

**DATA CLUSTERING USING MAXIMUM DEPENDENCY
OF ATTRIBUTES AND ITS APPLICATION TO
CLUSTER AGRICULTURAL PRODUCTS**

HAFIZ BIN KAMAL LEANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: _____

SESI PENGAJIAN: _____

Saya: _____
(HURUF BESAR)

mengaku membenarkan tesis (Projek Sarjana Muda/Sarjana/Doktor Falsafah)* ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (4)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

Nama penyelia:

Tarikh: _____

Tarikh: _____

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

*** Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertasi bagi pengajian secara

**DATA CLUSTERING USING MAXIMUM DEPENDENCY OF ATTRIBUTES AND ITS
APPLICATION TO CLUSTER AGRICULTURAL PRODUCTS**

HAFIZ BIN KAMAL LEANG

**A thesis submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)**

FACULTY OF COMPUTER SYSTEM AND SOFTWARE ENGINEERING

UNIVERSITI MALAYSIA PAHANG

MAY, 2012

SUPERVISOR’S DECLARATION

“I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the Degree in Computer Science (Software Engineering)”

Signature :

Supervisor : TUTUT HERAWAN

Date :

STUDENT'S DECLARATION

I hereby declare that this thesis entitled "*Clustering Data Using Maximum Dependency of Attributes and its Application to Cluster Agricultural Products*" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : Hafiz Bin Kamal Leang

Date :

DEDICATION

To my beloved parents and also my lovely siblings for always being there for me through my days and nights.

To Dr. Tutut Herawan, thank you for your guidance, advices and support in completing the task.

My friends. Thanks for your support and cooperation.

“May Allah bless yours”

Sincerely

Hafiz Bin Kamal Leang

ACKNOWLEDGMENT

Alhamdulillah, I am very grateful and thankful to Allah SWT for giving me the strength, patience and ability to complete this thesis project. Without His help, this thesis would not be possible for me to complete.

Firstly, I would like to express my biggest gratitude to my supervisor, Dr. Tutut Herawan for always being there for me to guide, giving ideas, encouragement and constant support in this research. Without him, this research will not be complete like it should be.

Secondly, I want to give my sincere gratitude to my beloved family for the support that they gave when I need it the most. Especially to my parents who work day and night to support my study and research.

Lastly, I want to give my thanks to all my friend and to all individuals who have helped me, and contribute any kind of effort in making my research a success.

ABSTRACT

This project is about understanding the method of Clustering Data using Rough set Theory. The technique used is Maximum Dependency of attributes. The way this technique work is by calculating the degree of each attribute and selecting the highest dependency based on the degree. The highest degree of attribute will be chosen as the best attribute to be used to cluster the data. A system will be built by using Visual Basic (VB) that will implement this technique to cluster large data faster and easier.

ABSTRAK

Projek ini adalah mengenai kajian untuk memahami teknik untuk mengklasifikasikan data menggunakan teori set kasar. Teknik yang digunakan adalah teknik pergantungan maksimum sifat-sifat. Teknik ini digunakan dengan mengira darjah setiap sifat dan seterusnya memilih pergantungan yang paling tinggi berdasarkan darjah yang dikira. Darjah sifat yang paling tinggi akan dipilih sebagai sifat yang paling bagus untuk mengklasifikasikan data. Sebuah sistem akan dibina menggunakan perisian komputer Visual Basic (VB) yang fungsinya untuk melaksanakan teknik ini dalam mengklasifikasikan data yang besar dengan cepat dan senang.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	SUPERVISOR DECLARATION	ii
	STUDENT'S DECLARATION	iii
	DEDICATION	iv
	ACKNOWLEDGEMENT	v
	ABSTRACT	vi
	ABSTRAK	vii
	TABLE OF CONTENTS	viii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
1	INTRODUCTION	
	1.1 Background	1-3
	1.2 Problem Statement	3
	1.3 Objectives	3
	1.4 Scopes	3
	1.5 Thesis Organization	4
2	LITERATURE REVIEW	
	2.1 Agriculture	5-6
	2.1.1 Agriculture in Malaysia	6-8
	2.2 Knowledge Discovery in Databases	8-9
	2.2.1 KDD Process	9-10
	2.2.2 Example of KDD Process	10-13
	2.2.3 Application of KDD in computer science fields	13-14
	2.3 Data Mining	15-16
	2.3.1 Example of Data Mining	16-26
	2.3.2 Application of Data Mining in computer fields	26-27

2.4	Data Clustering	27-28
2.4.1	Classification vs Clustering	29-31
2.4.2	Clustering Techniques	31-35
2.4.3	Clustering on Numerical Dataset	35
2.4.4	Clustering on Categorical Dataset	36-37
2.4.5	Applications of Clustering Techniques	37-38
2.5	Rough set Theory	38-39
2.5.1	Fuzzy Set	39
2.5.2	Relation between fuzzy and rough set theories	40-41
2.5.3	Application of rough set	41
2.5.4	Rough Clustering	41-42
2.5.5	Rough set theory in categorical data clustering	42-43
3	METHODOLOGY	
3.1	Rough Set Theory	44-45
3.1.1	Information System	45-48
3.1.2	Indiscernibility Relation	49
3.1.3	Set Approximations	50-53
3.2	Maximum Dependency of Attributes (MDA)	53
3.2.1	Selecting a clustering attribute	53
3.2.2	Model for selecting a clustering attribute?	53
3.3	Maximum Dependency of Attributes	54
3.3.1	Dependency of Attributes in a Information System	54-55
3.3.2	Algorithm of MDA	55-56
3.3.3	Example	56-68
3.4	Object Splitting Model	69
3.4.1	A clustering attribute with the Max-Max Roughness is found	69
3.4.2	The splitting point attributes a_1 is determined	69-70
4	RESULT AND DISCUSSION	
4.1	Implementation	71
4.2	Datasets	71-72
4.3	Interface	73-85
5	CONCLUSIONS	86
	REFERENCES	87-91
	APPENDIX	92-105

LIST OF TABLE

TABLE NO.	TITLE	PAGE
1	A simple example of database	17
2	Logical database corresponding with the original database	18
3	Value set of attribute items in database	19
4	K=1 Items and corresponding larger sets	20
5	K=2 Items and corresponding larger sets	21
6	Confidence of K=2 Larger sets	21
7	K=3 Items and corresponding larger sets	22
8	Confidence of K=3 larger sets	23
9	K=4 items and corresponding larger sets	24
10	Confidence of {1.5.7.9} 4 larger sets	25
11	An information system	45
12	A mushrooms decision system	46
13	Data of bananas	48
14	Algorithm of MDA	56
15	Mushrooms datasets	57
16	Calculation of the degree of dependency attributes in table 15	68
17	Maximum Dependency of Attributes	69

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1	Preprocessing	12
2	KDD Process	13
3	Data Clustering	28
4	Set approximations	51
5	Clustering Attribute Diagram	53
6	Main Interface	73
7	Creator Window	73
8	About Window	74
9	Function Window	74

CHAPTER 1

INTRODUCTION

This chapter briefly discuss on the overview of this research. It contains five parts. The first part is background of the research, followed by the problem statement. Next are the objectives where the project goals are determined. After that the scopes of the system and lastly is the thesis organization which briefly describes the structure of this thesis.

1.1 Background

Knowledge discovery is a concept that describes the process of automatically searching large volumes of data for patterns that can be considered knowledge about the data. Also known as deriving knowledge from the input data. Knowledge discovery can be divided into categories based on what kind of data is searched and in what form is the result of the search represented. It is also developed out of the data mining domain, and is closely related to it in terms of methodology and terminology. Knowledge discovery is the most well-known branch of data mining and also known as Knowledge Discovery in Database (KDD). The way it works is, it creates abstractions of the input data. Gained through the process is the knowledge that may become additional data that can be used for further usage and discovery.

Data mining is one of the step in KDD process where data analysis is applied and discovery algorithms that, under certain conditions, produce a particular enumeration of patterns over the data. The data mining component of the KDD process usually involves repeated iterative application of particular data mining

method. The methods are classifications, regressions, summarization, dependency modeling, and change and deviation detection. After the general methods of data mining have been outlined, it will then construct specific algorithms to implement these methods. The three primary components that can be identified in any data mining algorithm are model representation, model evaluation, and search.

Clustering is the task of assigning a set of objects into groups so that the objects in the same cluster are more similar to each other than to those in other clusters. Classification is a data mining technique used to predict group membership for data instances. For example, classification can be used to predict whether the weather on a particular day will be “sunny”, “rainy” or “cloudy”.

In real life, there are many of type of data that can be collect to be analyzed. When analyzing the data, there are often problems when we want to group the data according to their uniqueness. This often because there is no unique attributes in the data.

There are many types of fruits that can be found in Malaysia. There are so many types of fruits that sometimes not all of them have been seen or ate by a person. Because of this, fruits also have become one of the main sources of income for people living in Malaysia. The reason for why fruits need to be classified is that, when selling fruits, they need to know what attribute that the fruits have and after that separate it into several groups of fruits. This is so that the fruits can graded and sell with a different price.

In this research, the data that have been used are fruits data. The problem from using this data is, it is hard to group the fruits because of no uniqueness in the

attributes. To solve this problem, this research will use the maximum dependency of attributes technique to group the fruits data.

1.2 Problem Statement

Having no unique attributes makes it hard to group the data. Thus, another technique is used to cluster the agricultural data.

Rough set is used because this technique able to handle with this kind of data compared to other techniques. Most of the other kind of techniques only can handle numerical data type which is not the kind of data used in this research. Rough set techniques can handle multi-valued data in this research.

1.3 Objectives

The following shows the objectives of the research:

- i. To group the mushrooms data according to their dependencies of their attributes
- ii. To apply the rough set technique into real life case.

1.4 Scopes

The scopes of this research are shown below:

- i. The clustering used maximum dependency of attributes technique.
- ii. The used of agricultural data consists of mushrooms.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 1 will contain the introduction of this research. Chapter 2 will contain all the literature review that are found for the purpose of doing this research. Chapter 3 consists of the methodology of this research that includes all the technique, algorithm and all the method that are needed to obtain the objectives of this research. Chapter 4 contain the information of the implementation of the application developed based on this research. Chapter 5 will have the conclusions for this research.

CHAPTER II

LITERATURE REVIEW

This chapter briefly discusses about the literature review of Agriculture, Knowledge Discovery in Database (KDD), Data Mining, Data Clustering, and Rough Set Theory (RST). The first section is about Agriculture, followed by KDD. After that data mining and data clustering, and lastly Rough Set Theory.

2.1 Agriculture

Agriculture is basically referred to as the cultivation of animals, plants, fungi and other life forms for food, fiber, and other products that are used supply human daily life. Agriculture was the main method in rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that nurtured the development of civilization. Agricultural science is the study of agriculture. Agriculture also includes the observation of certain species of ant and termite, but generally speaking refers to human activities.

(<http://en.wikipedia.org/wiki/Agriculture>)

Now days, agriculture products were sold using a knowledge-based intelligent e-commerce system. This system will provides products sales, financial analysis and sales forecasting, and not only that it also provides feasible solutions or actions based on the results of rule-based reasoning. This intelligent system will integrates a database, a rule base and a model base to create a tool of which managers can use to deal with decision-making problems using the internet.(U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, 1996).

In agricultural production, several types of different methodologies and processes that require a rather high energy input. At the same time, the markets require output products of high quality. The activities can be classified based on the applied methodology, technology and application fields (Hashimoto et al., 2002). These issues appear among the scientific topics of the workshops and conferences organized by the two technical committees, which are the Modeling and Control in Agricultural Processes and Intelligent Control in Agricultural Automation, within the International Federation of Automatic Control (IFAC). (I. Farkas, 2003)

2.1.1 Agriculture in Malaysia

Agriculture in Malaysia contributes up to twelve percent of the nation's Gross Domestic Product (GDP). From the population of Malaysia, sixteen percent are employed through some sort of agriculture. British have established the large-scale plantations. Opportunity has been opened by these plantations, for new crops such as rubber (1876), palm oil (1917), and cocoa (1950). A number of crops are grown for domestic purpose such as bananas, coconuts, durian, pineapple, rice, and rambutan. Productions of exotic produce in Malaysia are mainly because of the proper climate in Malaysia. It is located on a peninsula in Southeast Asia. This area is very rarely affected by hurricanes or drought. Humidity level maintains around ninety percent in Malaysia, it is because of its location close to the equator. The weather stays hot and humid all year round. Malaysia is very populated with hills and large scale agriculture requires a huge amount of flat land. Malaysia does not have a strong temperature climate, because of these disadvantages, Malaysia cannot produce enough rice and other food products to supply the country and forces it to import.

(http://en.wikipedia.org/wiki/Agriculture_in_Malaysia)

In 1999, Malaysia produced 10.55 million metric tons of palm oil, thus making it one of the world's largest producers till today. Almost 85 percent or 8.8 million metric tons of this was exported to international market. Malaysia is one of the world's leading suppliers of rubber, producing 767,000 metric tons of rubber in

1999. However, in 1990s, palm oil production is focused more by large plantations companies as it is more profitable. In producing of cocoa, Malaysia has claimed world's fourth-largest with 84,000 metric tons in 1999.

Logging in the tropical rainforest is an important export revenue earner in East Malaysia and in the northern states of Peninsular Malaysia. In 2000, Malaysia produced 21.94 million cubic meters of sawed logs, earning RM1.7 billion from exports. Tropical logs and sawed tropical timber is sold more by Malaysia abroad than any other country, and is one of the biggest exporters of hardwood. Despite attempts at administrative control and strict requirements regarding reforestation in the early 1990s, logging companies often damage the fragile tropical environment. Sharp criticism from local and international environmentalist groups gradually led to bans on the direct export of timber from almost all states, except Sarawak and Sabah. In December 2000, the government and representatives of indigenous and environmentalist groups agreed that there is a need to adopt standards set by the international Forest Stewardship Council (FSC), which certifies that timber comes from well-managed forests and logging companies have to be responsible for reforestation. (<http://www.nationsencyclopedia.com/economies/Asia-and-the-Pacific/Malaysia-AGRICULTURE.html>)

Malaysia has relied heavily on conventional methods to produce, increase and sustain food productions in the early years of developing the agricultural sector. This is because, a large amount of chemicals fertilizers are needed to supply plant nutrients and chemicals to get rid of pest and diseases. However, in recent years, as a result of increasing awareness on health and environment issues, systematic programs have been introduced to optimize the use of resources on a sustainable basis including the recycling of waste products for food production and environment protection. The successful use of agriculture wastes such as rice, straws and husks, empty oil palm fruit bunches, saw dust, animal droppings, POME etc. and the implementation of good agricultural practices including biological control methods such as IPM are positive steps undertaken to reduce the dependence on chemicals,

and to move towards more natural and healthier methods of food production. Integrated and mixed farming is one successful way of optimizing the use of resources for maximizing income.(F. Ahmad, 2001)

2.2 Knowledge Discovery in Databases

Data Mining and Knowledge Discovery in Databases (KDD) are rapidly evolving areas of research that are at the intersection of several disciplines, including statistics, databases, pattern recognition/AI, visualization, and high-performance and parallel computing.(U. Fayyad, 1997).

At an abstract level, the KDD field is concerned with the development of methods and techniques for making sense of data. The basic problem addressed by the KDD process is one of mapping low-level data, which are typically too voluminous to understand and digest easily into other forms that might be more compact, for example, a short report, more abstract, for example, a descriptive approximation or model of the process that generated the data, or more useful, for example, a predictive model for estimating the value of future cases. At the core of the process is the application of specific data-mining methods for pattern discovery and extraction. (W. Wen, 2007)

Knowledge Discovery in Databases (KDD) is the automated discovery of patterns and relationships in large databases. Large databases are not uncommon. Cheaper and larger computer storage capabilities have contributed to the proliferation of such databases in a wide range of fields. Scientific instruments can produce terabytes and petabytes of data at rates reaching gigabytes per hour. Point of sale information, government records, medical records and credit card data, are just a few other sources for this information explosion. Not only are there more large databases, but the databases themselves are getting larger. The number of fields in large databases can approach magnitudes of 10^2 to 10^3 . Record numbers in these databases approach

magnitudes of 10^9 . KDD employs methods from various fields such as machine learning, artificial intelligence, pattern recognition, database management and design, statistics, expert systems, and data visualization. It is said to employ a broader model view than statistics and strives to automate the process of data analysis, including the art of hypothesis generation. KDD has been more formally defined as “the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.” (Susan P. Imberman, 2001)

2.2.1 KDD Process



KDD begins with the Business Case Definition and proceeds with Data Preparation, Data Mining and Evaluation processes in cyclic order. But the processes are very iterative in nature. Any issues or configuration settings in Data Preparation may result into revisiting and fine-tuning the Business Case Definition. Findings or non-interpretable results from Data Mining process may fall back on the Data Preparation or back to Business Case Definition. Same is the case with Evaluation process.

The Knowledge Base is merely a representation of the database where the business case model, data, metadata, data preparation rules, data mining algorithms, results

and evaluation information is kept. It acts as a common pool of information / knowledge, which facilitates the iterations and improves the quality of the model for better results. (Graham J. Williams, Z. Huang, 1996)

The following shows the activities involved in each of the KDD processes. (<http://www.executionmih.com/data-mining/kdd-process-preparation-evaluation.php>)

Business Case Definition

- Business Goals, Objectives, Critical Success Factors
- High level business cases / issues
- Gap analysis with respect to the current business processes and IT systems
- Framework for the complete Data Mining process

Data Preparation

- Data (as well as Metadata) Quality Analysis
- Data Mining input parameter specification
- Data selection and preparation

Data Mining

- Data Management
- Data Mining Model Build
- Output construction in form of Visualization and Interfaces

Evaluation

- Utilization of data mining output in business processes
- Collection of data from the business processes after data mining
- Assessment / interpretation of Data Mining output

2.2.2 Example of KDD Processes

A model of KDD process from an insurance domain has been taken to make a useful example of KDD process. The insurance company maintains large databases with

millions of records to be maintain. Accessible to the customer are the data of transaction oriented, that contain the transactions performed on individual policies. The transaction might be new business, a renewal, a cancellation of a policy, a change to some details of the policy, a claim on a policy, etc. The Source Data is made by the data owners by constructing a relational table and various relational operations from the original data.

