. **Close** will close the window.



Generally, 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.

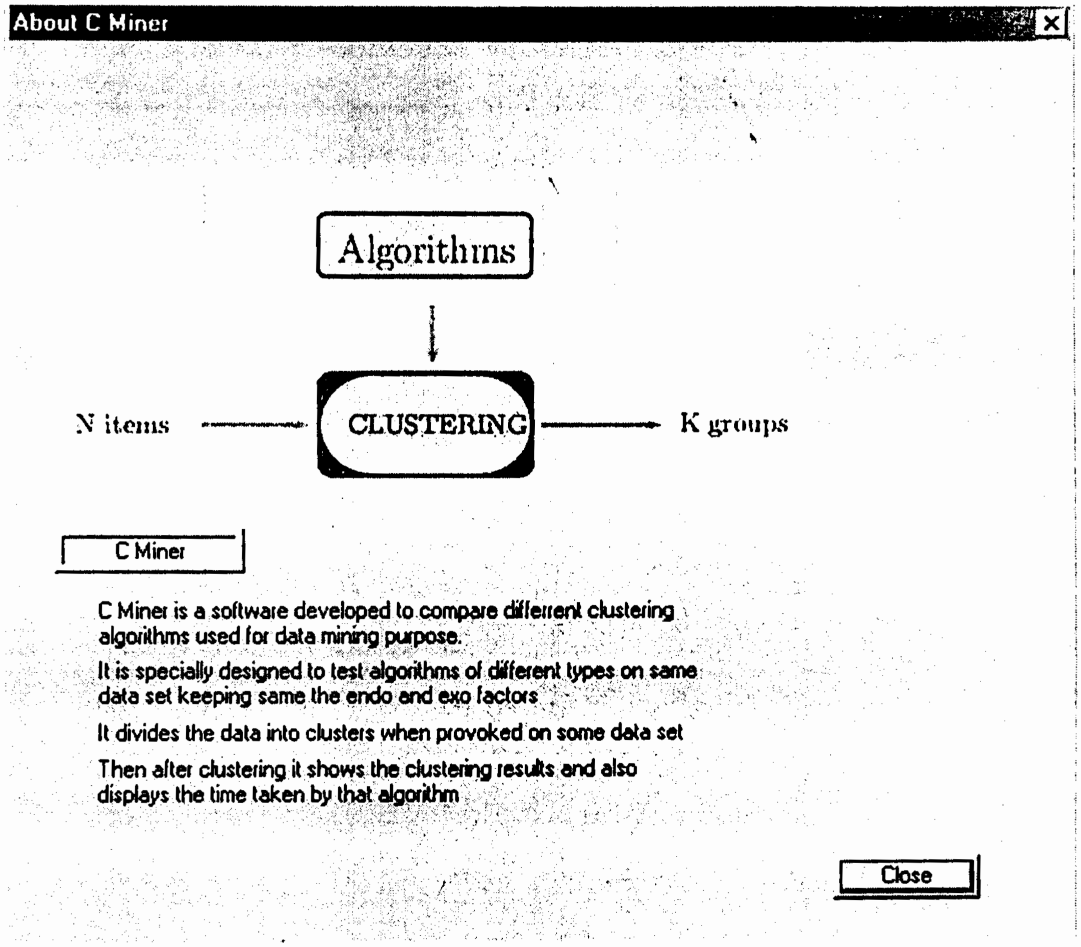
Data mining refers to "using a variety of techniques to identify nuggets of information or decision-making knowledge in bodies of data.

and extracting these in such a way that they can be put to use in the areas such as decision support, prediction, forecasting and estimation

The data is often voluminous, but as it stands of low value as no direct use can be made of it; it is the hidden information in the data that is useful"

Figure 6.2: A brief Introduction of Data Mining

The **About C Miner** button leads towards the following window. It displays a bricf introduction of C Miner. **Close** will close the window.



C Miner is a software developed to compare different clustering algorithms used for data mining purpose.

It is specially designed to test algorithms of different types on same data set keeping same the endo and exo factors

It divides the data into clusters when provoked on some data set

Then after clustering it shows the clustering results and also displays the time taken by that algorithm

Figure 6. 4: About C Miner

This is the main process window of C Miner. The process is divided into two steps. In step 1 we have to select the two algorithms to be compared, and then we load a data set. In step 2 we run both algorithms on loaded data set one by one. The results are displayed in the respective text areas.