The Source Data then transformed into a Working Data by applying a suite of operations on it. The major transformation was from a transaction oriented view of the data to a policy oriented view. Involved in the transformation is an elaborate analysis of the data, which leads to the implementation of a collection of automated operations to perform the task. The important of the automation is so that different transformations could easily be performed on the data as the data became better understood. Figure 1 will describe the actual process with indications of size for a small trial database.

The primary task is encapsulated in the Preprocessing stage. This commenced with the cleansing of the Source Data:

- a. Records with missing (critical) values were removed
- b. Certain field values were transformed to forms more appropriate for analysis.

The transformations ranged from simple calculations, such as the determination of an age rather than a birth date, to mappings of large range categorical values to a smaller set of categorical values (required in the context of particular data mining tools).

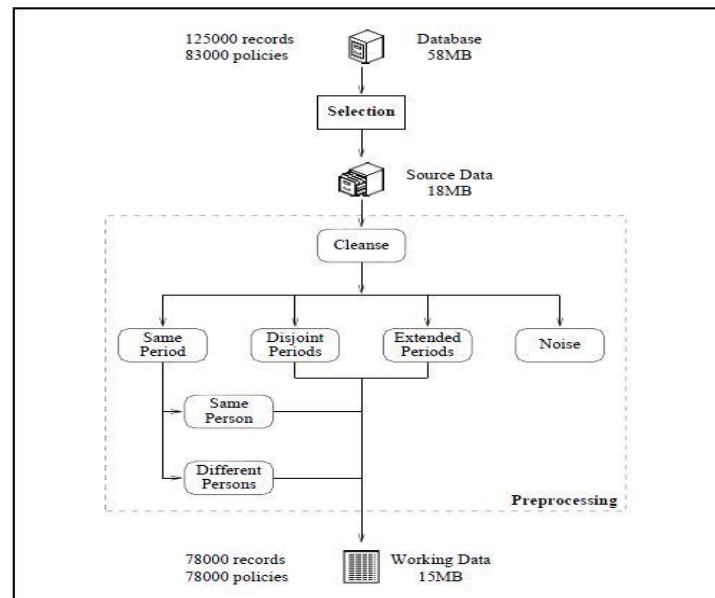


Figure 1: Preprocessing

The major effort expended in the Preprocessing stage was the merging of multiple transactions into single policies. This task required:

- The identification of individual policies
- The merging of multiple transaction policies into single records
- The creation of new fields to record aggregate information

To automate the process of merging, a collection of rules was developed. This facilitated the critical process of revising earlier stage as we iterated through the KDD process. The process was implemented as a collection of filters that could be linked together manually or via a user interface.

Having produced a clean Working Dataset, a task that required considerable effort, the task of most interest to the customer which is exploring the data with a variety of tools, could be addressed. This exploration required revisions to be made to earlier decisions, including the extraction of further attributes from the database and various tuning of the cleansing and merging tasks.

A variety of Data Mining explorations of the data were performed, although many not proving to be particularly insightful, but some leading to interesting snippets of knowledge. StarTree from the Darwin suite of Data Mining tools (Thinking

Machines Corporation 1995), for example, was used to build a decision tree to predict if a claim might be made of policy. This analysis identified a number of hot spots in the data which, when combined with further information derived from the data (relating to periods of exposure and size of claim costs, could be used to pinpoint previously unrecognized high risk areas.

The most important element in this KDD exercise was the determination of whether the discovered patterns were useful. Subjective opinion led to the development of some objective criteria for the evaluation of the patterns discovered. For example, a discovered rule was deemed to be interesting if it was derived from multiple policies where the total claim cost was significant (above a certain threshold). In determining the worthiness of a rule, extra data not used in the actual Data Mining stage was used, again sometimes requiring modifications to be made to earlier stages of the KDD process.

2.2.3 Application of KDD in computer science fields

Knowledge discovery in database (KDD) is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. In computer science, KDD is mostly used to manage a large amount of data using data mining.



Figure 2: KDD process

Figure 2 shows the process of KDD. It is interactive and iterative involving, more or less, the following steps (R. Kalavathy, R.M. Suresh, R. Akhila, 2007):

- a. Understanding the application domain includes relevant prior knowledge and goals of the application.
- b. Extracting the target data set includes selecting a data set or focusing on a subset of variables.
- c. Data cleaning and preprocessing includes basic operations, such as noise removal and handling of missing data. Data from real-world sources are often erroneous, incomplete, and inconsistent, perhaps due to operation error or system implementation flaws. Such low quality data needs to be cleaned prior to data mining.
- d. Data integration includes integrating multiple, heterogeneous data sources.
- e. Data reduction and projection includes finding useful features to represent the data, depending on the goal of the task, and using dimensionality reduction or transformation methods.
- f. Choosing the function of data mining includes deciding the purpose of the model derived by the data mining algorithm.
- g. Choosing the data mining algorithm(s) includes selecting method(s) to be used for searching patterns in data, such as deciding on which model and parameters may be appropriate.
- h. Data mining includes searching for patterns of interest in a particular representational form or a set of such representations.
- i. Interpretation includes interpreting the discovered patterns, as well as the possible visualization of the extracted patterns. One can analyze the patterns automatically or semi-automatically to identify the truly interesting or useful patterns for the user.
- j. Using discovered knowledge includes incorporating this knowledge into the performance system, taking actions based on knowledge.

2.3 Data Mining

Data mining is a knowledge that made offers to new theories, techniques, and tools for processing large volumes of data. Practitioners and researchers have been focusing their attention towards data mining, as evidence by the number of publications, conferences, and application reports. It is also defined as extracting structured information, such as patterns and regularities, from database. Also known as Knowledge discovery in database (KDD), the process is important because it provides means for understanding data, including the generation of predictive rules.

Data mining actual task is the automatic or semi-automatic analysis of large quantities of data in order to extract previously unknown interesting patters such as groups of data records, unusual records, and dependencies. These patterns can then be seen as a kind of summary of the input data, and used in further analysis or for example in machine learning and predictive analytics. For example, the data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system. Neither the data collection, data preparation nor result interpretation and reporting are part of the data mining step, but do belong to the overall data mining process as additional steps.

Data mining is not a brute-force crunching of bulk data, blind application of algorithms, going to find relationships where none exist, presenting data in different ways, a database intensive task, and a difficult to understand technology requiring an advance degree in computer science, but it is a hot buzzword for a class of techniques that find patters in data, a user-centric, interactive process which leverages analysis technologies and computing power, a group of technique that find relationships that have not previously been discovered, not reliant on an existing database, and a relatively easy task that requires knowledge of the business problem or subject matter expertise.

To conduct an effective data mining, the first step to take is to examine what kind of features an applied knowledge discovery system is expected to have and what kind

of challenges one may face at the development of data mining techniques. List of them are as follows:

- a. Handling of different types of data.
- b. Efficiency and scalability of data mining algorithms.
- c. Usefulness, certainty and expressiveness of data mining results.
- d. Expression of various kinds of data mining results.
- e. Interactive mining knowledge at multiple abstraction levels.
- f. Mining information from different sources of data.
- g. Protection of privacy and data security.

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

- a. What type of databases to work on.
- b. What type of knowledge to be mined.
- c. What type of techniques to be utilized.

Methods for mining different kinds of knowledge, including association rules, characterization, classification, clustering etc. are examined in depth. For mining a particular kind of knowledge, different approaches, such as machine learning approach, statistical approach, and large database-oriented approach, are compared, with an emphasis on the database issues, such as efficiency and scalability.

2.3.1 Example of Data Mining

In KDD process, the model of association rules is important and the most representative association rules algorithm is Apriori algorithm. The objective of association rules mining is to fast discover the interesting association or related

relationship between attributes of the mass data in large-scale database. Since the association rules extraction has something to do with the source database system, then in a sense, the corresponding association rules are not generated by the direct use of DBS but by certain transformation. Therefore, in order to increase the scanning speed of the large-scale DBS and extract the association rules quickly, we must change the quantity related problems into logical related problems.

Table 1 gives a simple database example. The table mainly shows the potential associations among higher education, wages, sex, teacher status and age, and shows the future tendencies which may be lead.

Table 1: A simple example of database

RECID	SEX	AGE	KNOWLEDGE	OCCUPATION	WAGES
100	Male	46	Doctor	Teacher	7500
200	Female	32	Master	Teacher	6500
300	Male	35	Bachelor	Technician	4900
400	Male	40	Master	Teacher	6000
500	Male	37	Doctor	Teacher	7000
600	Male	25	Bachelor	Technician	4000

We do dualization for sex SEX (1:Male, 2:Female); discretization for age AGE (Age ≥ 40 ,3:middle; Age < 40 ,4:young); discretization for whether received postgraduate education KNOWLEDGE (master or doctor degree, 5: high; undergraduate diploma or low than undergraduate diploma, 6: low); dualization for occupation (college teachers, 7: teacher; not college teachers, 8:technician); dualization for wages WAGES (average monthly income is higher than 5000, 9: wages > 5000 ; 10: wages $<$

5000). Through the specifications above. Table 2 gives logical table which corresponds with Table 1.

Table 2: Logical Database corresponding with the original database

RECID	SEX		AGE		KNOWLEDGE		OCCUP ATION		WAGES	
	1	2	3	4	5	6	7	8	9	10
100	1	0	1	0	1	0	1	0	1	0
200	0	1	0	1	1	0	1	0	1	0
300	1	0	0	1	0	1	0	1	0	1
400	1	0	1	0	1	0	1	0	1	0
500	1	0	0	1	1	0	1	0	1	0
600	1	0	0	1	0	1	0	1	0	1

Using the association rules to find the potential and valuable associations between the attributes in the Table 2, we can assume that the length variable is K , namely $K=\{1, \dots, 10\}$. Meaning, our task is to retrieve the associations between occupations and wages, sex and occupations, higher education and wages and so on. These associations will provide the important associated information for personnel arrangement, occupational classification, management and distribution of wages, and so on. By database retrieval, we could get the value for each item. It is shown in the Table 3.

Table 3: Value set of attribute items in database

Recid	Items
100	1,3,5,7,9
200	2,4,5,7,9
300	1,4,6,8,10
400	1,3,5,7,9
500	1,4,5,7,9
600	1,4,6,8,10

To get association rules, assuming minimum support 0.5, namely $\min \text{sup}=0.5$, and minimum confidence 0.7, namely $\min \text{conf}=0.7$, we get K candidate sets. K larger sets and association rules. L_k stands for K larger sets.

i. Get 1 items and 1 larger sets

By scanning the database, we obtain the support of the items when the length $K=1$. Then compare the obtained support with the minimum support 0.5, we get L_1 ($K=1$ larger sets). It is shown in Table 4 below.

Table 4: K=1 Items and corresponding larger sets

Item	Sum	Sup (I)	L ₁
{1}	5	5/6	√
{2}	1	1/6	
{3}	2	2/6	
{4}	4	4/6	√
{5}	4	4/6	√
{6}	2	2/6	
{7}	4	4/6	√
{8}	2	2/6	
{9}	4	4/6	√
{10}	2	2/6	

So the corresponding larger sets when K=1 (the length is 1) are $L_1 = \{1, 4, 5, 7, 9\}$.

ii. K=2 Larger sets and association rules

We get the candidate sets when K=2 by K=1 larger sets L_1 , and then calculate the support of 2 items to get the 2 larger sets L_2 . It is shown in table 5 below.

Table 5: K=2 Items and corresponding larger sets.

Items	Sum	Sup ($I_m \cup I_n$)	L_2
{1,4}	3	3/6	√
{1,5}	3	3/6	√
{1,7}	3	3/6	√
{1,9}	3	3/6	√
{4,5}	2	2/6	
{4,7}	2	2/6	
{4,9}	2	2/6	
{5,7}	4	4/6	√
{5,9}	4	4/6	√
{7,9}	4	4/6	√

So 2 larger sets are $L_2 = \{\{1,4\}, \{1,5\}, \{1,7\}, \{1,9\}, \{5,7\}, \{5,9\}, \{7,9\}\}$. According to $conf(I_m \rightarrow I_n) = \text{Sup}(I_m \cup I_n) / \text{Sup}(I_m)$, we can get the confidence of 2 larger sets on basis of the support $\text{Sup}(A)$ of 1 larger sets. It is shown in the following Table 6.

Table 6: Confidence of K=2 Larger sets

Items	Sup($I_m \cup I_n$)	Sup(I_m)	Sup(I_n)	Conf($I_m \rightarrow I_n$)	2 association rules
{1,4}	3/6	5/6	4/6	3/5	
{1,5}	3/6	5/6	4/6	3/5	
{1,7}	3/6	5/6	4/6	3/5	
{1,9}	3/6	5/6	4/6	3/5	
{5,7}	4/6	4/6	4/6	1	√
{5,9}	4/6	4/6	4/6	1	√
{7,9}	4/6	4/6	4/6	1	√

For the minimum confidence $\text{minconf}=0.7$, so the generated association rules are $I(5) \rightarrow (7)$; $I(5) \rightarrow I(9)$; $I(7) \rightarrow (9)$.

iii. K=3 Larger sets and association rules

We can get the candidate sets when $K=3$ by $K=2$ large sets L_2 , then calculate the support of 3 items to get the 3 larger sets L_3 . It is shown in the following Table.

Table 7: K=3 Items and corresponding larger sets

Items	Sum	Sup($I_m \cup I_n \cup I_p$)	L_3
{1,4,5}	1	1/6	
{1,4,7}	1	1/6	
{1,4,9}	1	1/6	
{1,5,7}	3	3/6	√
{1,5,9}	3	3/6	√
{1,7,9}	3	3/6	√
{5,7,9}	4	4/6	√

So, the three larger sets are $L_3 = \{ \{1,5,7\}, \{1,5,9\}, \{1,7,9\} \}$. The L_3 confidence is shown in Table 8.

Table 8: K=3 larger sets confidence

Items	I_m (antecedent)	I_n (consequent)	$Sup(I_m)$	$Conf(I_m \rightarrow I_n)$	3 association rule
{1,5,7} $Sup(I_m \cup I_n) = 3/6$	1	5,7	5/6	3/5	
	5	1,7	4/6	3/4	√
	7	1,5	4/6	3/4	√
	1,5	7	3/6	1	√
	1,7	5	3/6	1	√
	5,7	1	4/6	3/4	√
{1,5,9} $Sup(I_m \cup I_n) = 3/6$	1	5,9	5/6	3/5	
	5	1,9	4/6	3/4	√
	9	1,5	4/6	3/4	√
	1,5	9	3/6	1	√
	1,9	5	3/6	1	√
	5,9	1	4/6	3/4	√
{1,7,9} $Sup(I_m \cup I_n) = 3/6$	1	7,9	5/6	3/5	
	7	1,9	4/6	3/4	√
	9	1,7	4/6	3/4	√
	1,7	9	3/6	1	√
	1,9	7	3/6	1	√
	7,9	1	4/6	3/4	√
{5,7,9} $Sup(I_m \cup I_n) = 4/6$	5	7,9	4/6	1	√
	7	5,9	4/6	1	√
	9	5,7	4/6	1	√
	5,7	9	4/6	1	√
	5,9	7	4/6	1	√
	7,9	5	4/6	1	√

For the minimum confidence is 0.7, so we can generate such association rules:

$I(5) \rightarrow I(1,7)$; $I(7) \rightarrow I(1,5)$; $I(1,5) \rightarrow I(7)$; $I(1,7) \rightarrow I(5)$; $I(5,7) \rightarrow I(1)$; $I(5) \rightarrow I(1,9)$;
 $I(9) \rightarrow I(1,5)$; $I(1,5) \rightarrow I(9)$; $I(1,9) \rightarrow I(5)$; $I(5,9) \rightarrow I(1)$; $I(7) \rightarrow I(1,9)$; $I(9) \rightarrow I(1,7)$;
 $I(1,7) \rightarrow I(9)$; $I(1,9) \rightarrow I(7)$; $I(7,9) \rightarrow I(1)$; $I(5) \rightarrow I(7,9)$; $I(7) \rightarrow I(5,9)$; $I(9) \rightarrow I(5,7)$;
 $I(5,7) \rightarrow I(9)$; $I(5,9) \rightarrow I(7)$; $I(7,9) \rightarrow I(5)$

iv. K=4 Larger sets and association rules

Because the 3 larger sets $L_3 = \{\{1,5,7\}, \{1,5,9\}, \{1,7,9\}, \{1,7,9\}, \{5,7,9\}\}$, we know there is only one item $\{1,5,7,9\}$ in the 4 items. It is shown in Table 9 below.

Table 9: K=4 items and corresponding larger sets

Items	Sum	Sup($I_m \cup I_n \cup I_p$)	L_4
{1,5,7,9}	3	3/6	√

Table 10 gives the confidence of $\{1,5,7,9\}$ which is the K=4 larger sets.

Table 10: Confidence of {1,5,7,9} 4 larger sets

Items	I_m (ante cedent)	I_n (conse quent)	Sup (I_m)	Conf ($I_m \rightarrow I_n$)	4 association rules
{1,5,7,9}	1	5,7,9	5/6	3/5	
Sup($I_m \cup I_n$) =3/6	5	1,7,9	4/6	3/4	√
	7	1,5,9	4/6	3/4	√
	9	1,5,7	4/6	3/4	√
	1,5	7,9	3/6	1	√
	1,7	5,9	3/6	1	√
	1,9	5,7	3/6	1	√
	5,7	1,9	4/6	3/4	√
	5,9	1,7	4/6	3/4	√
	7,9	1,5	4/6	3/4	√
	1,5,7	9	4/6	3/4	√
	1,5,9	7	3/6	1	√
	1,7,9	5	3/6	1	√
	5,7,9	1	4/6	3/4	√

For the minimum confidence is assumed as 0.7, so we obtain the association rules as follows:

$I(5) \rightarrow I(1,7,9)$; $I(7) \rightarrow I(1,5,9)$; $I(9) \rightarrow I(1,5,7)$; $I(1,5) \rightarrow I(7,9)$;

$I(1,7) \rightarrow I(5,9)$; $I(1,9) \rightarrow I(5,7)$; $I(5,7) \rightarrow I(1,9)$; $I(5,9) \rightarrow I(1,7)$;

$I(7,9) \rightarrow I(1,5)$; $I(1,5,7) \rightarrow I(9)$; $I(1,5,9) \rightarrow I(7)$; $I(1,7,9) \rightarrow I(5)$; $I(5,7,9) \rightarrow I(1)$

For the last one, we need to make the explanation and visualization for the retrieved association rules. In other words, we need to give back the data of specified discretization to the original meaning, execute the explanation so that users who use

the association rules know the accurate meanings of the conclusions which are calculated above.

We illustrate part of the obtained association rules.

- (1) $I(7) \rightarrow I(9)$ means: in the level of the minimum support 0.5 and the minimum confidence 0.7, a college teacher \rightarrow wages are more than 5000 RMB.
- (2) $I(5) \rightarrow I(1,7)$ means: in the level of the minimum support 0.5 and the minimum confidence 0.7, a Doctor or Master degree \rightarrow sex is male and can become a college teacher.
- (3) $I(1,5,7) \rightarrow I(9)$ means: in the level of the minimum support 0.5 and the minimum confidence 0.7, sex male and a Doctor or Master degree and a college teacher \rightarrow wages are more than 5000 RMB.

From the running results above, we can illustrate the following potential relevance between the higher education and wages, occupation and wages, higher education and sex, university teachers and sex, high wages and education, and higher education and ages, and so on. We can use these results to analyze the future tendencies and potential associations that may be lead. It is states that “if the distribution of the attributes is reasonable, we will have sufficient theoretical basis to develop our causes if not, then we can take measures to solve these problems timely” (Yan Chen, Ming Yang, Lin Zhang, 2009).

2.3.2 Application of Data Mining in computer science fields

In computer science field, data mining is the process of discovering interesting and useful patterns and relationships in large volumes of data. The filed combines tools from statistics and artificial intelligence with database management to analyze large digital collections, known as data sets. Data mining is also often used in the field of business, science research and government security.

Since the 80's, there have been an increased in computer storage thus making many companies began to store more transactional data. The resulting record collections also known as data warehouses, were too large to be analyzed with traditional statistical approaches. Several computer science conferences and workshops were held to consider how recent advances in the field of artificial intelligences (AI) such as discoveries from expert systems, genetic algorithms, machine learning, and neural network, could be adapted for knowledge discovery. In 1995, this process has led to the First International Conference on Knowledge Discovery and Data Mining.

One of the earliest successful applications of data mining was credit-card fraud detection. By observing a consumer's purchasing behavior, a typical pattern usually becomes apparent and purchase made outside this pattern can then be marked for later investigation or to cancel a transaction. However, the wide variety of normal behaviors makes this challenging because there is no single distinction between normal and fraudulent behavior works for everyone or all the time. Every individual is likely to make some purchases that will be different from the types he or she has made before, thus making the method of relying on what is normal for a single individual is likely to give too many false alarms. One of the effort to improving reliability is first to group individuals that have similar purchasing patterns, since group models are less sensitive to minor anomalies. For example, a "frequent business travelers" group will likely have a pattern that includes unprecedented purchases in diverse locations, but members of this group might be flagged for other transactions, such as catalog purchases, that do not fit that group's profile. (<http://www.britannica.com/EBchecked/topic/1056150/data-mining>)

2.4 Data Clustering

In data mining process, clustering is useful for discovering groups and identifying interesting distributions and patterns in the underlying data. The problem in clustering is about partitioning a given data set into groups such that the data points in a cluster are more similar to each other than points in different clusters. 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 and engineering. Clustering also have several different names that can be found under different contexts, such as unsupervised learning, numerical taxonomy , typology and partition. There are no predefined classes in clustering and no examples that would show what kind of desirable relations should be valid among the data that is why it is perceived as an unsupervised process. On the other hand, classification is a procedure of assigning a data item to a predefined set of categories. Clustering produces initial categories in which values of a data set are classified during the classification process.(M. Halkidi, Y Batistakis, M.Vazirgiannis, 2001)

Clustering is very useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations; including data mining, document retrieval, image segmentation, and pattern classification. However, in many such problems, there is little prior information available about the data, and the decision-maker must make as few assumptions about the data as possible. It is because of this restrictions that clustering methodology is not really the best method for the exploration of interrelationships among the data points to make an assessment (perhaps preliminary) of their structure.(A.K. Jain, M.N. Murty, P.J. Flynn, 1999)

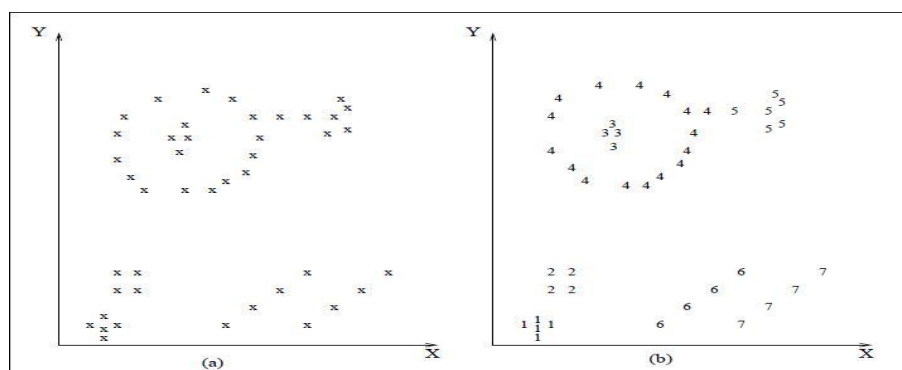


Figure 3: Data Clustering(A.K. Jain, M.N. Murty, P.J. Flynn, 1999)

2.4.1 Classification vs Clustering

Both classification and clustering characterized objects by one or more features. Classification make labels for some points, and also have a rule that will accurately assign labels to new points. Classification is also a supervised learning, while clustering is an unsupervised learning. Clustering also has no labels for the points. It group points into clusters based on how close they are to one another and it also identify structure in data.(OCW MIT OpenCourseWare, 2008)

Classification

Classification is one of data mining function that make an items in a collection to target categories or classes. The function of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

A classification task is initiate with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on studied data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments.. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.

Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical target. A predictive model with a numerical target uses a regression algorithm instead of classification algorithm.

The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high

credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating.

In the model build process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different techniques are used in finding relationship for each different classification. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

To test the classification models, the predicted values are compared to known target in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model.

Scoring a classification model results in class assignments and probabilities for each case. For example, a model that classifies customers as low, medium, or high value would also predict the probability of each classification for each customer.

Classification has many applications in customer segmentation, business modeling, marketing, credit analysis, and biomedical and drug response modeling. (http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/classify.htm)

Clustering

Clustering analysis finds clusters of data objects that are similar in some sense to one another. The members of a cluster are similar to each other than they are to the members of other clusters. The goal of clustering analysis is to find high-quality clusters such that the inter-cluster similarity is low and the intra-cluster similarity is high.

Clustering is used to segment the data which is same as classification, but unlike classification, clustering models segment data into groups that were not previously

defined. Classification models segment data by assigning it to previously-defined classes, which are specified in a target. Clustering models do not use a target.

Clustering is useful for exploring data. If there are many cases and no obvious groupings, clustering algorithms can be used to find natural groupings. Clustering can also serve as a useful data-preprocessing step to identify homogeneous groups on which to build supervised models.

Clustering can also be used for anomaly detection. Once the data has been segmented into clusters, you might find that some cases do not fit well into any clusters. These cases are anomalies or outliers. (http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/clustering.htm)

2.4.2 Clustering Techniques

There several methods of algorithm that can be used to cluster data, that are Hierarchical Clustering Algorithms, Partitional Algorithms, Mixture-Resolving and Mode-Seeking Algorithms, Nearest Neighbor Clustering, and Fuzzy Clustering.

Hierarchical Clustering Algorithm

Hierarchical clustering solutions which is in the form of trees called *dendrograms* are of great interest for a number of application domains. Hierarchical trees provide a view of the data at different levels of abstraction. The consistency of clustering solutions at different levels of granularity allows flat partitions of different granularity to be extracted during data analysis, making them ideal for interactive exploration and visualization. In addition, there are many times when clusters have subclusters, and hierarchical structures represent the underlying application domain naturally.

Hierarchical clustering solutions have been primarily obtained using agglomerative algorithms in which objects are initially assigned to their own cluster and then pairs of clusters are repeatedly merged until the whole tree is formed. However, partitional can also be used to obtain hierarchical clustering solutions via a sequence of repeated bisections. In recent years, various researchers have recognized that partitional clustering algorithms are well-suited for clustering large document datasets due to their relatively low computational requirements. Nevertheless, there is the common belief that in terms of clustering quality, partitional algorithms are actually inferior and less effective than their agglomerative counterparts.

This belief is based both on experiments with low dimensional datasets as well as a limited number of studies in which agglomerative approaches in general outperformed partitional K -means based approaches. For this reason, existing reviews of hierarchical document clustering methods focused mainly on agglomerative methods and entirely ignored partitional methods.

In addition, most of the previous studies evaluated various clustering methods by how well the resulting clustering solutions can improve retrieval. The comparisons in terms of how well the resulting hierarchical trees are consistent with the existing class information are limited and only based on very few datasets.(Ying Zhao, G. Karypis, 2003)

Partitional Algorithms

A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure. Partitional methods have advantages in applications involving large data sets for which the construction of a dendrograms is computationally prohibitive. The one problem when using this algorithms is the use of a partitional algorithm is the choice of the number of desired output clusters. The partitional techniques usually produce clusters by optimizing a criterion function defined either locally (on a subset of the patterns) or globally (defined over all of the patterns). Combinatorial search of the set of possible labeling for an optimum value of a

criterion is clearly computationally prohibitive. Therefore, usually the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs is used as the output clustering. (A.K. Jain, M.N. Murty, P.J. Flynn, 1999)

Mixture-Resolving and Mode-Seeking Algorithms

The mixture resolving approach to cluster analysis has been addressed in a number of ways. The underlying assumption is that the patterns to be clustered are drawn from one of several distributions, and the goal is to identify the parameters of each and their number. Most of the work in this area has assumed that the individual components of the mixture density are Gaussian, and in this case the parameters of the individual Gaussians are to be estimated by the procedure. More recently, the Expectation Maximization (EM) has been applied to the problem of parameter estimation. The EM procedure begins with an initial estimate of the parameter vector and iteratively rescores the patterns against the mixture density produced by the parameter vector. The recorded patterns are then used to update the parameter estimates. In a clustering context, the scores of the patterns can be viewed as hints at the class of the pattern. Those patterns, placed by their scores in a particular component, would therefore be viewed as belonging to the same cluster. Nonparametric techniques for density-based clustering have also been developed [Jain and Dubes 1988]. Inspired by the Parzen window approach to nonparametric density estimation, the corresponding clustering procedure searches for bins with large counts in a multidimensional histogram of the input pattern set. Other approaches include the application of another partitional or hierarchical clustering algorithm using a distance measure based on a nonparametric density estimate. (A.K. Jain, M.N. Murty, P.J. Flynn, 1999)

Nearest Neighbor Clustering

Since proximity plays a key role in our intuitive notion of a cluster, nearest neighbor distances can serve as the basis of clustering procedures. An iterative procedure was

proposed in Lu and Fu [1978]; it assigns each unlabeled pattern to the cluster of its nearest labeled neighbor pattern, provided the distance to that labeled neighbor is below a threshold. The process continues until all patterns are labeled or no additional labeling occur. The mutual neighborhood can also be used to grow clusters from near neighbors.(A.K. Jain, M.N. Murty, P.J. Flynn, 1999)

Fuzzy Clustering

Traditional clustering approaches generate partitions; in a partition, each pattern belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern with every cluster using a membership function [Zadeh 1965]. The output of such algorithms is a clustering, but not a partition.(A.K. Jain, M.N. Murty, P.J. Flynn, 1999)

In **fuzzy clustering**, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. An overview and comparison of different fuzzy clustering algorithms is available.

Any point x has a set of coefficients giving the degree of being in the k th cluster $w_k(x)$. With fuzzy c -means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$c_k = \frac{\sum_x w_k(x)x}{\sum_x w_k(x)}.$$

The degree of belonging, $w_k(x)$, is related inversely to the distance from x to the cluster center as calculated on the previous pass. It also depends on a parameter m that controls how much weight is given to the closest center. The fuzzy c -means algorithm is very similar to the k -means algorithm:

- a. Choose a number of clusters.

- b. Assign randomly to each point coefficients for being in the clusters.
- c. Repeat until the algorithm has converged:
 - a. Compute the centroid for each cluster, using the formula above.
 - b. For each point, compute its coefficients of being in the clusters, using the formula above.

The algorithm minimizes intra-cluster variance as well, but has the same problems as k -means; the minimum is a local minimum, and the results depend on the initial choice of weights.

The expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes.

Fuzzy c-means has been a very important tool for image processing in clustering objects in an image. In the 70's, mathematicians introduced the spatial term into the FCM algorithm to improve the accuracy of clustering under noise. (http://en.wikipedia.org/wiki/Fuzzy_clustering)

2.4.3 Clustering on Numerical Dataset

Clustering analysis consist of polythetic methods that use all attributes, agglomerative methods that begin with all cases being singleton cluster and in divisive methods which initially all cases are placed in one cluster. Different numerical attributes were standardized by dividing the attribute values by the corresponding attributes standard deviation. Cluster formation was started by computing a distance matrix between every cluster. By using agglomerative discretization method, new clusters were formed by merging two existing clusters that were the closest to each other. When such a pair was founded (cluster b and c), they were fused to form a new cluster d . Definability of Sets (let $B \subseteq A$), a union of some intersections of attributes-value pair blocks, attributes are members of B , will be called a B-locally definable set. (Jerzy W. Grzymala-Busse, 2007)

2.4.4 Clustering on Categorical Dataset

Categorical data have no single ordering, there are several ways in which they can be ordered, but there is no single one which is more semantically sensible than others. It also can be visualized depending on a specific ordering and it define a priori structure to work with. Other than that, categorical data can be mapped onto unique numbers and, as a consequences, Euclidean distance could be used to prescribe their proximities, with questionable consequences though.

One of the algorithm in the database community, oriented towards categorical data sets is an extension to *k-means*, called *k-modes* [Hua98]. The idea is the same as in *k-means* and the structure of the algorithm does not change. The only difference is in the similarity measure used to compare the data objects.

More specifically the differences are that a different dissimilarity measure is used, the *means* are replaced by *modes*, and a frequency based methods is used to update modes.

Given two categorical data objects x and y , their dissimilarity is found using the following expression:

$$d(\hat{x}, \hat{y}) = \sum_{i=1}^n \delta(x_i, y_i)$$

$$\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases}$$

Intuitively, the above expression counts the number of mis-matches the two data objects have on their corresponding attributes. Note that every attribute is given the same weight. If we consider the frequencies of the values in a data set, then the dissimilarity expression becomes:

$$d(\hat{x}, \hat{y}) = \sum_{i=1}^n \frac{n_{x_i} + n_{y_i}}{n_{x_i} n_{y_i}} \delta(x_i, y_i)$$

Where n_x , and n_y are the numbers of objects in the data set with attributes values x and y , for attribute i , respectively. The mode of a set is the value that appears the most in this set. For a data set of dimensionality n , every cluster c , $1 \leq c \leq k$, has a mode defined by a vector $Q^c = (x_1^c, x_2^c, \dots, x_n^c)$. The set of Q^c 's that minimize the expression:

$$E = \sum_{c=1}^k \sum_{\hat{x} \in c} d(\hat{x}, Q^c)$$

The similarities, in structure and behavior, with *k-means* are obvious, with *k-modes* carrying, unfortunately, all the disadvantages of the former. An interesting extension to data sets of both numerical and categorical attributes is that of *k-prototypes*. It is an integration of *k-means* and *k-modes* employing:

- s^r : dissimilarity on numeric attributes;
- s^c : dissimilarity on categorical attributes;
- dissimilarity measure between two objects:

$$s^r + \gamma s^c$$

where γ is a weight to balance the two parts and avoid favoring either type of attribute. γ is a parameter specified by the user. (P. Andritos, 2002)

2.4.5 Applications of Clustering Techniques

Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters. Clustering has its roots in many areas,

including Datamining, statistics, biology, and machine learning. Clustering is a type of multivariate statistical analysis also known as cluster analysis, unsupervised classification analysis, or numerical taxonomy. Cluster analysis is based on a mathematical formulation of a measure of similarity. In other words cluster analysis is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. In other words, cluster analysis simply discovers structures in data without explaining why they exist. Clustering is an excellent example of an unsupervised learning technique and we cannot observe the (real) number of clusters in the data. However, it is reasonable to replace the usual notion (applicable to supervised learning) of "accuracy" with that of "distance." In general, we can apply the v-fold cross-validation method to a range of numbers of clusters in K-Means clustering, and observe the resulting average distance of the observations (in the cross-validation or testing samples) from their cluster centers (for K-Means clustering). L. Shen et al. described the stock prediction for use of investors. In his paper, the original decision table together with a new decision attribute obtained by Self-Organizing Maps (SOM) is reconstructed. The SOM is applied as a cluster method. Questier et al. described the uses of rough set theory to construct the reducts in a supervised way for reducing the number of features in an unsupervised clustering. S. Susanto et.al developed a new approach for the allocation of the students using fuzzy clustering algorithm. T. Maciag et al. applied K-Means Clustering and Rough Set Exploration System (RSES) for feature selection and decision making.(K . Thangavel, Qiang Shen, A. Pethalakshmi, 2006)

2.5 Rough Set Theory

The idea of rough set was proposed by Pawlak (1981) as a new mathematical tool to deal with vague concepts. Comer, Grzymala-Busse, Iwinski, Nieminen, Novotny, Pawlak, Obtulowicz, and Pomykala have studied algebraic properties of rough sets. Different algebraic semantics have been developed by P. Pagliani, I. Duntsch, M. K.