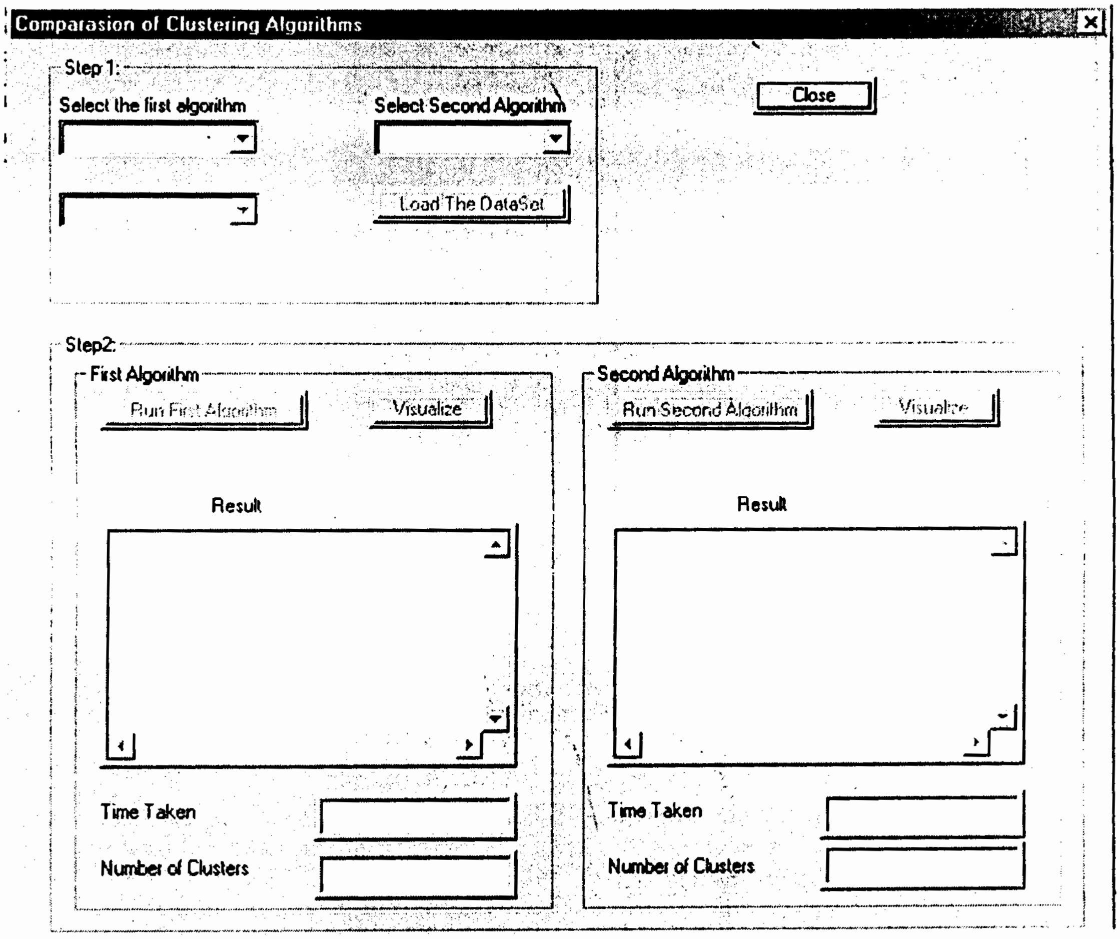
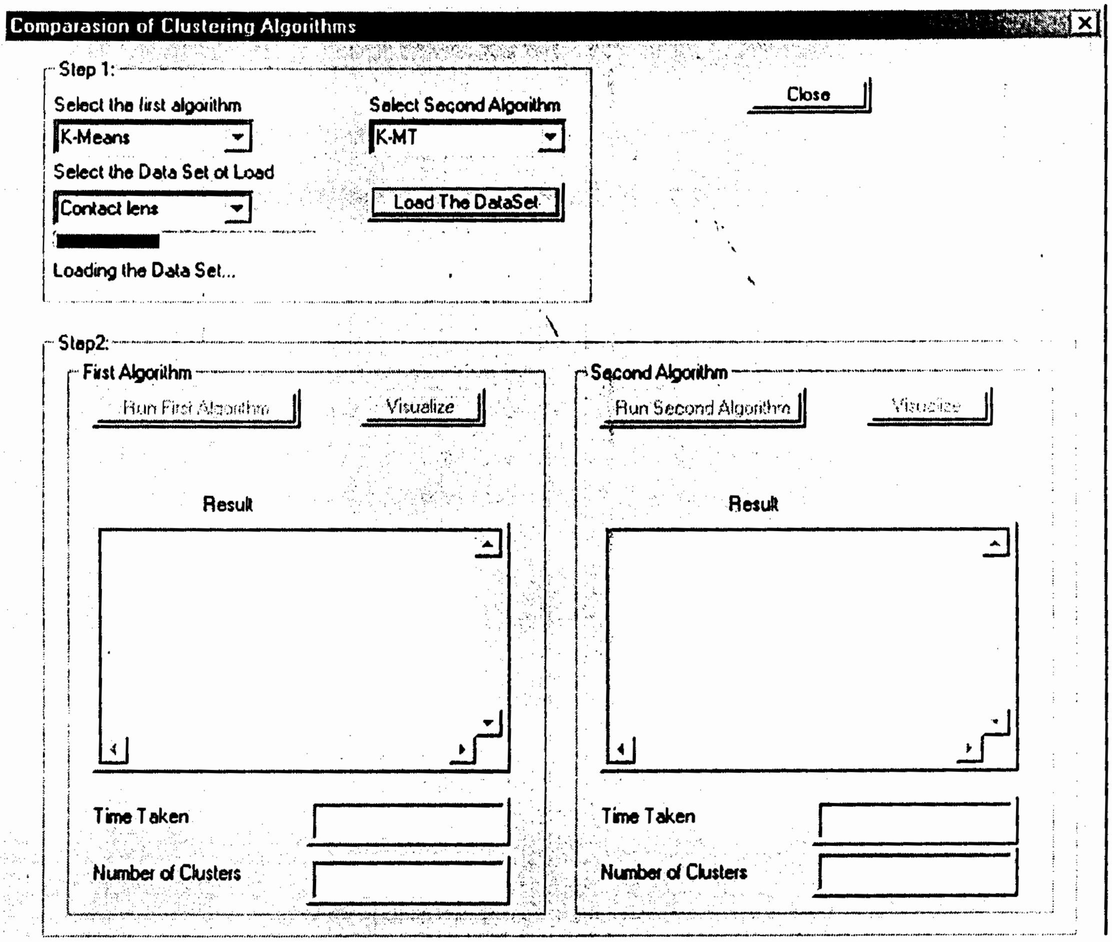


Figure 6.5: The Window showing the comparison environment.