Chakraborty, M. Banerjee and A. Mani; these have been extended to more generalized rough sets by D. Cattaneo and A. Mani, in particular. Rough sets can be used to represent ambiguity, vagueness and general uncertainty. Fuzzy-rough sets further extend the rough set concept through the use of fuzzy equivalence classes. (http://en.wikipedia.org/wiki/Rough_set#History)

2.5.1 Fuzzy set

Fuzzy sets are sets whose elements have degrees of membership. Fuzzy sets were introduced simultaneously by Lotfi A. Zadeh and Dieter Klaua in 1965 as an extension of the classical notion of set. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition — an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval $[0, 1]$. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1. In fuzzy set theory, classical bivalent sets are usually called crisp sets. The fuzzy set theory can be used in a wide range of domains in which information is incomplete or imprecise, such as bioinformatics.

Fuzzy sets can be applied, for example, to the field of genealogical research. When an individual is searching in vital records such as birth records for possible ancestors, the researcher must contend with a number of issues that could be encapsulated in a membership function. Looking for an ancestor named John Henry Pittman, who you think was born in (probably eastern) Tennessee circa 1853 (based on statements of his age in later censuses, and a marriage record in Knoxville), what is the likelihood that a particular birth record for "John Pittman" is your John Pittman? What about a record in a different part of Tennessee for "J.H. Pittman" in 1851? (It has been suggested by Thayer Watkins that Zadeh's ethnicity is an example of a fuzzy set). (http://en.wikipedia.org/wiki/Fuzzy_set)

2.5.2 Relation between fuzzy and rough set theories

Theories of fuzzy sets and rough sets are generalizations of classical set theory for modeling vagueness and uncertainty. A fundamental question concerning both theories is their connections and differences. There have been many studies on this topic. While some authors argued that one theory is more general than the other, it is generally accepted that they are related but distinct and complementary theories. The two theories model different types of uncertainty. The rough set theory takes into consideration the indiscernibility between objects. The indiscernibility is typically characterized by an equivalence relation. Rough sets are the results of approximating crisp sets using equivalence classes. The fuzzy set theory deals with the ill-definition of the boundary of a class through a continuous generalization of set characteristic functions. The indiscernibility between objects is not used in fuzzy set theory. A fuzzy set may be viewed as a class with unsharp boundaries, whereas a rough set is a crisp set which is coarsely described.

Klir compared the roles played by non-classical logics, such as many-valued logics and modal logics, for interpreting fuzzy sets and rough sets. According to Haack, a non-classical logic is a deviation of classical two-valued logic, i.e., a deviant logic, if the two logics have the same logical vocabulary but different axioms or rules. Many-valued logics may be viewed as deviant logics. A non-classical logic is an extension, i.e., an extended logic, if it adds new vocabulary along with new axioms or rules for the new vocabulary. Modal logics may be viewed as extended logics. Classical set-theoretic operators reflect the

corresponding logic connectives in classical two-valued logic. Similar correspondence may also be established between non-classical set-theoretic operators and non-classical logic connectives. Non-classical set theories may therefore be similarly viewed as deviations and extensions of classical set theory. From such a point of view, this paper presents a comparative study of theories of fuzzy sets and rough sets.

As pointed out recently by Zadeh, fuzzy logic has many facets: the logical facet, the set-theoretic facet, the relational facet, and the epistemic facet. Each of these facets

may be further divided. In the same way, there are many different formulations and interpretations of the theory of rough sets. It is very important to realize that our comparisons of two theories are based on very specific interpretations of each theory. Furthermore, many issues involved in both theories are not taken into consideration. Although conclusions drawn from such comparisons should be read cautiously, the examination may provide more insights into both theories.(Y.Y. Yao, 1998)

2.5.3 Application of rough set

The Rough Sets Theory has been used effectively to handle efficiently problems where large amounts of data are produced. Rough Sets theory constitutes a framework for inducing minimal decision rules. These rules, in turn, can be used to perform a classification task. The main goal of the rough set analysis is to search large databases for meaningful decision rules and finally acquire new knowledge. This approach is based on four main topics: indiscernibility, approximation, reducts and decision rules. A reduct is a minimal set of attributes from the whole attributes set that preserves the partitioning of the finite set of objects and therefore the original classes. It means that the redundant attributes are eliminated. When the reducts are found, the task of creating definite rules for the value of the decision attribute of the information system is practically performed. Decision rules are generated combining the attributes of the reducts with the values. Decision rules extract knowledge, which can be used when classifying new objects, not in the original information system.(C.I.F Agreira, C.M.M. Ferreira, F.P.M. Barbosa, 2010)

2.5.4 Rough clustering

In 1982, Pawlak introduced the theory of Rough sets. This theory was initially developed for a finite universe of discourse in which the knowledge base is a partition, which is obtained by any equivalence relation defined on the universe of discourse. In rough sets theory, the data is organized in a table called decision table.

Rows of the decision table correspond to objects, and columns correspond to attributes. In the data set, a class label to indicate the class to which each row belongs. The class label is called as decision attribute, the rest of the attributes are the condition attributes. Here, C is used to denote the condition attributes, D for decision attributes, where $C \cap D = \Phi$, and t_j denotes the j th tuple of the data table. Rough sets theory defines three regions based on the equivalent classes induced by the attribute values: lower approximation, upper approximation, and boundary. Lower approximation contains all the objects, which are classified surely based on the data collected, and Upper approximation contains all the objects, which can be classified probably, while the boundary is the difference between the upper approximation and the lower approximation. Hu et al., presented the formal definitions of rough set theory. A.Kusiak (<http://www.britannica.com/EBchecked/topic/1056150/data-mining>) described the basic concepts of rough set theory, and other aspects of Data mining. The other aspects of data mining are Equivalence classes, Atoms, Approximation accuracy, Boundary approximation, Classification accuracy, Classification quality, Sensitivity, Specificity, Positive predicted value, Negative predicted value, Rule length, Rule strength, Exact rule, Approximate rule, Rule support, Rule coverage, Rule acceptance and Discrimination level.(K . Thangavel, Qiang Shen, A. Pethalakshmi, 2006)

2.5.5 Rough set theory in categorical data clustering

Data clustering is a popular data analysis task that involves the distribution of ‘unannotated’ data (i.e. with no a priori class information), in an inductive manner, into a finite sets of categories or cluster such that data items within a cluster are similar in some respect and unlike those from other cluster. If one regards data as an underlying quantitative statement about a system’s behavior-either human or engineered-within a particular environment, then exploratory data clustering algorithms attempt to learn the topology of the data by analyzing the inherent similarities and differences of the individual data items in the untagged data set.

Notwithstanding the efficacy of traditional data clustering techniques, it can be argued that the outcome of a data clustering task does not necessarily explicate the intrinsic relationship between the various attributes of the dataset. Given an un-annotated dataset satisfying the above assumption, we first partition it into k cluster, where each cluster comprises data-vectors with similar inherent characteristic. Note that the data clustering task is carried out with no a priori knowledge about the intrinsic class structure-i.e. how the data is inherently partitioned into distinct clusters. This is the phase 1 then next phase is data discretisation. The motivation for thus phase is driven by the fact the ordinal or continuous valued attributes are proven to be rather unsuitable for the extraction of concise symbolic rules. Last phase is symbolic rule discovery. We use rough set approximation-an interesting alternative to a variety of symbolic rule methods to derive symbolic rules that explain the inherent dependencies, attribute significance and structural characteristic of the annotated and clustered data-set.(Syed Sibte Raza Abidi, Kok Meng Hoe, Alwyn Goh)

CHAPTER III

METHODOLOGY

This chapter briefly discusses on the method and the procedures of clustering data using Rough Set Theory by using Maximum Dependency of Attributes Technique.

3.1 Rough Set Theory

The problem of imprecise knowledge has been tackled for a long time by mathematicians. Recently it became a crucial issue for computer scientists, particularly in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate imprecise knowledge. The most successful one is, no doubt, the fuzzy set theory proposed by Zadeh (Zadeh, 1965). The basic tools of the theory are possibility measures. There is extensive literature on fuzzy logic with also discusses some of the problem with this theory. The basic problem of fuzzy set theory is the determination of the grade of membership of the value of possibility (Busse, 1998).

In the 1980's, Pawlak introduced rough set theory to deal this problem (Pawlak, 1982). Similarly to rough set theory it is not an alternative to classical set theory but it is embedded in it. Fuzzy and rough sets theories are not competitive, but complementary to each other (Pawlak and Skowron, 2007; Pawlak, 1985). Rough set theory has attracted attention to many researchers and practitioners all over the world, who contributed essentially to its development and applications. The original goal of the rough set theory is induction of approximations of concepts. The idea consists of approximation of a subset by a pair of two precise concepts called the *lower approximation* and *upper approximation*. Intuitively, the lower approximation

of a set consists of all elements that surely belong to the set, whereas the upper approximation of the set constitutes of all elements that possibly belong to the set. The difference of the upper approximation and the lower approximation is a *boundary region*. It consists of all elements that cannot be classified uniquely to the set or its complement, by employing available knowledge. Thus any rough set, in contrast to a crisp set, has a non-empty boundary region. Motivation for rough set theory has come from the need to represent a subset of a universe in terms of equivalence classes of a partition of the universe.

3.1.1 Information System

Data are often presented as a table, columns of which are labeled by *attributes*, rows by *objects* of interest and entries of the table are *attribute values*. By an *information system*, we mean a 4-tuple (quadruple) $S = (U, A, V, f)$, where U is a non-empty finite set of objects, A is a non-empty finite set of attributes, $V = \bigcup_{a \in A} V_a$, V_a is the domain (value set) of attribute a , $f : U \times A \rightarrow V$ is a total function such that $f(u, a) \in V_a$, for every $(u, a) \in U \times A$, called information (knowledge) function. An information system is also called a knowledge representation systems or an attribute-valued system and can be intuitively expressed in terms of an information table (refer to Table 11).

Table 11: An information system

U	a_1	a_2	\dots	a_k	\dots	$a_{ A }$
u_1	$f(u_1, a_1)$	$f(u_1, a_2)$	\dots	$f(u_1, a_k)$	\dots	$f(u_1, a_{ A })$
u_2	$f(u_2, a_1)$	$f(u_2, a_2)$	\dots	$f(u_2, a_k)$	\dots	$f(u_2, a_{ A })$
\vdots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
$u_{ U }$	$f(u_{ U }, a_1)$	$f(u_{ U }, a_2)$	\dots	$f(u_{ U }, a_k)$	\dots	$f(u_{ U }, a_{ A })$

Example 3.1: Suppose that data about 10 mushrooms is given, as shown in Table 12.

Table 12: A mushrooms decision system

Mushroom	Cap- Shape	Cap- Color	Odor	Gill- Spacing	Gill- Size	Gill- Color	Stalk- Shape	Ring- Type
1	convex	brown	pungent	close	narrow	black	enlarging	pendant
2	convex	yellow	almond	close	broad	black	enlarging	pendant
3	bell	White	anise	close	broad	brown	enlarging	pendant
4	convex	White	pungent	close	narrow	pink	enlarging	pendant
5	convex	Gray	none	crowded	broad	black	tapering	evanescent
6	convex	yellow	almond	close	broad	brown	enlarging	pendant
7	bell	White	almond	close	broad	gray	enlarging	pendant
8	bell	White	anise	close	broad	brown	enlarging	pendant
9	convex	White	pungent	close	narrow	pink	enlarging	pendant
10	bell	yellow	almond	close	broad	gray	enlarging	pendant

The following values are obtained from Table 12,

$$U = \{1,2,3,4,5,6,7,8,9,10\},$$

$$A = \left\{ \begin{array}{l} \text{Cap - Shape, Cap - Color, Odor, Gill - Spacing, Gill - Size, Gill - Color,} \\ \text{Stalk - Shape, Veil - Type, Ring - Number, Ring - Type, Classes} \end{array} \right\}$$

$$V_{\text{Cap-Shape}} = \{\text{convex, bell}\},$$

$$V_{\text{Cap-Color}} = \{\text{brown, yellow, white, grey}\},$$

$$V_{\text{Odor}} = \{\text{pungent, almond, anise, none}\},$$

$$V_{\text{Gill-Spacing}} = \{\text{close, crowded}\}.$$

$$V_{\text{Gill-Size}} = \{\text{narrow, broad}\}$$

$$V_{\text{Gill-Color}} = \{\text{black, brown, grey, pink}\}$$

$$V_{\text{Stalk-Shape}} = \{\text{enlarging, tapering}\}$$

$$V_{\text{Ring-Type}} = \{\text{pendant, evanescent}\}$$

In many applications, there is an outcome of classification that is known. This *a posteriori* knowledge is expressed by one (or more) distinguished attribute called decision attribute; the process is known as supervised learning. An information system of this kind is called a decision system. A *decision system* is an information system of the form $D = (U, A = C \cup D, V, f)$, where D is the set of *decision attributes* and $C \cap D = \emptyset$. The elements of C are called *condition attributes*. A simple example of decision system is given in Table 12

Example 3.2: Suppose that data about 9 bananas is given, as shown in Table 13.

Table 13: Data of bananas

U/E	SIZE	SMELL	COLOR	FIELD	CLASSES
X ₁	15	Good	Yellow	#1	FRUIT
X ₂	15	Good	Yellow	#1	FRUIT
X ₃	14	Good	Yellow	#1	FRUIT
X ₄	7	Good	Yellow	#1	JUICE
X ₅	12	Bad	Yellow	#1	JUICE
X ₆	13	Good	Brown	#2	JUICE
X ₇	12	Bad	Brown	#1	REJECT
X ₈	14	Bad	Black	#2	REJECT
X ₉	15	Bad	Black	#2	REJECT

A relational database may be considered as an information system in which rows are labeled by the objects (entities), columns are labeled by attributes and the entry in row u and column a has the value $f(u, a)$. It is noted that each map

$f(u, a): U \times A \rightarrow V$ is a tuple $t_i = (f(u_i, a_1), f(u_i, a_2), f(u_i, a_3), \dots, f(u_i, a_{|A|}))$, for $1 \leq i \leq |U|$, where $|X|$ is the cardinality of X . Note that the tuple t is not necessarily associated with entity uniquely (refers to mushrooms4 and 9 in Table 12). In an information table, two distinct entities could have the same tuple representation (duplicated/redundant tuple), which is *not permissible* in relational databases. Thus, the concepts in information systems are a generalization of the same concepts in relational databases.

3.1.2 Indiscernibility Relation

From table 12, it is noted that mushrooms 4 and 9 are indiscernible with respect to the attribute Cap-Shape, Cap-Color, Odor, Gill-Spacing, Gill-Size, Gill-Color, Veil-Type, Ring-Number, and Ring-Type. Meanwhile mushrooms 2, 6, 7, and 10 are indiscernible with respect to the attribute Odor, Gill-Spacing, Gill-Size, Stalk-Type, Veil-Type, Ring-Number, Ring-Type, and Classes. All mushrooms are indiscernible with respect to attribute Veil-Type and Ring-Number. Mushrooms that are indiscernible with respect to attribute Cap-Shape and Cap-Color are mushrooms 3, 7, and 8.

The starting point of rough set theory is the indiscernibility relation, which is generated by information about objects of interest. The indiscernibility relation is intended to express the fact that due to the lack of knowledge we are unable to discern some objects employing the available information. Therefore, generally, we are unable to deal with single object. Nevertheless, we have to consider clusters of indiscernible objects. The following definition precisely defines the notion of indiscernibility relation between two objects.

Definition 2.1. Let $S = (U, A, V, f)$ be an information system and let B be any subset of A . Two elements $x, y \in U$ are said to be B -indiscernible (indiscernible by the set of attribute $B \subseteq A$ in S) if and only if $f(x, a) = f(y, a)$, for every $a \in B$.

Obviously, every subset of A induces unique indiscernibility relation. Notice that, an indiscernibility relation induced by the set of attribute B , denoted by $IND(B)$, is an equivalence relation. It is well known that, an equivalence relation induces unique partition. The partition of U induced by $IND(B)$ in $S = (U, A, V, f)$ denoted by U/B and the equivalence class in the partition U/B containing $x \in U$, denoted by $[x]_B$.

3.1.3 Set Approximations

The indiscernibility relation will be used to define set approximations that are the basic concepts of rough set theory. The notions of lower and upper approximations of a set can be defined as follows.

Definition 2.3. Let $S = (U, A, V, f)$ be an information system, let B be any subset of A and let X be any subset of U . The B -lower approximation of X , denoted by $\underline{B}(X)$ and B -upper approximations of X , denoted by $\overline{B}(X)$, respectively, are defined by

$$\underline{B}(X) = \{x \in U \mid [x]_B \subseteq X\} \text{ and } \overline{B}(X) = \{x \in U \mid [x]_B \cap X \neq \emptyset\}.$$

From Definition 2.3, the following interpretations are obtained

- a. The *lower approximation* of a set X with respect to B is the set of all objects, which can be for *certain* classified as X using B (are certainly X in view of B).
- b. The *upper approximation* of a set X with respect to B is the set of all objects which can be *possibly* classified as X using B (are possibly X in view of B).

Hence, with respect to arbitrary subset $X \subseteq U$, the universe U can be divided into three disjoint regions using the lower and upper approximations

- a. The *positive region* $\text{POS}_B(X) = \underline{B}(X)$, i.e., the set of all objects, which can be for *certain* classified as X using B (are *certainly* X with respect to B).
- b. The *boundary region* $\text{BND}_B(X) = \overline{B}(X) - \underline{B}(X)$, i.e., the set of all objects, which can be classified neither as X nor as not- X using B .
- c. The *negative region* $\text{NEG}_B(X) = U - \overline{B}(X)$, i.e., the set of all objects, which can be for *certain* classified as not- X using B (are *certainly* not- X with respect to B).

These notions of lower and upper approximations can be shown clearly as in Figure 4.

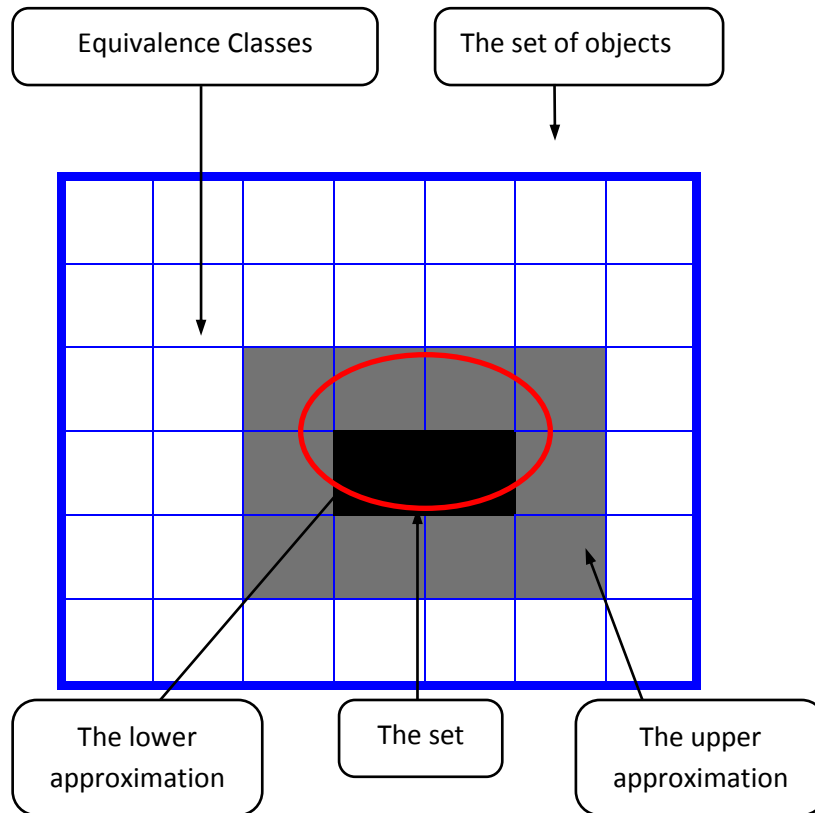





Figure 4: Set approximations

From Figure 4, three disjoint regions are given as follows

- a. The positive region 
- b. The boundary region 
- c. The negative region 

The accuracy of approximation (accuracy of roughness) of any subset $X \subseteq U$ with respect to $B \subseteq A$, denoted $\alpha_B(X)$ is measured by

$$\alpha_B(X) = \frac{|B(X)|}{|B(X)|}, \quad (2.1)$$

where $|X|$ denotes the cardinality of X . For empty set ϕ , it is defined that $\alpha_B(\phi) = 1$ (Pawlak and Skowron, 2007). Obviously, $0 \leq \alpha_B(X) \leq 1$. If X is a union of some equivalence classes of U , then $\alpha_B(X) = 1$. Thus, the set X is *crisp* (precise) with respect to B . And, if X is not a union of some equivalence classes of U , then $\alpha_B(X) < 1$. Thus, the set X is *rough* (imprecise) with respect to B (Pawlak and Skowron, 2007). This means that the higher of accuracy of approximation of any subset $X \subseteq U$ is the more precise (the less imprecise) of itself.

Example 2.2: Let us depict above notions by examples referring to Table 2.2. Consider the concept “Decision”, i.e., the set $X(\text{Classes} = \text{edible}) = \{2,3,4,5,6,7,8,10\}$ and the set of attributes

$$C = \left\{ \begin{array}{l} \text{Cap - Shape, Cap - Color, Odor, Gill - Spacing, Gill - Size,} \\ \text{Stalk - Shape, Veil - Type, Ring - Number, Ring - Type} \end{array} \right\}.$$

The partition of U induced by $IND(C)$ is given by

$$U/C = \{\{1\}, \{2,6\}, \{3,8\}, \{4,9\}, \{5\}, \{7\}, \{10\}\}.$$

The corresponding lower approximation and upper approximation of X are as follows

$$\underline{C}(X) = \{2,3,5,6,7,8,10\} \text{ and } \overline{C}(X) = \{2,3,4,5,6,7,8,9,10\}.$$

Thus, concept “Decision” is imprecise (rough). For this case, the accuracy of approximation is given as

$$\alpha_c(X) = \frac{7}{9}.$$

It means that the concept “Decision” can be characterized partially employing attributes Cap-Shape, Cap-Color, Odor, Gill-Spacing, Gill-Size, Stalk-Shape, Veil-Type, Ring-Number, and Ring-Type.

3.2 Maximum Dependency of Attributes (MDA)

3.2.1 Selecting a clustering attribute

By using clustering methods, we are applicable to cluster the data that having numerical values for attributes. Then, we can focused on attributes with numerical values due to the fact it is relatively easy to define similarities from geometric position of the numerical data. Some element has similarity attributes than others and we cannot define their best attributes. That is why we are using clustering to cluster their attributes then using rough set to choose the best attributes among the attributes.

3.2.2 Model for selecting a clustering attribute?

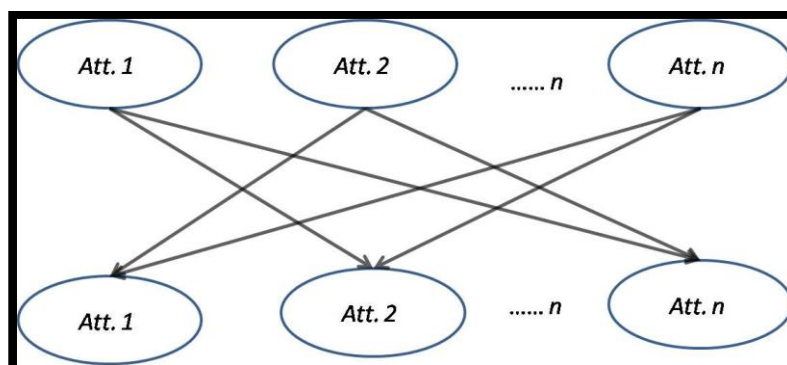


Figure 5: Clustering Attribute Diagram

3.3 Maximum Dependency of Attributes

3.3.1 Dependency of Attributes in an Information System

Discovering dependencies between attributes is one of the important issues in database analysis. Intuitively, a set of attributes D depends totally on a set of attributes C , denoted $C \Rightarrow D$, if all values of attributes from D are uniquely determined by values of attributes from C . In other words, D depends totally on C , if there exists a functional dependency between values of D and C . The formal definition of attributes dependency is given as follows.

Definition 4.1. Let $S = (U, A, V, f)$ be an information system and let D and C be any subsets of A . Attribute D is functionally depends on C , denoted $C \Rightarrow D$, if each value of D is associated exactly one value of C .

Since the concepts in information systems are generalization of the same concepts in relational databases. It is needed also a generalization concept of functional dependency of attributes, called a *partial dependency* of attributes. The notion of generalized dependency of attributes is given in the following definition.

Definition 4.2. Let $S = (U, A, V, f)$ be an information system and let D and C be any subsets of A . The dependency attribute D on C in a degree k ($0 \leq k \leq 1$), is denoted by $C \Rightarrow_k D$, where

$$k = \frac{\sum_{X \in U/D} |\underline{C}(X)|}{|U|}. \quad (4.1)$$

Obviously, $0 \leq k \leq 1$. If all set X are crisp, then $k = 1$. The expression $\sum_{X \in U/D} |\underline{C}(X)|$, called a lower approximation of the partition U/D with respect to C , is the set of all elements of U that can be uniquely classified to blocks of the partition U/D , by means of C . D is said to be fully depends (in a degree of k) on C if $k = 1$. Otherwise, D is partially depends on C . Thus, D fully (partially) depends

on C , if all (some) elements of the universe U can be uniquely classified to equivalence classes of the partition U/D , employing C .

Example 4.1. From Table 13, there are no total attributes dependencies whatsoever. If in Table 13, the value of the attribute Smell for Banana 5 were “*bad*” instead of “*Good*”, there would be a total dependency

$$\{\text{Smell}\} \Rightarrow \{\text{Classes}\},$$

because to each value of the attribute Smell there would correspond unique value of the attribute Decision. For example, for dependency

$$\{\text{Size, Smell, Color}\} \Rightarrow \{\text{Classes}\},$$

the degree value is given by $k = \frac{7}{9}$, because seven out of nine bananas can be uniquely classified as having Classes or not, employing attributes Size, Smell, and Color.

3.3.2 Algorithm of MDA

The following Table shows step-by-step to calculate Maximum Dependency of Attributes.

Table 14: Algorithm of Maximum Dependency of Attributes

Step	Maximum Dependency of Attributes
1	Given data set
2	Each attribute in data set considered as a candidate attribute to partition
3	Determine equivalence classes of attribute-value pairs
4	Determine degree of dependency of attribute a_i on attribute a_j , $i \neq j$
5	Determine the maximum degree of dependency of attribute a_i on attribute a_j , $i \neq j$
6	Select a clustering attribute

3.3.3 Example

The following example shows a calculation result of Maximum Dependency of Attributes through an information system.

Table 15: Mushrooms datasets

Mushroom	Cap- Shape	Cap- Color	Odor	Gill- Spacing	Gill- Size	Gill- Color	Stalk- Shape	Ring- Type
1	convex	brown	pungent	close	narrow	black	enlarging	pendant
2	convex	yellow	almond	close	broad	black	enlarging	pendant
3	bell	White	anise	close	broad	brown	enlarging	pendant
4	convex	White	pungent	close	narrow	pink	enlarging	pendant
5	convex	gray	none	crowded	broad	black	tapering	evanescent
6	convex	yellow	almond	close	broad	brown	enlarging	pendant
7	bell	White	almond	close	broad	grey	enlarging	pendant
8	bell	White	anise	close	broad	brown	enlarging	pendant
9	convex	White	pungent	close	narrow	pink	enlarging	pendant
10	bell	yellow	almond	close	broad	gray	enlarging	pendant

As an information system, From Table 15, we have:

$$U = \{1,2,3,4,5,6,7,8,9,10\}$$

$$A = \left\{ \begin{array}{l} \text{Cap - Shape, Cap - color, Odor, Gill - Spacing, Gill - Size, Gill - Color,} \\ \text{Stalk - Shape, Ring - Type} \end{array} \right\}$$

$$V_{\text{Cap-Shape}} = \{\text{Convex, Bell}\}$$

$$V_{\text{Cap-Color}} = \{\text{Brown, Yellow, White, Grey}\}$$

$$V_{\text{Odor}} = \{\text{Pungent, Almond, Anise, None}\}$$

$$V_{\text{Gill-Spacing}} = \{\text{Close, Crowded}\}$$

$$V_{\text{Gill-Size}} = \{\text{Narrow, Broad}\}$$

$$V_{\text{Gill-Color}} = \{\text{Black, Brown, Pink, Grey}\}$$

$$V_{\text{Stalk-Shape}} = \{\text{Enlarging, Tapering}\}$$

$$V_{\text{Ring-Type}} = \{\text{Pendant, Evanescent}\}$$

Eleven partitions of U generated by indiscernibility relation of singleton attribute are:

$$a. \quad X(\text{Cap - Shape} = \text{convex}) = \{1,2,4,5,6,9\}, \quad X(\text{Cap - Shape} = \text{bell}) = \{3,7,8,10\}$$

$$U / IND(\text{Cap - Shape}) = \{\{1,2,4,5,6,9\}, \{3,7,8,10\}\}$$

$$b. \quad X(\text{Cap - Color} = \text{brown}) = \{1\}, \quad X(\text{Cap - Color} = \text{yellow}) = \{2,6,10\},$$

$$X(\text{Cap - Color} = \text{white}) = \{3,4,7,8,9\}, \quad X(\text{Cap - Color} = \text{Grey}) = \{5\},$$

$$U / IND(\text{Cap - Color}) = \{\{1\}, \{2,6,10\}, \{3,4,7,8,9\}, \{5\}\}$$

- c. $X(\text{Odor} = \text{pungent}) = \{1,4,9\}$, $X(\text{Odor} = \text{almond}) = \{2,6,7,10\}$,
 $X(\text{Odor} = \text{anise}) = \{3,8\}$, $X(\text{Odor} = \text{none}) = \{5\}$,
 $U / IND(\text{Odor}) = \{\{1,4,9\}, \{2,6,7,10\}, \{3,8\}, \{5\}\}$
- d. $X(\text{Gill} - \text{Spacing} = \text{Close}) = \{1,2,3,4,6,7,8,9,10\}$,
 $X(\text{Gill} - \text{Spacing} = \text{Crowded}) = \{5\}$,
 $U / IND(\text{Gill} - \text{Spacing}) = \{\{1,2,3,4,6,7,8,9,10\}, \{5\}\}$
- e. $X(\text{Gill} - \text{Size} = \text{Narrow}) = \{1,4,9\}$, $X(\text{Gill} - \text{Size} = \text{Broad}) = \{2,3,5,6,7,8,10\}$
 $U / IND(\text{Gill} - \text{Size}) = \{\{1,4,9\}, \{2,3,5,6,7,8,10\}\}$
- f. $X(\text{Gill} - \text{Color} = \text{Black}) = \{1,2,5\}$, $X(\text{Gill} - \text{Color} = \text{Brown}) = \{3,6,8\}$,
 $X(\text{Gill} - \text{Color} = \text{Pink}) = \{4,9\}$, $X(\text{Gill} - \text{Color} = \text{Grey}) = \{7,10\}$,
 $U / IND(\text{Gill} - \text{Color}) = \{\{1,2,5\}, \{3,6,8\}, \{4,9\}, \{7,10\}\}$
- g. $X(\text{Stalk} - \text{Shape} = \text{Enlarging}) = \{1,2,3,4,6,7,8,9,10\}$,
 $X(\text{Stalk} - \text{Shape} = \text{Tapering}) = \{5\}$,
 $U / IND(\text{Stalk} - \text{Shape}) = \{\{1,2,3,4,6,7,8,9,10\}, \{5\}\}$
- h. $X(\text{Ring} - \text{Type} = \text{Pendant}) = \{1,2,3,4,6,7,8,9,10\}$,
 $X(\text{Ring} - \text{Type} = \text{Evanescant}) = \{5\}$,
 $U / IND(\text{Ring} - \text{Type}) = \{\{1,2,3,4,6,7,8,9,10\}, \{5\}\}$

Calculation of the Dependency of Attributes on each attribute in Table 15

a. $a_j \Rightarrow a_{cap-shape}$, $j = cap - color, odor, gill - spacing, gil - size, gill - color, stalk - shape, ring - type$

1) $a_{cap-color} \Rightarrow_k a_{cap-shape}$

$$k = \frac{|POS_{a_{cap-color}}(a_{cap-shape})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{cap-shape})} IND(a_{cap-color})(X)|}{|U|}$$

$$= \frac{|IND(a_{cap-color})\{x | a_{cap-shape}(x) = convex\}| + |IND(a_{cap-color})\{x | a_{cap-shape}(x) = bell\}|}{|U|}$$

$$= \frac{|\{1,5\}| + |\phi|}{10} = 0.2$$

2) $a_{odor} \Rightarrow_k a_{cap-shape}$, $k = \frac{|\{1,4,5,9\}| + |\{3,8\}|}{10} = 0.6$

3) $a_{gill-spacing} \Rightarrow_k a_{cap-shape}$, $k = \frac{|\{5\}| + |\phi|}{10} = 0.1$

4) $a_{gill-size} \Rightarrow_k a_{cap-shape}$, $k = \frac{|\{1,4,9\}| + |\phi|}{10} = 0.3$

5) $a_{gill-color} \Rightarrow_k a_{cap-shape}$, $k = \frac{|\{1,2,4,5,9\}| + |\{7,10\}|}{10} = 0.7$

6) $a_{stalk-shape} \Rightarrow_k a_{cap-shape}$, $k = \frac{|\{5\}| + |\phi|}{10} = 0.1$

7) $a_{ring-type} \Rightarrow_k a_{cap-shape}$, $k = \frac{|\{5\}| + |\phi|}{10} = 0.1$

- b. $a_j \Rightarrow a_{cap-color}$, $j = cap - shape, odor, gill - spacing, gil - size, gill - color,$
stalk - shape, ring - type

1) $a_{cap-shape} \Rightarrow_k a_{cap-color}$

$$k = \frac{|POS_{a_{cap-shape}}(a_{cap-color})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{cap-color})} IND(a_{cap-shape})(X)|}{|U|}$$

$$= \frac{|\underbrace{IND(a_{cap-shape})\{x | a_{cap-color}(x) = brown\}}_{|\phi|}| + |\underbrace{IND(a_{cap-shape})\{x | a_{cap-color}(x) = yellow\}}_{|\phi|}| + |\underbrace{IND(a_{cap-shape})\{x | a_{cap-color}(x) = white\}}_{|\phi|}| + |\underbrace{IND(a_{cap-shape})\{x | a_{cap-color}(x) = grey\}}_{|\phi|}|}{|U|}$$

$$= \frac{|\phi| + |\phi| + |\phi| + |\phi|}{10} = 0$$

2) $a_{odor} \Rightarrow_k a_{cap-color}$, $k = \frac{|\phi| + |\phi| + |\{3,8\}| + |\{5\}|}{10} = 0.3$

3) $a_{gill-spacing} \Rightarrow_k a_{cap-color}$, $k = \frac{|\phi| + |\phi| + |\phi| + |\{5\}|}{10} = 0.1$

4) $a_{gill-size} \Rightarrow_k a_{cap-color}$, $k = \frac{|\phi| + |\phi| + |\phi| + |\phi|}{10} = 0$

5) $a_{gill-color} \Rightarrow_k a_{cap-color}$, $k = \frac{|\phi| + |\phi| + |\{4,9\}| + |\phi|}{10} = 0.2$

6) $a_{stalk-shape} \Rightarrow_k a_{cap-color}$, $k = \frac{|\phi| + |\phi| + |\phi| + |\{5\}|}{10} = 0.1$

7) $a_{ring-type} \Rightarrow_k a_{cap-color}$, $k = \frac{|\phi| + |\phi| + |\phi| + |\{5\}|}{10} = 0.1$

c. $a_j \Rightarrow a_{odor}$, $j = cap - shape, cap - color, gill - spacing, gil - size, gill - color, stalk - shape, ring - type$