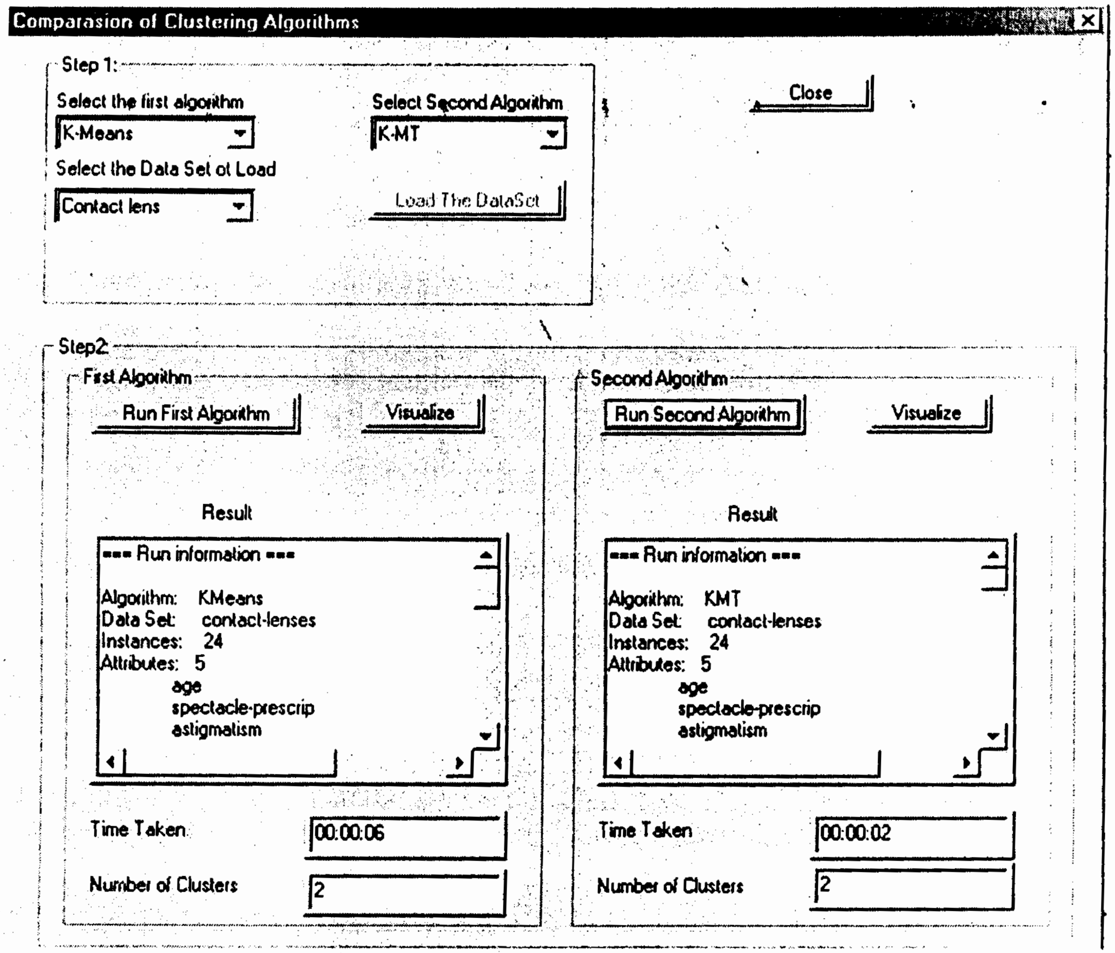
Select the first algorithm: K-Means

Select Second Algorithm: K-MT

Select the Data Set ot Load: Contact lens

Loading the Data Set...

Figure 6.6: Loading a data set.



Select the first algorithm: K-Means

Select Second Algorithm: K-MT

Select the Data Set ot Load: Contact lens

First Algorithm

```
=== Run information ===

Algorithm:  KMeans
Data Set:   contact-lenses
Instances:  24
Attributes: 5
            age
            spectacle-prescrip
            astigmatism
```

Time Taken: 00:00:06

Number of Clusters: 2

Second Algorithm

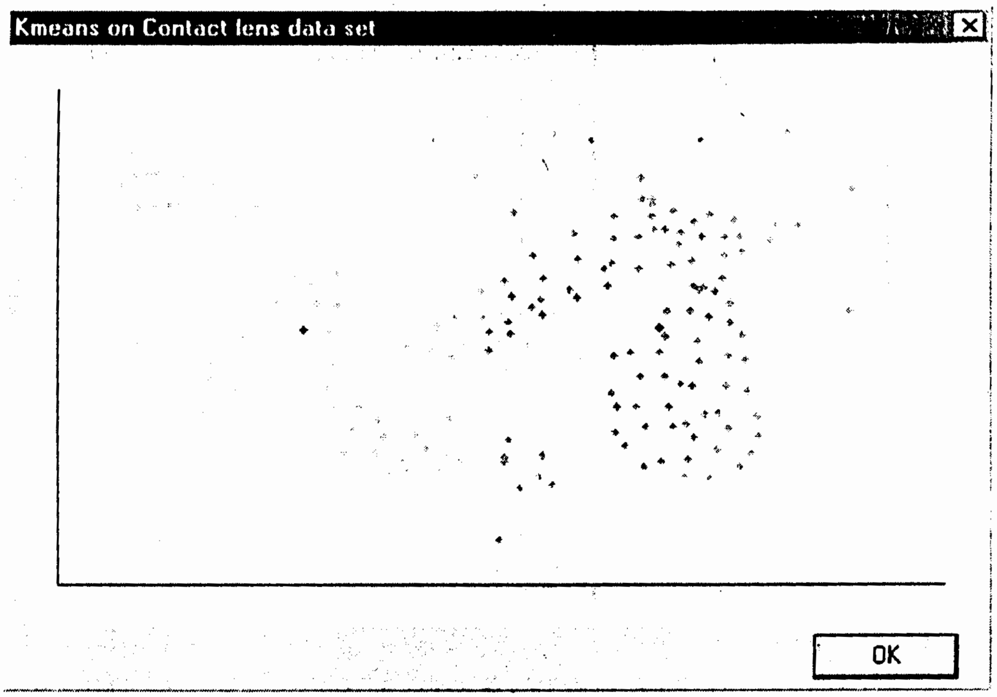
```
=== Run information ===

Algorithm:  KMT
Data Set:   contact-lenses
Instances:  24
Attributes: 5
            age
            spectacle-prescrip
            astigmatism
```