1) $a_{cap-shape} \Rightarrow_k a_{odor}$

$$k = \frac{|POS_{a_{cap-shape}}(a_{odor})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{odor})} IND(a_{cap-shape})(X)|}{|U|}$$

$$= \frac{|\underbrace{IND(a_{cap-shape})\{x|a_{odor}(x) = pungent\}}| + |\underbrace{IND(a_{cap-shape})\{x|a_{odor}(x) = almond\}}| + |\underbrace{IND(a_{cap-shape})\{x|a_{odor}(x) = anise\}}| + |\underbrace{IND(a_{cap-shape})\{x|a_{odor}(x) = none\}}|}{|U|}$$

$$= \frac{|\emptyset| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0$$

2) $a_{cap-color} \Rightarrow_k a_{odor}$, $k = \frac{|\{1\}| + |\{2,6,10\}| + |\emptyset| + |\{5\}|}{10} = 0.5$

3) $a_{gill-spacing} \Rightarrow_k a_{odor}$, $k = \frac{|\emptyset| + |\emptyset| + |\emptyset| + |\{5\}|}{10} = 0.1$

4) $a_{gill-size} \Rightarrow_k a_{odor}$, $k = \frac{|\{1,4,9\}| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0.3$

5) $a_{gill-color} \Rightarrow_k a_{odor}$, $k = \frac{|\{4,9\}| + |\{7,10\}| + |\emptyset| + |\emptyset|}{10} = 0.4$

6) $a_{stalk-shape} \Rightarrow_k a_{odor}$, $k = \frac{|\emptyset| + |\emptyset| + |\emptyset| + |\{5\}|}{10} = 0.1$

7) $a_{ring-type} \Rightarrow_k a_{odor}$, $k = \frac{|\emptyset| + |\emptyset| + |\emptyset| + |\{5\}|}{10} = 0.1$

d. $a_j \Rightarrow a_{gill-spacing}$, $j = cap - shape, cap - color, odor, gil - size, gill - color, stalk - shape, ring - type$

1) $a_{cap-shape} \Rightarrow_k a_{gill-spacing}$

$$\begin{aligned} k &= \frac{|POS_{a_{cap-shape}}(a_{gill-spacing})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{gill-spacing})} IND(a_{cap-shape})(X)|}{|U|} \\ &= \frac{|IND(a_{cap-shape})\{x | a_{gill-spacing}(x) = close\}| + |IND(a_{cap-shape})\{x | a_{gill-spacing}(x) = crowded\}|}{|U|} \\ &= \frac{|\{3,7,8,10\}| + |\emptyset|}{10} = 0.4 \end{aligned}$$

$$2) a_{cap-color} \Rightarrow_k a_{gill-spacing}, k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$$

$$3) a_{odor} \Rightarrow_k a_{gill-spacing}, k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$$

$$4) a_{gill-size} \Rightarrow_k a_{gill-spacing}, k = \frac{|\{1,4,9\}| + |\emptyset|}{10} = 0.3$$

$$5) a_{gill-color} \Rightarrow_k a_{gill-spacing}, k = \frac{|\{3,4,6,7,8,9,10\}| + |\emptyset|}{10} = 0.7$$

$$6) a_{stalk-shape} \Rightarrow_k a_{gill-spacing}, k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$$

$$7) a_{ring-type} \Rightarrow_k a_{gill-spacing}, k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$$

e. $a_j \Rightarrow a_{gill-size}$, $j = cap - shape, cap - color, odor, gill - spacing, gill - color,$
stalk - shape, ring - type

1) $a_{cap-shape} \Rightarrow_k a_{gill-size}$

$$k = \frac{|POS_{a_{cap-shape}}(a_{gill-size})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{gill-size})} IND(a_{cap-shape})(X)|}{|U|}$$

$$= \frac{|IND(a_{cap-shape})\{x | a_{gill-size}(x) = close\}| + |IND(a_{cap-shape})\{x | a_{gill-size}(x) = crowded\}|}{|U|}$$

$$= \frac{|\emptyset| + |\{3,7,8,10\}|}{10} = 0.4$$

2) $a_{cap-color} \Rightarrow_k a_{gill-size}$, $k = \frac{|\{1\}| + |\{2,5,6,10\}|}{10} = 0.5$

3) $a_{odor} \Rightarrow_k a_{gill-size}$, $k = \frac{|\{1,4,9\}| + |\{2,3,5,6,7,8,10\}|}{10} = 1$

4) $a_{gill-spacing} \Rightarrow_k a_{gill-size}$, $k = \frac{|\emptyset| + |\{5\}|}{10} = 0.1$

5) $a_{gill-color} \Rightarrow_k a_{gill-size}$, $k = \frac{|\{4,9\}| + |\{3,6,7,8,10\}|}{10} = 0.7$

6) $a_{stalk-shape} \Rightarrow_k a_{gill-size}$, $k = \frac{|\emptyset| + |\{5\}|}{10} = 0.1$

7) $a_{ring-type} \Rightarrow_k a_{gill-size}$, $k = \frac{|\emptyset| + |\{5\}|}{10} = 0.1$

f. $a_j \Rightarrow a_{gill-color}$, $j = cap - shape, cap - color, odor, gil - spacing, gill - size, stalk - shape, ring - type$

1) $a_{cap-shape} \Rightarrow_k a_{gill-color}$

$$\begin{aligned}
 k &= \frac{|POS_{a_{cap-shape}}(a_{gill-color})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{gill-color})} IND(a_{cap-shape})(X)|}{|U|} \\
 &= \frac{|\underbrace{IND(a_{cap-shape})\{x | a_{gill-color}(x) = black\}}_{\emptyset}| + |\underbrace{IND(a_{cap-shape})\{x | a_{gill-color}(x) = brown\}}_{\emptyset}| + \\
 &= \frac{|\underbrace{IND(a_{cap-shape})\{x | a_{gill-color}(x) = pink\}}_{\emptyset}| + |\underbrace{IND(a_{cap-shape})\{x | a_{gill-color}(x) = grey\}}_{\emptyset}|}{|U|} \\
 &= \frac{|\emptyset| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0
 \end{aligned}$$

2) $a_{cap-color} \Rightarrow_k a_{gill-color}$, $k = \frac{|\{1,5\}| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0.2$

3) $a_{odor} \Rightarrow_k a_{gill-color}$, $k = \frac{|\{5\}| + |\{3,8\}| + |\emptyset| + |\emptyset|}{10} = 0.3$

4) $a_{gill-spacing} \Rightarrow_k a_{gill-color}$, $k = \frac{|\{5\}| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0.1$

5) $a_{gill-size} \Rightarrow_k a_{gill-color}$, $k = \frac{|\emptyset| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0$

6) $a_{stalk-shape} \Rightarrow_k a_{gill-color}$, $k = \frac{|\{5\}| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0.1$

7) $a_{ring-type} \Rightarrow_k a_{gill-color}$, $k = \frac{|\{5\}| + |\emptyset| + |\emptyset| + |\emptyset|}{10} = 0.1$

g. $a_j \Rightarrow a_{stalk-shape}$, $j = cap - shape, cap - color, odor, gil - spacing, gill - size, gill - color, ring - type$

1) $a_{cap-shape} \Rightarrow_k a_{stalk-shape}$

$$\begin{aligned} k &= \frac{|POS_{a_{cap-shape}}(a_{stalk-shape})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{stalk-shape})} IND(a_{cap-shape})(X)|}{|U|} \\ &= \frac{|IND(a_{cap-shape})\{x | a_{stalk-shape}(x) = enl\ arg\ ing\}| + |IND(a_{cap-shape})\{x | a_{stalk-shape}(x) = tapering\}|}{|U|} \\ &= \frac{|\{3,7,8,10\}| + |\emptyset|}{10} = 0.4 \end{aligned}$$

2) $a_{cap-color} \Rightarrow_k a_{stalk-shape}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

3) $a_{odor} \Rightarrow_k a_{stalk-shape}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

4) $a_{gill-spacing} \Rightarrow_k a_{stalk-shape}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

5) $a_{gill-size} \Rightarrow_k a_{stalk-shape}$, $k = \frac{|\{1,4,9\}| + |\emptyset|}{10} = 0.3$

6) $a_{gill-color} \Rightarrow_k a_{stalk-shape}$, $k = \frac{|\{3,4,6,7,8,9,10\}| + |\emptyset|}{10} = 0.7$

7) $a_{ring-type} \Rightarrow_k a_{stalk-shape}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

h. $a_j \Rightarrow a_{ring-type}$, $j = cap - shape, cap - color, odor, gil - spacing, gill - size, gill - color, stalk - shape$

1) $a_{cap-shape} \Rightarrow_k a_{ring-type}$

$$\begin{aligned} k &= \frac{|POS_{a_{cap-shape}}(a_{ring-type})|}{|U|} = \frac{|\bigcup_{X \in U / IND(a_{ring-type})} IND(a_{cap-shape})(X)|}{|U|} \\ &= \frac{|IND(a_{cap-shape})\{x | a_{ring-type}(x) = pendant\}| + |IND(a_{cap-shape})\{x | a_{ring-type}(x) = evanescent\}|}{|U|} \\ &= \frac{|\{3,7,8,10\}| + |\emptyset|}{10} = 0.4 \end{aligned}$$

2) $a_{cap-color} \Rightarrow_k a_{ring-type}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

3) $a_{odor} \Rightarrow_k a_{ring-type}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

4) $a_{gill-spacing} \Rightarrow_k a_{ring-type}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

5) $a_{gill-size} \Rightarrow_k a_{ring-type}$, $k = \frac{|\{1,4,9\}| + |\emptyset|}{10} = 0.3$

6) $a_{gill-color} \Rightarrow_k a_{ring-type}$, $k = \frac{|\{3,4,6,7,8,9,10\}| + |\emptyset|}{10} = 0.7$

7) $a_{stalk-shape} \Rightarrow_k a_{ring-type}$, $k = \frac{|\{1,2,3,4,6,7,8,9,10\}| + |\{5\}|}{10} = 1$

Table 16: Calculation of the degree of dependency attributes in Table 15

Attribute	Dependency on							MDA
Cap-Shape	Cap-color	Odor	Gill-spacing	Gill-Size	Gill-Color	Stalk-Shape	Ring-type	0.7
	0.2	0.6	0.1	0.3	0.7	0.1	0.1	
Cap-color	Cap-Shape	Odor	Gill-spacing	Gill-Size	Gill-Color	Stalk-Shape	Ring-type	0.3
	0	0.3	0.1	0	0.2	0.1	0.1	
Odor	Cap-Shape	Cap-color	Gill-spacing	Gill-Size	Gill-Color	Stalk-Shape	Ring-type	0.5
	0	0.5	0.1	0.3	0.4	0.1	0.1	
Gill-spacing	Cap-Shape	Cap-color	Odor	Gill-Size	Gill-Color	Stalk-Shape	Ring-type	1
	0.4	1	1	0.3	0.7	1	1	
Gill-Size	Cap-Shape	Cap-color	Odor	Gill-spacing	Gill-Color	Stalk-Shape	Ring-type	1
	0.4	0.5	1	0.1	0.7	0.1	0.1	
Gill-Color	Cap-Shape	Cap-color	Odor	Gill-spacing	Gill-Size	Stalk-Shape	Ring-type	0.3
	0	0.2	0.3	0.1	0	0.1	0.1	
Stalk-Shape	Cap-Shape	Cap-color	Odor	Gill-spacing	Gill-Size	Gill-Color	Ring-type	1
	0.4	1	1	1	0.3	0.7	1	
Ring-type	Cap-Shape	Cap-color	Odor	Gill-spacing	Gill-Size	Gill-Color	Stalk-Shape	1
	0.4	1	1	1	0.3	0.7	1	

3.4 Object Splitting model

From Table 16, we calculate the Maximum Dependency of Attributes of all attributes

Table 17. Maximum Dependency of Attributes

Maximum Dependency of Attributes									MDA
Attribute	Cap- Shape	Cap- color	Odor	Gill- spacing	Gill- Size	Gill- Color	Stalk- Shape	Ring- type	1
	0.7	0.3	0.5	1	1	0.3	1	1	

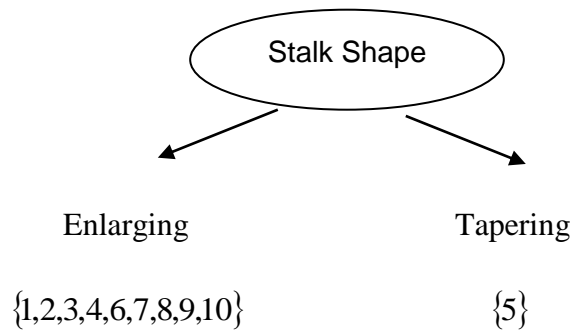
3.4.1 A clustering attribute with the Max-Max Roughness is found

Table 17 shows the calculation and illustrates that attribute a_1 and a_3 have the same Maximum Dependency of Attributes. It is recommended to look at the next highest Maximum Dependency of Attributes inside the attributes that are tied and so on until the tie is broken. In the Table 17, the second Maximum Dependency of Attributes corresponding to attribute a_1 is lower than that of a_3 . Therefore, attribute a_1 is selected as a clustering attribute and binary splitting is conducted.

3.4.2 The splitting point attributes a_1 is determined

The splitting set should include the attribute value which has Maximum Dependency of Attributes. Taking a look at Table 16:

$X(a_i = \textit{small})$ has overall minimum roughness with respect to a_i , ($i = 2, \dots, 6$) comparing to $X(a_i = \textit{medium})$ and $X(a_i = \textit{big})$. Thus, splitting on $X(a_i = \textit{small})$ versus $X(a_i = \textit{medium})$ and $X(a_i = \textit{big})$ is chosen. The partition at this stage can be represented as a tree and is shown in figure below.



CHAPTER IV

RESULT AND DISCUSSION

This chapter briefly discusses on the expected results and followed by discussion. This chapter comprises of 2 sections where the first part explain the implementation and the second part explain the dataset used in order to achieve the result. Then the expected result will be discussed.

4.1 Implementation

The proposed Data Clustering using Maximum Dependency of Attributes will be implemented using Visual Basic (VB). The system will firstly ask the user to input the data that they want to analyze. So, we can get the size of the table according to the object and parameter inserted. The system first can calculate the lower and upper class approximations based on the attributes selected by the user. After that, the system can calculate the dependency of the attributes as well as the maximum dependency of attributes. The way the calculation for the maximum dependency of attributes work is by analyzing the inserted data set. Then, each of the attribute in the data set will be considered as a candidate attribute to partition. After that, the system will determine the equivalence classes of attribute-value pairs followed by determining the degree of dependency of attribute a_i on attribute a_j where I is not equal to j . Finally, based on the result of the degree for each attribute, the system will select a clustering attribute.

4.2 Datasets

Firstly, use the small datasets of mushrooms with the criteria or attributes. The data set contains 8 attributes. The attributes are: cap-shape, cap-color, odor, gill-spacing, gill-size, gill-color, stalk-shape, and ring-type.

Secondly, is a real dataset of mushrooms with the attributes taken on “Internet”. The dataset consists of few mushrooms with the attributes with several attributes used to determine the decision.

From the proposed system of Data Clustering using Maximum Dependency of Attributes, it is expected that the system can select the best attributes that can be used to cluster the data based on its attributes, by making a fast and accurate calculation of each of the attributes degree so that the maximum dependency of the attributes can be determined. The result is important as it is the based on which attribute that is the most suitable to be selected to cluster the data. If the calculations are accurate, the best kind of clustering can be accomplished by using the most suitable attribute.

By having this system, it is also expected that user can easily cluster a data using the maximum dependency of attributes technique easily. The calculation work will be faster as the system will do it for the user especially with large datasets.

Since this system will make decision based on the data inserted by the user, there are certain cases that cannot be solved by this system. This system is not perfected yet as the method of Data Clustering using Maximum Dependency of Attributes might require some improvement later.

4.3 Interface

Main Interface



Figure 6: Main Interface

This interface is shown when the users run the software. It is the first interface of the software.

The button ‘Creator’ will take the user to the information of the creator of this software.

The button ‘About’ will take the user to the information regarding the software.

The button ‘Enter’ will take the user to the main function of the software.

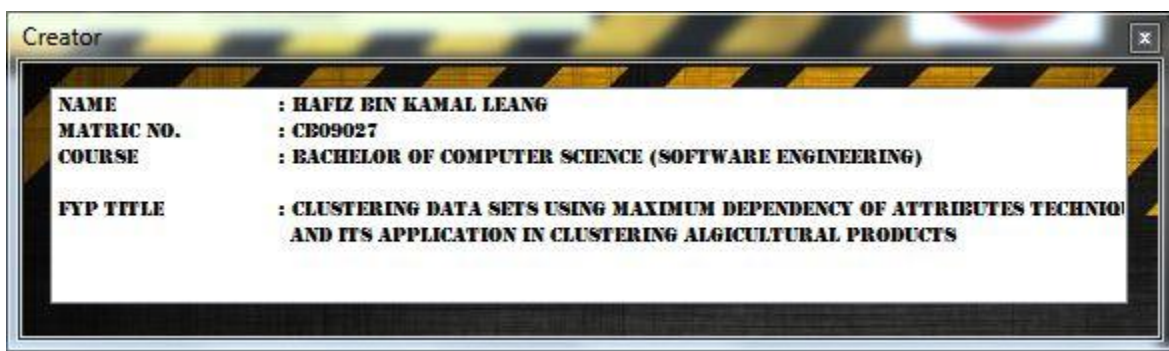


Figure 7: Creator Window

This is the interface showing the information regarding the creator of the software.



Figure 8: About Window

This is the interface showing the information about the software.

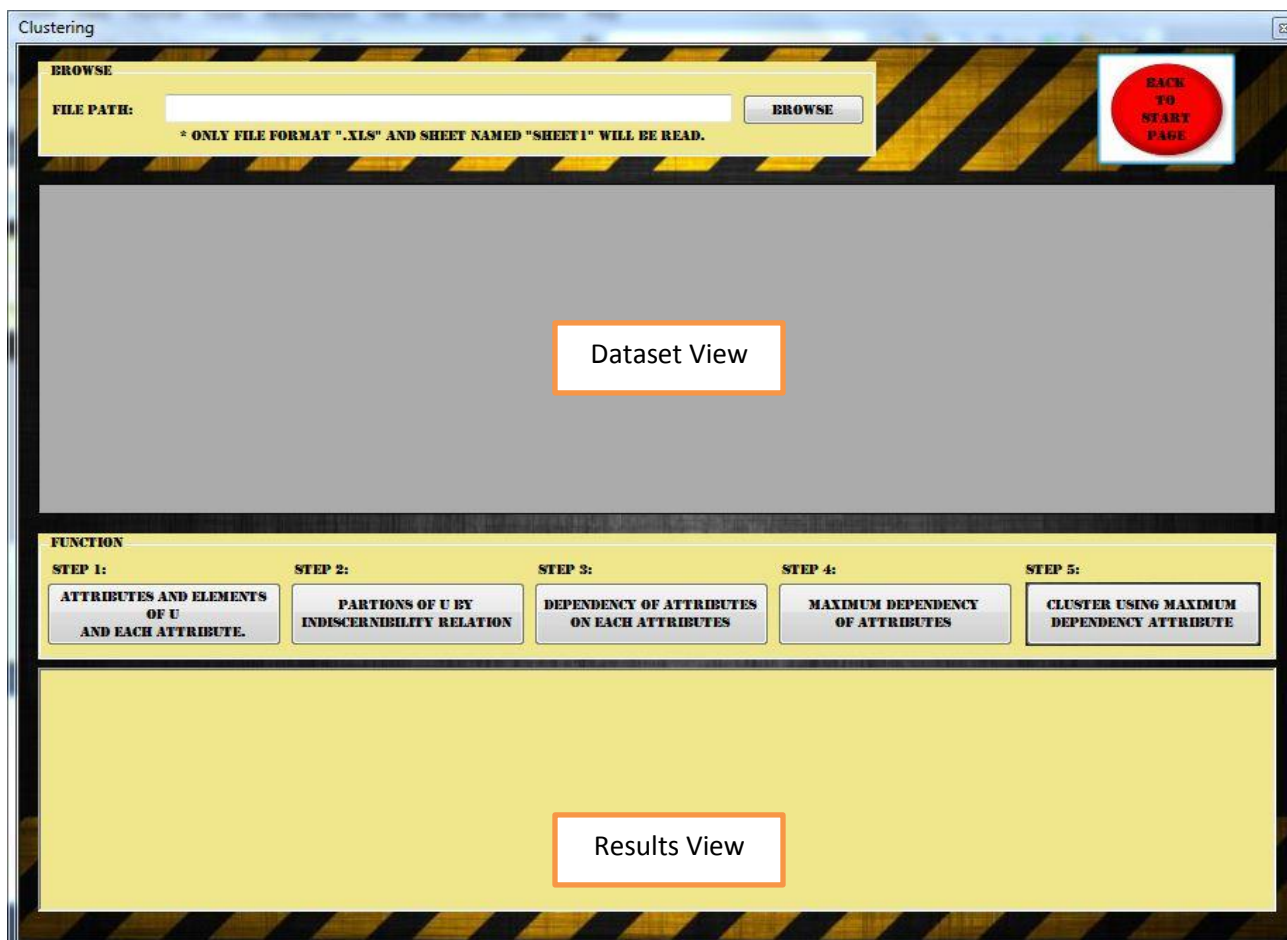


Figure 9: Function Window

This interface have 2 parts that are browse and function. In the browse section, there is browse function where user will browse for the datasets that they want to use in this application. The user need to click the browse button and then choose the dataset from their computer directory. The datasets then will be displayed at the 'Dataset View' section. In the function section, there is several button that the user can click and it represent the steps of the technique implemented in this application. With each step there will be results of calculation and it will be displayed at the 'Results View' section. At the top of the interface there is a back to start page button and that button if click will bring the user to the first interface.

Browse Table and Show Table Function:

```
Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowse.Click
    ofd1.InitialDirectory = "::{20D04FE0-3AEA-1069-A2D8-
08002B30309D}"
    ofd1.ShowDialog()
    txtFilePath.Text = ofd1.FileName.ToString

    Dim dt As New DataTable()
    Dim connStr As String = "Provider=Microsoft.Jet.OLEDB.4.0;" &
-
        "Data Source=" + txtFilePath.Text + ";" & _
        "Extended Properties=Excel 8.0;"
    Dim sqlStr As String = "SELECT * FROM [SHEET1$]"
    Dim conn As New OleDb.OleDbDataAdapter(sqlStr, connStr)

    conn.Fill(dt)
    conn.Dispose()
    Dgv1.DataSource = dt

    ReDim maxAttVar(Dgv1.Columns.Count - 1)
    ReDim attVariable(Dgv1.Columns.Count - 1, 1)

End Sub
```

Function 1: Attributes and Elements of U and Each Attribute

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    Dim maxAttValue, count As Integer

    rtbResult.Text = ""
```



```

' ##### listing element of U : #####
rtbResult.Text = "U = " & vbNewLine & "{ "
For a = 0 To Dgv1.Rows.Count - 2
    rtbResult.Text = rtbResult.Text & CStr(Dgv1.Item(0,
a).Value) & ", "
Next
rtbResult.Text = rtbResult.Text & CStr(Dgv1.Item(0,
Dgv1.Rows.Count - 1).Value) & " }"

rtbResult.Text = rtbResult.Text & vbNewLine & vbNewLine &
vbNewLine

' ##### listing attributes : #####
rtbResult.Text = rtbResult.Text & "Attributes = " & vbNewLine
& "{ "
For a = 1 To Dgv1.Columns.Count - 2
    rtbResult.Text = rtbResult.Text &
CStr(Dgv1.Columns(a).HeaderText) & ", "
Next
rtbResult.Text = rtbResult.Text &
CStr(Dgv1.Columns(Dgv1.Columns.Count - 1).HeaderText) & " }" &
vbNewLine & vbNewLine & vbNewLine

' ##### listing element of each attributes :
#####

For a = 1 To Dgv1.Columns.Count - 1
    For b1 = 1 To Dgv1.Rows.Count - 1
        Dgv1.Item(a, b1).Tag = "True"
    Next
Next

For a = 1 To Dgv1.Columns.Count - 1
    rtbResult.Text = rtbResult.Text & "V of " &
CStr(Dgv1.Columns(a).HeaderText) & " : { " & Dgv1.Item(a, 0).Value &
", "
    maxAttValue = 0
    count = 1
    For b1 = 1 To Dgv1.Rows.Count - 1
        For b2 = 0 To (b1 - 1)
            If Dgv1.Item(a, b1).Value = Dgv1.Item(a, b2).Value
Then
                Dgv1.Item(a, b1).Tag = "False"
            End If
        Next
    Next

```

```

        If Dgv1.Item(a, b1).Tag <> "False" Then
            maxAttValue = maxAttValue + 1
        End If
    Next
    For b1 = 1 To Dgv1.Rows.Count - 1
        If Dgv1.Item(a, b1).Tag <> "False" And maxAttValue <>
count Then
            rtbResult.Text = rtbResult.Text & Dgv1.Item(a,
b1).Value & ", "
            count = count + 1
        ElseIf Dgv1.Item(a, b1).Tag <> "False" And maxAttValue
= count Then
            rtbResult.Text = rtbResult.Text & Dgv1.Item(a,
b1).Value & " }" & vbNewLine & vbNewLine
        End If
    Next
Next
End Sub

```

Function 2: Partition of U By Indiscernibility Relation

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

```

```

    Dim maxAttValue As Integer
    ' ##### finding U/IND for each attribute : #####
    ' tagging all items to "True"
    For a = 1 To Dgv1.Columns.Count - 1
        For b1 = 1 To Dgv1.Rows.Count - 1
            Dgv1.Item(a, b1).Tag = "True"
        Next
    Next

    rtbResult.Text = "
++++++" & vbNewLine
& vbNewLine
    For a = 1 To Dgv1.Columns.Count - 1

        rtbResult.Text = rtbResult.Text & "Attribute " &
Dgv1.Columns(a).HeaderText & vbNewLine & vbNewLine & vbNewLine
        maxAttValue = 1
    Next

```

```

' finding the max number variable for each attribute
For b1 = 1 To Dgv1.Rows.Count - 1
    For b2 = 0 To (b1 - 1)
        If Dgv1.Item(a, b1).Value = Dgv1.Item(a, b2).Value
Then
            Dgv1.Item(a, b1).Tag = "False"
        End If
    Next
    If Dgv1.Item(a, b1).Tag <> "False" Then
        maxAttValue = maxAttValue + 1
    End If
Next

'change item tag back to "True"
For c1 = 0 To Dgv1.Rows.Count - 1
    Dgv1.Item(a, c1).Tag = "True"
Next

'write in richtextbox the (X=att) and U/IND equation for
all attribute
Uind = ""
For b = 1 To maxAttValue
    For c1 = 0 To Dgv1.Rows.Count - 1
        If Dgv1.Item(a, c1).Tag = "True" Then
            rtbResult.Text = rtbResult.Text & "X(" &
Dgv1.Columns(a).HeaderText & " = " & Dgv1.Item(a, c1).Value & ") = {
"
                UInd = UInd & "{ "
                For c2 = c1 To Dgv1.Rows.Count - 1
                    If Dgv1.Item(a, c2).Value = Dgv1.Item(a,
c1).Value Then
                        UInd = UInd & Dgv1.Item(0, c2).Value &
", "
                        rtbResult.Text = rtbResult.Text &
Dgv1.Item(0, c2).Value & ", "
                        Dgv1.Item(a, c2).Tag = "False"
                    End If
                Next
                UInd = UInd & " }, " & vbNewLine
                rtbResult.Text = rtbResult.Text & "}" &
vbNewLine & vbNewLine
            End If
        Next
        UInd = UInd & " }, " & vbNewLine
        rtbResult.Text = rtbResult.Text & "}" &
vbNewLine & vbNewLine
    End If
Next
    b = b + 1
Next

```

```

        rtbResult.Text = rtbResult.Text & "U/IND(" &
Dgv1.Columns(a).HeaderText & ")" & " = { " & vbNewLine & Uind & " }" &
vbNewLine & vbNewLine
        rtbResult.Text = rtbResult.Text & "
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++" & vbNewLine
& vbNewLine
        Next
    End Sub

```

Function 3: Dependency of Attributes on Each Attribute

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click

```

```

    Dim tag As Integer

    For e1 = 1 To Dgv1.Columns.Count - 1
        For e2 = 0 To Dgv1.Rows.Count - 1
            Dgv1.Item(e1, e2).Tag = "True"           'tag all items
to "True".
        Next
    Next
    For a = 1 To Dgv1.Columns.Count - 1
        maxAttVar(a - 1) = 1
        '1D array to store the total number of variable in each columns |
        firstly set to default value 1 | e.g; for column cap-shape, variable =
        convex & bell, maxAttVar(column cap-shape) = 2.
        attVariable(a - 1, (maxAttVar(a - 1) - 1)) = Dgv1.Item(a,
0).Value           '2D array to store the variables for each column
| attVariable(column?, variable array) | every 1st array,
attVariable(column?, 0), is equal to the 1st item in each columns.
        For b1 = 1 To Dgv1.Rows.Count - 1
            For b2 = 0 To (b1 - 1)
                If Dgv1.Item(a, b1).Value = Dgv1.Item(a, b2).Value
Then
                    Dgv1.Item(a, b1).Tag = "False"
                'tag 1st occurrence of each variable item as "True", other as "False".
                End If
            Next
            If Dgv1.Item(a, b1).Tag <> "False" Then

```

```

        maxAttVar(a - 1) = maxAttVar(a - 1) + 1
'count how many "True" in each row, then store in the
maxAttVar(column?) array.
        If maxAttVar(a - 1) > wholeMaxValue Then
            wholeMaxValue = maxAttVar(a - 1)
'count how many variable in a row.
            ReDim Preserve attVariable(Dgv1.Columns.Count
- 1, wholeMaxValue) 'ReDim, re-declaring bcause need to change the
array size | size of attVariable(column?, variable?) depends on the
how many variable in each row.
            End If
            attVariable(a - 1, (maxAttVar(a - 1) - 1)) =
Dgv1.Item(a, b1).Value 'set the values of attVariable(column?,
variable?) array.
            End If
        Next
        tag = 1
        For b1 = 0 To Dgv1.Rows.Count - 1
            If Dgv1.Item(a, b1).Tag = True Then
                Dgv1.Item(a, b1).Tag = CStr(tag)
'for each "True" tag found, change tag to 1, 2, 3, ..., n.
                tag = tag + 1
            End If
        Next
        For tag = 1 To maxAttVar(a - 1)
            For b1 = 0 To Dgv1.Rows.Count - 1
                If Dgv1.Item(a, b1).Tag = CStr(tag) Then
                    For b2 = (b1 + 1) To Dgv1.Rows.Count - 1
                        If Dgv1.Item(a, b2).Value = Dgv1.Item(a,
b1).Value And Dgv1.Item(a, b2).Tag <> CStr(tag) Then
                            Dgv1.Item(a, b2).Tag = CStr(tag)
'tag each item as 1, 2, ..n based on the variable.
                        End If
                    Next
                End If
            Next
        End If
    Next
Next
Next

Dim count, count2, countOfDA, sameObjectCount As Integer
Dim DA As Double
rtbResult.Text = "
+++++++& vbNewLine
    For a = 1 To Dgv1.Columns.Count - 1
        rtbResult.Text = rtbResult.Text & vbNewLine & "Attribute "
& Dgv1.Columns(a).HeaderText & " :" & vbNewLine & vbNewLine

```

```

For a2 = 1 To Dgv1.Columns.Count - 1
    If a <> a2 Then
        rtbResult.Text = rtbResult.Text & vbNewLine &
Dgv1.Columns(a2).HeaderText & " ⇒ " & Dgv1.Columns(a).HeaderText &
", k = ( "
        countOfDA = 0
        For tag = 1 To maxAttVar(a - 1)
            count = 0
            For b1 = 0 To Dgv1.Rows.Count - 1
                If Dgv1.Item(a, b1).Tag = CStr(tag) Then
                    count = count + 1
'counting the number of occurrence for each attribute variable in
column(a)
                End If
            Next
            For b1 = 0 To Dgv1.Rows.Count - 1
                Dgv1.Item(0, b1).Tag = "None"
'tag all item in 1st column (or U) as "None"
            Next
            'comparing each set in column(a) to each set
in column(a2) to find the dependency attribute
            For tag2 = 1 To maxAttVar(a2 - 1)
                count2 = 0
                For b1 = 0 To Dgv1.Rows.Count - 1
                    If Dgv1.Item(a2, b1).Tag = CStr(tag2)
Then
                        count2 = count2 + 1
'counting the number of occurrence for each attribute variable in
column(a2)
                    End If
                Next
                sameObjectCount = 0
                For e1 = 0 To Dgv1.Rows.Count - 1
                    If Dgv1.Item(a, e1).Tag = CStr(tag)
Then
                        For e2 = 0 To Dgv1.Rows.Count - 1
                            If Dgv1.Item(a2, e2).Tag =
CStr(tag2) And Dgv1.Item(0, e2).Value = Dgv1.Item(0, e1).Value Then
                                sameObjectCount =
sameObjectCount + 1
'comparing sets to find out the if both
set contain the same object, for every occurrence of same object found,
sameObjectCount+1
                            End If
                        Next
                    End If
                Next
            End If
        Next
    End If
End If

```


Function 4: Maximum Dependency of Attributes

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    'sorting
    Dim biggestIndex, biggestCount As Integer
    Dim temp, biggest As Double
    For a = 1 To Dgv1.Columns.Count - 1
        For a2 = 1 To Dgv1.Columns.Count - 1
            biggestIndex = a2
            For b = a2 + 1 To Dgv1.Columns.Count - 1
                If Dependency(a - 1, b - 1) >= Dependency(a - 1,
biggestIndex - 1) Then
                    biggestIndex = b
                End If
            Next
            temp = Dependency(a - 1, a2 - 1)
            Dependency(a - 1, a2 - 1) = Dependency(a - 1,
biggestIndex - 1)
            Dependency(a - 1, biggestIndex - 1) = temp
        Next
    Next

    'write in rtbresult
    rtbResult.Text = ""
    For a = 1 To Dgv1.ColumnCount - 1
        rtbResult.Text = rtbResult.Text & "    |    "
        For a2 = 1 To Dgv1.ColumnCount - 2
            rtbResult.Text = rtbResult.Text & Format(Dependency(a
- 1, a2 - 1), "0.00") & "    |    "
        Next
        rtbResult.Text = rtbResult.Text & "    -->" & vbCrLf &
Dgv1.Columns(a).HeaderText & vbCrLf
    Next

    'find the max value
    For a = 1 To Dgv1.Columns.Count - 1
        If Dependency(a - 1, 0) > biggest Then
            biggest = Dependency(a - 1, 0)
            maxColumn = a
        End If
    Next
    rtbResult.Text = rtbResult.Text & vbCrLf & "Max Value = " &
Format(biggest, "0.00") & vbCrLf

```



```

'count how many occurrence of the biggest value
biggestCount = 0
For a = 1 To Dgv1.Columns.Count - 1
    If biggest = Dependency(a - 1, 0) Then
        biggestCount += 1
    End If
Next

'in case of more than 1 of the same value of biggest
For a1 = 2 To Dgv1.ColumnCount - 2
    If biggestCount > 1 Then
        biggest = 0
        For a = 1 To Dgv1.Columns.Count - 1
            If Dependency(a - 1, a1 - 1) > biggest Then
                biggest = Dependency(a - 1, a1 - 1)
                maxColumn = a
            End If
        Next
    End If
Next

        biggestCount = 0
        For a = 1 To Dgv1.Columns.Count - 1
            If biggest = Dependency(a - 1, a1 - 1) Then
                biggestCount += 1
            End If
        Next
    End If
Next

Dim rowStringMax As String = ""
Dim rowStringOther As String
If biggestCount = 1 Then
    rtbResult.Text = rtbResult.Text & vbCrLf & "MDA = " &
biggest
    rtbResult.Text = rtbResult.Text & vbCrLf & "Splitting
Attribute = " & Dgv1.Columns(maxColumn).HeaderText
ElseIf biggestCount > 1 Then
    rtbResult.Text = rtbResult.Text & vbCrLf & "There are
more than 1 spitting attribute choices:"
    rtbResult.Text = rtbResult.Text & vbCrLf & "MDA = " &
Format(Dependency(maxColumn - 1, 0), "0.00")
    rtbResult.Text = rtbResult.Text & vbCrLf & "Splitting
Attribute = "
    For a = 1 To Dgv1.ColumnCount - 2
        rowStringMax = rowStringMax &
CStr(Dependency(maxColumn - 1, a - 1))

```

```

        Next
        For a1 = 1 To Dgv1.ColumnCount - 1
            rowStringOther = ""
            For a = 1 To Dgv1.ColumnCount - 2
                rowStringOther = rowStringOther &
CStr(Dependency(a1 - 1, a - 1))
            Next
            If rowStringOther = rowStringMax Then
                rtbResult.Text = rtbResult.Text &
Dgv1.Columns(a1).HeaderText & ", "
            End If
        Next
    End If

End Sub

```

Function 5: Cluster Using Maximum Dependency Attribute

```

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    rtbResult.Text = "Clustering result based on the Splitting
attribute " & Dgv1.Columns(maxColumn).HeaderText & vbNewLine &
vbNewLine

    For a = 0 To Dgv1.Rows.Count - 1
        Dgv1.Item(maxColumn, a).Tag = "True"
    Next

    For a = 0 To Dgv1.RowCount - 1
        If Dgv1.Item(maxColumn, a).Tag = "True" Then
            rtbResult.Text = rtbResult.Text & "-->" & vbTab & "{ "
& Dgv1.Item(0, a).Value & ", "
            For b = a + 1 To Dgv1.Rows.Count - 1
                If Dgv1.Item(maxColumn, a).Value =
Dgv1.Item(maxColumn, b).Value Then
                    rtbResult.Text = rtbResult.Text & Dgv1.Item(0,
b).Value & ", "
                    Dgv1.Item(maxColumn, b).Tag = "False"
                End If
            Next
            rtbResult.Text = rtbResult.Text & "}" & vbNewLine
        End If
    Next
Next

End Sub

```

CHAPTER V

CONCLUSIONS

5.1 Summary

Clustering is the problem of identifying the distribution of patterns and intrinsic correlation in large data set by partitioning the data points into similar classes. Data clustering is a common technique for statistical data analysis, which is used in many fields including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar object into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait often proximity according to some similarity. In this project, a clustering algorithm using rough set theory has been presented. For calculating the maximum dependency of attributes of mushrooms, the Total Roughness technique is based on Rough Set Theory. This theory was produced by Pawlak in 1981. First element is to the lower and upper approximation for the data set based on the attributes. Next is to find the dependency of the attributes and after that we will find the maximum dependency of attributes by calculating the degree for each of the element. The clustering system will be developed using Microsoft Visual Basic 2010 express as a programming language.