Time Taken: 00:00:02

Number of Clusters: 2

Figure 6.5: The Window showing the results of \both algorithms after running them.

The visualize button displays a graphical representation of the clusters formed after applying the algorithm



CHAPTER 7
REFERENCES

REFERENCES

1. Knowledge Discovery Nuggets: http://www.kdnuggets.com/
2. Adriaans, P. and Zantinge, D.: Data Mining, Addition-Wesley, 1996.
3. Berry, M. and Linoff, G.: Data Mining Techniques for Marketing, Sales and Customer Support, John Wiley & Sons, Inc., 1997.
4. Dorian, P.: Data Preparation for Data Mining, Morgan Kaufmann, 1999.
5. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, S., and Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining, M.I.T. Press, 1996.
6. Liu, H. and Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining, Kluwer International, 1998.
7. Weiss, S.M. and Kulikowski, C.A.: Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann, 1991.
8. Weiss, S.M. and Indurkhya, N.: Predictive Data Mining: A Practical Guide, Morgan Kaufmann, 1997.
9. Westphal, C. and Blaxton, T.: Data Mining Solutions: Methods and Tools for Real-World Problems, Wiley, 1998
10. S. Theodoridis, K. Koutroubas (1999). Pattern Recognition. Academic Press
11. S. Guha, R.Rastogi, K. Shim(1998). CURE: An Efficient Clustering Algorithm for Large Databases. In Proceedings of the ACM SIGMOD Conference
12. S. Guha, R.Rastogi, K. Shim(1998). CURE: An Efficient Clustering Algorithm for Large Databases. In Proceedings of the ACM SIGMOD Conference.
13. M. Halkidi, Y. Batistakis, M. Vazirgiannis: On Clustering Validation Techniques, Department of Informatics, Athens University of Economics & Business
14. [M.J.A . Berry, G. Linoff (1996). Data Mining Techniques For Marketing, Sales and Customer Support. John Wiley & Sons, Inc., USA.
15. M.U. Fayyad, G. Piatesky-Shapiro, P. Smuth, R. Uthurusamy (1996). Advances in Knowledge Discovery and Data Mining. AAAI Press.
16. Rezaee, B.P.F. Lelieveldt, J.H.C. Reiber. (1998). A New Cluster Validity Index for the Fuzzy c-Mean. Pattern Recognition Letters, 19, 237–246.
17. MacQueen, J.B. Some Methods for Classification and Analysis of Multivariate Observations. In Proceedingsof 5th Berkley Symposium on Mathematical Statistics and Probability, Volume I:

CHAPTER 8
PUBLICATIONS

From: "Euro Asian Journal of Applied Sciences" <euroasianjournals@lycos.com> Add to Address Book

To: "Sikandar Khiyal" <sikandar_hayat_khiyal@yahoo.com>

Date: Thu, 17 Feb 2005 13:29:49 -0500

Subject: your article - REF 00113

Dear Dr. Khiyal

Your article has been accepted for publication. It will be processed as soon as we receive your publication fee of $ 69.00 through western union. (for a western union agent in your proximity please refer to www.westernunion.com). please make the payment as instructed below.

Sincerely,

Adrian M. Steinberg, PhD
Editor-In-Chief
European Journal of Scientific Research
Euro-Asian Journal of Applied Sciences
http://www.lulu.com/content/105819

PUBLICATION FEE PAYMENT INSTRUCTIONS FOR ACCEPTED ARTICLES:

Please make the payment through western union with the following details:

Beneficiary Name: METE FERIDUN
Recepient country: NORTHERN CYPRUS (not "CYPRUS" or "TURKEY" - please note that this is a common mistake)

Please forward us the 10 digit MTCN code along with the name of the sender exactly as it appears on the payment slip

--

C-minor platform for comparing different clustering algorithms

Tehmina Amjad[1], Dr. Malik Sikandar Hayat Khiyal[1] and Dr. S. Tauseef-ur-Rehman[2]
[1]Department of Computer Science, Faculty of Applied Sciences,
[2]Department of Telecommunication Engineering and Computer Engineering,
International Islamic University, Sector H-10,
Islamabad, Pakistan

Abstract

In this paper focus is on specific algorithms of clustering technique and their review in a comparative manner. It describes a system designed to choose an appropriate algorithm of the Clustering technique, for a given dataset and its comparison with another algorithm using same data set. The design will provide a platform for such comparisons under same endo and exo factors.

1. Introduction

Clustering distributes data into several groups so that similar objects fall into same group. The objective is to create clusters where the intra-dissimilarity is minimized and inter-dissimilarity is maximized. Clustering is one of the most useful tasks in the process of data mining for discovering groups in the underlying data and identifying interesting distributions and patterns in it [1]. Clustering problem is about partitioning a given data set into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters [2]. Cluster analysis divides data into groups (clusters) for the purposes of summarization or improved understanding. For example, consider a retail database records containing items purchased by customers. A clustering procedure could group the customers in such a way that customers with similar buying patterns are in the same cluster. Thus, the main concern in the clustering process is to reveal the organization of patterns into "sensible" groups, which allow us to discover similarities and differences, as well as to derive useful conclusions about them. This idea is applicable in many fields, such as life sciences, medical sciences, engineering, statistics, pattern recognition, data mining, and other fields. Clustering may be found under different names in different contexts, such as unsupervised learning (in pattern recognition), numerical taxonomy (in biology, ecology), typology (in social sciences) and partition (in graph theory) [3]. Although the field of clustering has a long history and a large number of clustering techniques have been developed, but still the significant challenges prevail.

1.1 Difference between Clustering and Classification

In clustering no predefined classification is required and no prevailing examples that could demonstrate that what kind of required relations should be valid among the data. That's why Clustering is termed as Unsupervised Process [4]. The task is to learn a classification from the data. Clustering algorithms divide a data set into Natural groups (Clusters). On the other hand, classification is a procedure of assigning a data item to a predefined set of categories [5]. Clustering produces initial categories in which values of a data set are classified during the classification process.

Unsupervised learning: Clustering is an unsupervised task, i.e., the training data doesn't specify what we are trying to learn (the clusters).

Supervised learning: Classification requires supervised learning, i.e., the training data has to specify what we are trying to learn (the classes).

1.2 Different Steps in Clustering Process

The clustering process may result in different partitioning of a data set, depending on the specific criterion used for clustering. Thus, there is a need of preprocessing before we assume a clustering task in a data set. The basic steps to develop clustering process are presented in Fig 1.1 and can be summarized as follows [1,5].

1.2.1 Feature selection. The goal is to select properly the features on which clustering is to be performed so as to encode as much information as possible concerning the task of our interest. Thus, preprocessing of data may be necessary prior to their utilization in clustering task.

1.2.2 Clustering algorithm. This step refers to the choice of an algorithm that results in the definition of a good clustering scheme for a data set. A proximity measure and a clustering criterion mainly characterize a clustering algorithm as well as its efficiency to define a clustering scheme that fits the data set.

Proximity measure: It is a measure that quantifies how "similar" two data points (i.e. feature vectors) are. In most of the cases we have to ensure that all selected features contribute equally to the computation of the proximity measure and there are no features that dominate others.

Clustering criterion: In this step, we have to define the clustering criterion, which can be expressed via a cost function or some other type of rules. We should stress that we have to take into account the type of clusters that are expected to occur in the data set. Thus, we may define a "good" clustering criterion, leading to a partitioning that fits well the data set.

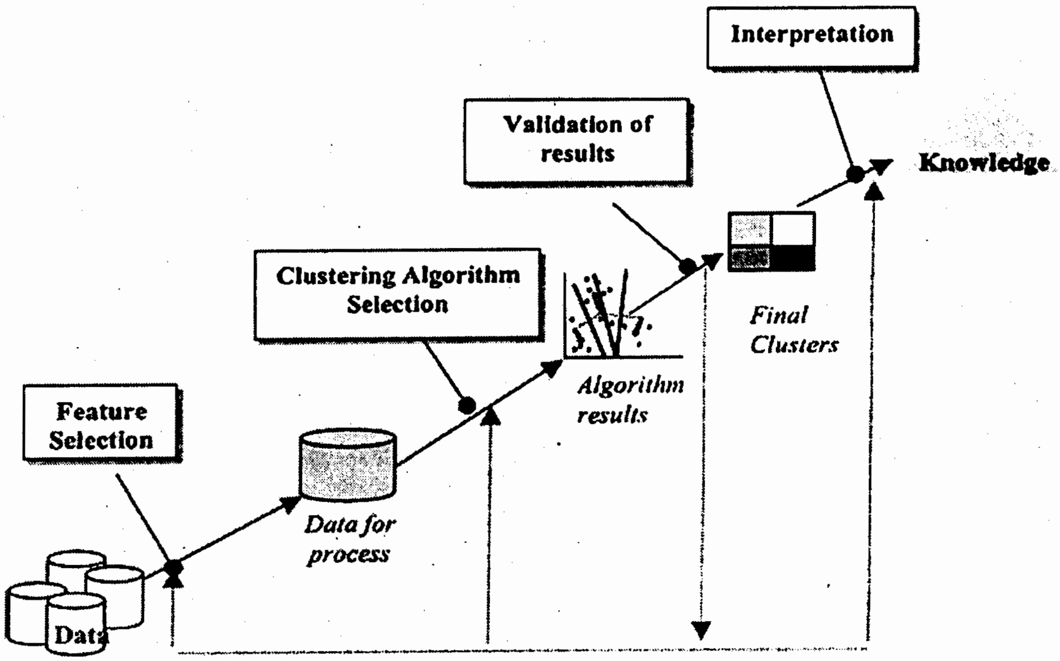


Fig 1.1 Steps of clustering process

1.2.3 Validation of the results. The correctness of clustering algorithm results is verified using appropriate criteria and techniques. Since clustering algorithms define clusters that are not known a priori, irrespective of the clustering methods, the final partition of data requires some kind of evaluation in most applications [6].

1.2.4 Interpretation of the results. In many cases, the experts in the application area have to integrate the clustering results with other experimental evidence and analysis in order to draw the right conclusion.

1.3 Clustering Applications

Recently, clustering has been applied to a wide range of topics and areas. Uses of clustering techniques can be found in pattern recognition, as in following:

Spatial Data Analysis: Clustering in spatial data mining is to group similar objects based on their distance, connectivity, or their relative density in space. In the real world, there exist many physical obstacles such as rivers, lakes and high-ways, and their presence may affect the result of clustering substantially

Image Processing: Clustering in image processing is the grouping together of pixels from an image, depending the calculated similarity between them. Clustering can be often defined as an unsupervised classification of pixels. The color image data is naturally clustered in three-dimensional color space (usually RGB). All dominant colors in the image create dense clusters in the color space.

Economic Science (especially market research): The reason people lend themselves to clustering in market research is that they wish to differentiate themselves. You just draw a circle around a bunch of dots on your market segmentation map and proclaim that a target market.

WWW: Document classification and Cluster Web log data to discover groups of similar access patterns.

1.3 Clustering Algorithm Categories

The clustering algorithms are categorized into different domains depending upon the following.

- The type of data input to the algorithm.
- The definition of the similarity between data points.
- The theory and fundamental concepts on which clustering analysis techniques are based (e.g. fuzzy theory, statistics).

Thus according to the method adopted to define clusters, the algorithms can be broadly classified into the following types:

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

2. Clustering Algorithms

2.1. Partitional Algorithms

The following are the most commonly used algorithms in the category of partitional algorithms.

K-Means is the most commonly used algorithm in this category. It discovers K (non-overlapping) clusters by finding K centroids ("central" points) and then assigning each point to the cluster associated with its nearest centroid (A cluster centroid is typically the mean or median of the points in its cluster and "nearness" is defined by a distance or similarity function). The algorithm begins by initializing a set of *c* cluster centers. Then, it assigns each object of the dataset to the cluster whose center is the nearest, and re-computes the centers. The process continues until the centers of the clusters stop changing.

PAM (***Partitioning Around Medoids***) is another algorithm in this category. The objective of PAM is to determine a representative object (*medoid*) for each cluster, that is, to find the most centrally located objects within the clusters. The algorithm begins by selecting an object as medoid for each of *c* clusters. Then, each of the non-selected objects is grouped with the medoid to which it is the most similar. PAM swaps medoids with other non-selected objects until all objects qualify as medoid. It is clear that PAM is an expensive algorithm as regards finding the medoids, as it compares an object with entire dataset [7].

CLARA (***Clustering Large Applications***) is an implementation of PAM in a subset of the dataset. It draws multiple samples of the dataset, applies PAM on samples, and then outputs the best clustering out of these samples [7].

CLARANS (***Clustering Large Applications based on Randomized Search***) combines the sampling techniques with PAM. The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids. The clustering obtained after replacing a medoid is called the *neighbor* of the current clustering. CLARANS selects a node and compares it to a user-defined number of their neighbors searching for a local minimum. If a better neighbor is found (i.e., having lower-square error), CLARANS moves to the neighbor's node and the process start again; otherwise the current clustering is a local optimum. If the local optimum is found, CLARANS starts with a new randomly selected node in search for a new local optimum. Finally *K*-prototypes, *K-mode* [8] are based on *K*-Means algorithm, but they aim at clustering categorical data.

2.2. Hierarchical Algorithms

Hierarchical clustering algorithms according to the method that produce clusters can further be divided into [3]:

Agglomerative algorithms. They produce a sequence of clustering schemes of decreasing number of clusters at east step. The clustering scheme produced at each step results from the previous one by merging the two closest clusters into one.

Divisive algorithms. These algorithms produce a sequence of clustering schemes of increasing number of clusters at each step. Contrary to the agglomerative algorithms the clustering produced at each step results from the previous one by splitting a cluster into two.

Some representative *hierarchical clustering* algorithms.

BIRCH [9] uses a hierarchical data structure called CF-tree for partitioning the incoming data points in an incremental and dynamic way. CF-tree is a height balanced tree, which stores the clustering features and it is based on two parameters: *branching factor B* and *threshold T*, which referred to the diameter of a cluster (the diameter (or radius) of each cluster must be less than *T*). *BIRCH* can typically find a good clustering with a single scan of the data and improve the quality further with a few additional scans. It is also the first clustering algorithm to handle noise effectively [9]. However, it does not always correspond to a natural cluster, since each node in CF-tree can hold a limited number of entries due to its size. Moreover, it is order-sensitive as it may generate different clusters for different orders of the same input data. *CURE* [2] represents each cluster by a certain number of points that are generated by selecting well-scattered points and then shrinking them toward the cluster centroid by a specified fraction. It uses a combination of random sampling and partition clustering to handle large databases. *ROCK* [10], is a robust clustering algorithm for Boolean and categorical data. It introduces two new concepts, that is a point's neighbors and links, and it is based on them in order to measure the similarity/proximity between a pair of data points.

2.3. Density-based Algorithms

Density based algorithms typically regard clusters as dense regions of objects in the data space that are separated by regions of low density. A widely known algorithm of this category is DBSCAN [11]. The key idea in DBSCAN is that for each point in a cluster, the neighborhood of a given radius has to contain at least a minimum number of points. DBSCAN can handle noise (outliers) and discover clusters of arbitrary shape. Moreover, DBSCAN is used as the basis for an incremental clustering algorithm proposed in [12]. Due to its density-based nature, the insertion or deletion of an object affects the current clustering only in the neighborhood of this object and thus efficient algorithms based on DBSCAN can be given for incremental insertions and deletions to an existing clustering [12]. In [13] another density-based clustering algorithm, DENCLUE, is proposed. This algorithm introduces a new approach to cluster large multimedia databases. The basic idea of this approach is to model the overall point density analytically as the sum of influence functions of the data points. The influence function can be seen as a function, which describes the impact of a data point within its neighborhood. Then clusters can be identified by determining density attractors. Density attractors are local maximum of the overall density function. In addition, clusters of arbitrary shape can be easily described by a simple equation based on overall density function. The main advantages of DENCLUE are that it has good clustering properties in data sets with large amounts of noise and it allows a compact mathematically description of arbitrary shaped clusters in high-dimensional data sets. However, DENCLUE clustering is based on two parameters and as in most other approaches the quality of the resulting clustering depends on the choice of them. These parameters are [13]:

- parameter N which determines the influence of a data point in its neighborhood and
- < describes whether a density-attractor is significant, allowing a reduction of the number of density-attractors and helping to improve the performance.

2.4. Grid-based Algorithms

Recently a number of clustering algorithms have been presented for spatial data, known as grid-based algorithms. These algorithms quantize the space into a finite number of cells and then do all operations on the quantized space. STING (Statistical Information Grid-based method) is representative of this category. It divides the spatial area into rectangular cells using a hierarchical structure. STING [14] goes through the data set and computes the statistical parameters (such as mean, variance, minimum, maximum and type of distribution) of each numerical feature of the objects within cells. Then it generates a hierarchical structure of the grid cells so as to represent the clustering information at different levels. Based on this structure STING enables the usage of clustering information to search for queries or the efficient assignment of a new object to the clusters.

WaveCluster [15] is the latest grid-based algorithm proposed in literature. It is based on signal processing techniques (wavelet transformation) to convert the spatial data into frequency domain. More specifically, it first summarizes the data by imposing a multidimensional grid structure onto the data space [16]. Each grid cell summarizes the information of a group of points that map into the cell. Then it uses a wavelet transformation to transform the original feature space. In wavelet transform, convolution with an appropriate function results in a transformed space where the natural clusters in the data become distinguishable. Thus, we can identify the clusters by finding the dense regions in the transformed domain. A-priori knowledge about the exact number of clusters is not required in WaveCluster.

3. Comparison of Partitional Clustering Algorithms

Clustering is broadly recognized as a useful tool in many applications. Researchers of many disciplines have addressed the clustering problem. However, it is a difficult problem, which combines concepts of diverse scientific fields (such as databases, machine learning, pattern recognition, and statistics). Thus, the differences in assumptions and context among different research communities caused a number of clustering methodologies and algorithms to be defined.

This section offers an overview of the main characteristics of the Partitional clustering algorithms presented in a comparative way.

More specifically our study is based on the following features of the algorithms:

- The type of the data that an algorithm supports (numerical, categorical).
- The shape of clusters.
- Ability to handle noise and outliers.
- The clustering criterion.
- Complexity.

3.1 COBWEB

COBWEB is an incremental clustering algorithm, based on probabilistic categorization trees. The search for a good clustering is guided by a qualitative measure for partitions of data. The algorithm reads **unclassified examples,** given in **attribute-value representation.** Pure COBWEB only supports nominal attributes; later extensions incorporate numerical values as well. That's why Cobweb performs well on the Contact Lens and Iris data cell but the performance decreases on CPU data set which shows that its scalability and versatility is affected.

3.2 K MEANS

K- Means is a least squares partitioning method allowing users to divide a collection of objects into k groups. During iterations it tries to minimize the distances of the objects to the respective group centroids. With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small). K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular. K-means is one of the so-called NP-Hard problems, so it can't guarantee that the absolute minimum of objective function has been reached. However it is scalable and has minimal input parameter. But the problem is that different initial partitions can result in different final clusters. In the experiments that I performed k-means worked well for the Contact Lens data set but its performance is reduced for CPU and Iris data set.

3.3 DBSCAN

The key idea of density-based clustering is that for each point of a cluster the neighborhood of a given radius (Eps) has to contain at least a minimum number of points (MinPts), i.e. the cardinality of the neighborhood has to exceed some threshold. It is significantly more affective in discovering clusters of the arbitrary shapes. The requirements of domain knowledge to determine input parameter are minimal. It is also affective for large databases.

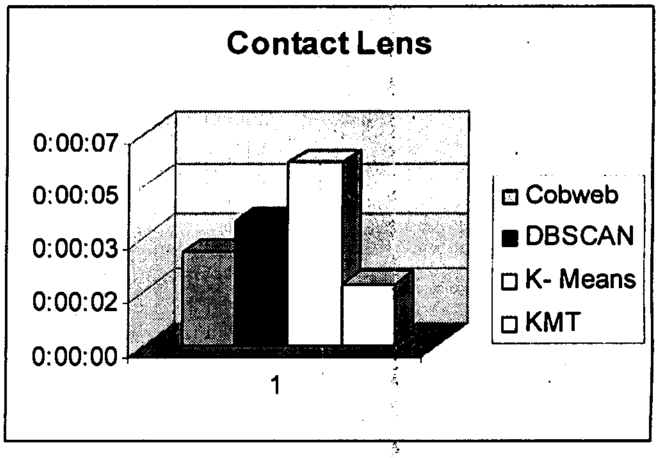
3.4 KMT

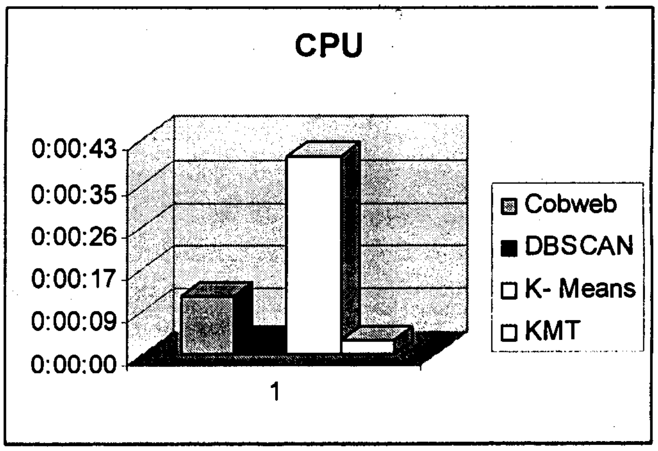
KMT reduces the size and time. The size is reduced as the number of iterations is minimized and time is automatically reduced as the number of iterations is reduced. So it outperforms k-means and thus extends the size of the size of the data sets that can be clustered. It differs from the initial version in how the initial means are chosen. As a result is shows efficient result on all the three data sets.

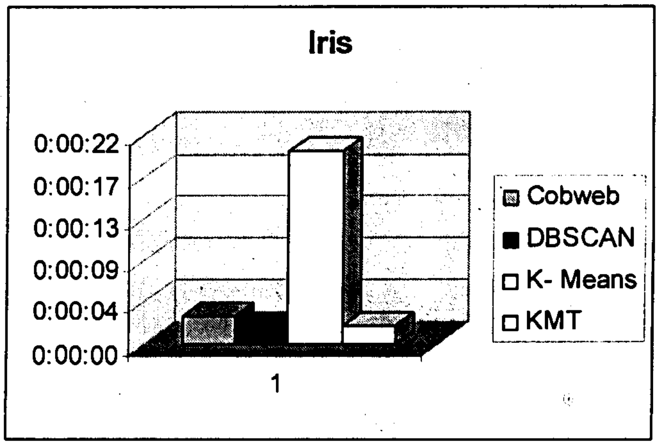
3.5 Results

Different results are obtained when C Miner is executed on different data sets using different algorithms. The following table shows the time taken by each algorithm when executed on different data sets.

Data Set	Algorithm	Time taken
Contact Lens	Cobweb	00:00:03
	DBSCAN	00:00:04
	K- Means	00:00:06
	KMT	00:00:02
CPU	Cobweb	00:00:12
	DBSCAN	00:00:03
	K- Means	00:00:40
	KMT	00:00:03
IRIS	Cobweb	00:00:03
	DBSCAN	00:00:02
	K- Means	00:00:20
	KMT	00:00:02







References:

[1] M. Halkidi, Y. Batistakis, M. Vazirgiannis: On Clustering Validation Techniques, *Department of Informatics, Athens University of Economics & Business*

[2] S. Guha, R.Rastogi, K. Shim(1998). CURE: An Efficient Clustering Algorithm for Large Databases. In *Proceedings of the ACM SIGMOD Conference.*

[3] S. Theodoridis, K. Koutroubas (1999). Pattern Recognition. Academic Press.

[4] M.J.A . Berry, G. Linoff (1996). *Data Mining Techniques For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., USA.

[5] M.U. Fayyad, G. Piatesky-Shapiro, P. Smuth, R. Uthurusamy (1996). *Advances in Knowledge Discovery and Data Mining.* AAAI Press.

[6] R. Rezaee, B.P.F. Lelieveldt, J.H.C. Reiber. (1998). A New Cluster Validity Index for the Fuzzy c-Mean. *Pattern Recognition Letters*, 19, 237–246.

[7] R.Ng, J. Han,(1994). Effecient and Effictive Clustering Methods for Spatial Data Mining. In *Proceeding's of the 20th VLDB Conference*, Santiago, Chile.

[8] Z. Huang (1997). A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets in Data Mining. *DMKD.*

[9] T. Zhang, R. Ramakrishnman, M. Linvy (1996). BIRCH: An Efficient Method for Very Large Databases. *ACM SIGMOD*, Montreal, Canada.

[10] S. Guha, R. Rastogi, K. Shim (1999). ROCK: A Robust Clustering Algorithm for Categorical Attributes. In *Proceedings of the IEEE Conference on Data Engineering.*

[11] M. Ester, H-P. Kriegel, J. Sander, X. Xu (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceeding of 2nd Int. Conf.On Knowledge Discovery and Data Mining.*

[12] M. Ester, H-P. Kriegel, J. Sander, M. Wimmer, X. Xu (1998). Incremental Clustering for Mining in a Data Warehousing Environment. In *Proceedings of 24th VLDB Conference*, New York, USA.

[13] A. Hinneburg, D. Keim (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proceedings of KDD Conference.*

[14] W. Wang, J.Yang, R. Muntz (1997). STING: A Ststistical Information Grid Approach to Spatial Data Mining. In *Proceedings of 23rd VLDB Conference.*

[15] C. Sheikholeslami, S. Chatterjee, A. Zhang (1998). WaveCluster: A-MultiResolution Clustering Approach for Very Large Spatial Database. In *Proceedings of 24th VLDB Conference*, New York, USA.

[16] J. Han, M. Kamber (2001). *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers, USA.