5.2 Suggestion and Improvement

Suggestion to improve the application:

- Add more function so that it can perform more to cluster data.

Improvements that can be done:

- Add more clustering technique into the application
- Should be implemented to other devices

REFERENCES

I. Farkas. Artificial intelligence in agriculture:

Computers and Electronics in Agriculture, Volume 40, Issues 1-3, pp. 1-3. 2003.

W. Wen, (2007). A knowledge-based intelligent electronic commerce system for selling agricultural products:

Computers and Electronics in Agriculture, Volume 57, pp. 33-46.

Graham J. Williams, Z. Huang (1996). Modeling the KDD Process:

a Four Stage Process and Four Element Model.

A. Kusiak (2001). Rough Set Theory:

a Data Mining Tool for semiconductor manufacturing. *IEEE Transactions on Electronics Packaging Manufacturing*, Volume 24, Issue 1, pp. 44-50.

Ming-Syan Chen, Jiawei Han, P.S. Yu (1996):

Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, Volume 8, Issue 6, pp. 866-883.

M. Halkidi, Y Batistakis, M.Vazirgiannis (2001):

On Clustering Techniques. *Journal of Intelligent Information System*, 2001, pp.107-145.

A.K. Jain, M.N. Murty, P.J. Flynn, (1999):

Data Clustering: A Review. *ACM Computing Surveys*, Volume 31, No.3, pp. 264-323.

Ying Zhao, G. Karypis (2003):

Hierarchical Clustering Algorithms for Document Datasets. *Data Mining and Knowledge Discovery*, Volume 10, pp. 141-168.

A.K. Jain, M.N. Murty, P.J. Flynn, (1999):

Data Clustering: A Review. *ACM Computing Surveys (CSUR)*, Volume 31, Issue 3, (1999), pp. 264-323.

K. Thangavel, Qiang Shen, A. Pethalakshmi, (2006):

Application of Clustering for Feature Selection Based on Rough Set Theory Approach. *AIML Journal*, Volume 6, Issue 1, pp. 19-27.

Y.Y. Yao, (1998):

A comparative study of fuzzy sets and rough sets, *Information Sciences*, Vol. 109, No. 1-4, pp. 227-242.

F. Ahmad (2001):

Sustainable Agriculture System Malaysia. Regional Workshop on Integrated Plant Nutrition System (IPNS), Development in Rural Poverty Alleviation. *In Proceeding of United Nations Conference Complex*, 2001.

U. Fayyad (1997):

Data Mining and Knowledge Discovery in Databases, Scientific and Statistical Database Management. *In Proceeding of Ninth International Conference 1997*, pp. 2-11.

Susan P. Imberman (2001):

Effective Use of the KDD Process and Data Mining for Computer Performance Professionals. *In Proceeding of 27th International Computer Measurement Group Conference 2001*, pp. 611-620.

R. Kalavathy, R.M. Suresh, R. Akhila (2007):

KDD and Data Mining. *In Proceeding of Information and Communication Technology in Electrical Sciences (ICTES 2007), 2007. ICTES. IET-UK International Conference.*

R. Catral, F. Oppacher, D. Deugo (2001):

Supervised and Unsupervised Data Mining with an Evolutionary Algorithm. *In Proceeding of the 2001 Congress on Evolutionary Computation*, 2001, Volume 2, pp. 767-774.

Yan Chen, Ming Yang, Lin Zhang (2009):

General Data Mining Model System Based on Sample Data Division. *In Proceeding of KAM '09 Second International Symposium on Knowledge Acquisition and Modeling*, pp.182-185.

C.I.F Agreira, C.M.M. Ferreira, F.P.M. Barbosa, (2010):

The Rough Set theory Applied to a set of the new severity indices. *In Proceeding of 2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pp. 497-502.

U. Fayyad, G. Piatetsky-Shapiro, P. Smyth (1996):

American Association for Artificial Intelligence, pp 37-54.

Wikipedia Online:

Agriculture, <http://en.wikipedia.org/wiki/Agriculture>. Retrieved October 15, 2011.

Wikipedia Online:

Agriculture in Malaysia, http://en.wikipedia.org/wiki/Agriculture_in_Malaysia. Retrieved October 15, 2011.

Nationsencyclopedia Online:

Encyclopedia of nations, Malaysia Agriculture, <http://www.nationsencyclopedia.com/economies/Asia-and-the-Pacific/Malaysia-AGRICULTURE.html>. Retrieved October 15, 2011.

Executionmih, Online:

Knowledge Discovery in Databases Process, <http://www.executionmih.com/data-mining/kdd-process-preparation-evaluation.php>, Retrieved October 14, 2011.

Wikipedia Online:

Data Mining, http://en.wikipedia.org/wiki/Data_mining. Retrieved October 17, 2011.

Britannica, Online:

Data Mining, Encyclopedia Britannica, <http://www.britannica.com/EBchecked/topic/1056150/data-mining>. Retrieved October 18, 2011.

Oracle, Online:

Classification, Oracle. http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/classify.htm. Retrieved October 20, 2011.

Oracle, Online:

Clustering, Oracle. http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/clustering.htm. Retrieved October 20, 2011.

Wikipedia, Online:

Fuzzy Clustering, Wikipedia. http://en.wikipedia.org/wiki/Fuzzy_clustering. Retrieved October 21, 2011.

Wikipedia, Online:

Rough set, Wikipedia. http://en.wikipedia.org/wiki/Rough_set#History. Retrieved October 21, 2011.

Wikipedia, Online:

Fuzzy Set, Wikipedia. http://en.wikipedia.org/wiki/Fuzzy_set. Retrieved October 21, 2011.

P. Andritos (2002):

Data Clustering Techniques. *Qualifying Oral Examination Paper*, (2002).

Syed Sibte Raza Abidi, Kok Meng Hoe, Alwyn Goh:

Analyzing Data Clusters: A Rough Set Approach to Extract Cluster-Defining Symbolic Rules.

L. Squier (2001):

What is Data Mining?. DAMA-NCR, 13 November, 2001.

OCW MIT OpenCourseWare (2008) :

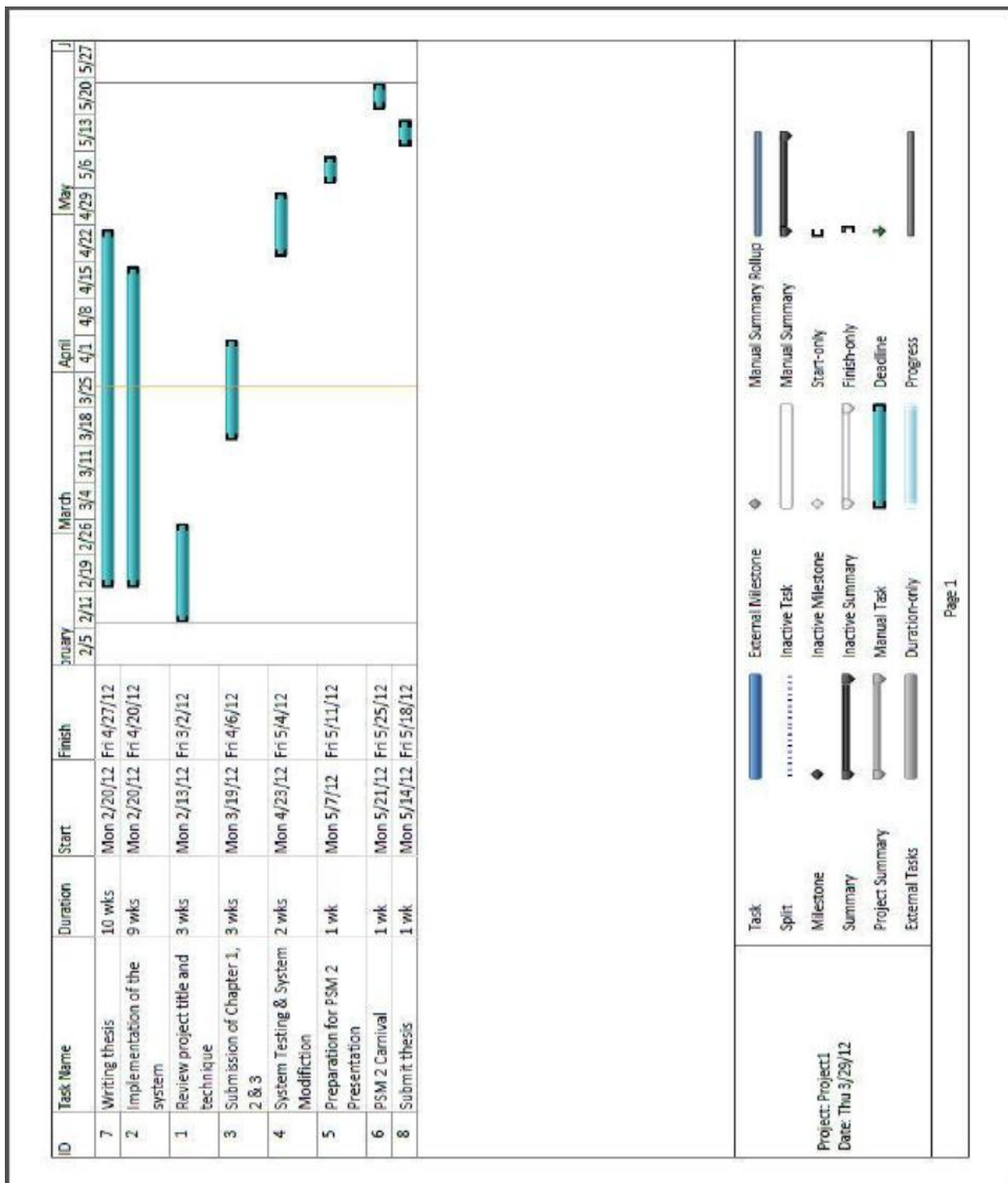
Computational Biology: Genomes, Networks, Evolution, (2008).

Jerzy W. Grzymala-Busse, (2007):

Mining Numerical Data- A rough Set Approach, *Institute of Computer Science, Polish Academy of Sciences 01-237 Warsaw, Poland.*

APPENDIX A

GANTT CHART



APPENDIX B

SOURCE CODE

Browse Table and Show Table Function:

```
Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowse.Click
    ofd1.InitialDirectory = "::{20D04FE0-3AEA-1069-A2D8-
08002B30309D}"
    ofd1.ShowDialog()
    txtFilePath.Text = ofd1.FileName.ToString

    Dim dt As New DataTable()
    Dim connStr As String = "Provider=Microsoft.Jet.OLEDB.4.0;" &
-
        "Data Source=" + txtFilePath.Text + ";" & _
        "Extended Properties=Excel 8.0;"
    Dim sqlStr As String = "SELECT * FROM [SHEET1$]"
    Dim conn As New OleDb.OleDbDataAdapter(sqlStr, connStr)

    conn.Fill(dt)
    conn.Dispose()
    Dgv1.DataSource = dt

    ReDim maxAttVar(Dgv1.Columns.Count - 1)
    ReDim attVariable(Dgv1.Columns.Count - 1, 1)

End Sub
```

Function 1: Attributes and Elements of U and Each Attribute

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    Dim maxAttValue, count As Integer

    rtbResult.Text = ""

    ' ##### listing element of U : #####
    rtbResult.Text = "U = " & vbNewLine & "{ "
    For a = 0 To Dgv1.Rows.Count - 2
        rtbResult.Text = rtbResult.Text & CStr(Dgv1.Item(0,
a).Value) & ", "
    Next
    rtbResult.Text = rtbResult.Text & CStr(Dgv1.Item(0,
Dgv1.Rows.Count - 1).Value) & " }"

    rtbResult.Text = rtbResult.Text & vbNewLine & vbNewLine &
vbNewLine

    ' ##### listing attributes : #####
    rtbResult.Text = rtbResult.Text & "Attributes = " & vbNewLine
& "{ "
    For a = 1 To Dgv1.Columns.Count - 2
        rtbResult.Text = rtbResult.Text &
CStr(Dgv1.Columns(a).HeaderText) & ", "
    Next
    rtbResult.Text = rtbResult.Text &
CStr(Dgv1.Columns(Dgv1.Columns.Count - 1).HeaderText) & " }" &
vbNewLine & vbNewLine & vbNewLine

    ' ##### listing element of each attributes :
#####

    For a = 1 To Dgv1.Columns.Count - 1
        For b1 = 1 To Dgv1.Rows.Count - 1
            Dgv1.Item(a, b1).Tag = "True"
        Next
    Next

    For a = 1 To Dgv1.Columns.Count - 1
        rtbResult.Text = rtbResult.Text & "V of " &
CStr(Dgv1.Columns(a).HeaderText) & " : { " & Dgv1.Item(a, 0).Value &
", "
        maxAttValue = 0
    
```

```
        count = 1
        For b1 = 1 To Dgv1.Rows.Count - 1
            For b2 = 0 To (b1 - 1)
                If Dgv1.Item(a, b1).Value = Dgv1.Item(a, b2).Value
Then
                    Dgv1.Item(a, b1).Tag = "False"
                End If
            Next
            If Dgv1.Item(a, b1).Tag <> "False" Then
                maxAttValue = maxAttValue + 1
            End If
        Next
        For b1 = 1 To Dgv1.Rows.Count - 1
            If Dgv1.Item(a, b1).Tag <> "False" And maxAttValue <>
count Then
                rtbResult.Text = rtbResult.Text & Dgv1.Item(a,
b1).Value & ", "
                count = count + 1
            ElseIf Dgv1.Item(a, b1).Tag <> "False" And maxAttValue
= count Then
                rtbResult.Text = rtbResult.Text & Dgv1.Item(a,
b1).Value & " }" & vbNewLine & vbNewLine
            End If
        Next
    Next
End Sub
```

Function 2: Partition of U By Indiscernibility Relation

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
```

```

    Dim maxAttValue As Integer
    ' ##### finding U/IND for each attribute : #####
    ' tagging all items to "True"
    For a = 1 To Dgv1.Columns.Count - 1
        For b1 = 1 To Dgv1.Rows.Count - 1
            Dgv1.Item(a, b1).Tag = "True"
        Next
    Next

    rtbResult.Text = "
++++++" & vbNewLine
& vbNewLine
    For a = 1 To Dgv1.Columns.Count - 1

        rtbResult.Text = rtbResult.Text & "Attribute " &
Dgv1.Columns(a).HeaderText & vbNewLine & vbNewLine & vbNewLine
        maxAttValue = 1

        ' fnding the max number variable for each attribute
        For b1 = 1 To Dgv1.Rows.Count - 1
            For b2 = 0 To (b1 - 1)
                If Dgv1.Item(a, b1).Value = Dgv1.Item(a, b2).Value
Then
                    Dgv1.Item(a, b1).Tag = "False"
                End If
            Next
            If Dgv1.Item(a, b1).Tag <> "False" Then
                maxAttValue = maxAttValue + 1
            End If
        Next

        'change item tag back to "True"
        For c1 = 0 To Dgv1.Rows.Count - 1
            Dgv1.Item(a, c1).Tag = "True"
        Next

        'write in richtextbox the (X=att) and U/IND equation for
all attribute
        Uind = ""

```

```

    For b = 1 To maxAttValue
        For c1 = 0 To Dgv1.Rows.Count - 1
            If Dgv1.Item(a, c1).Tag = "True" Then
                rtbResult.Text = rtbResult.Text & "X(" &
Dgv1.Columns(a).HeaderText & " = " & Dgv1.Item(a, c1).Value & ") = {
"
                    UInd = UInd & "{ "
                    For c2 = c1 To Dgv1.Rows.Count - 1
                        If Dgv1.Item(a, c2).Value = Dgv1.Item(a,
c1).Value Then
                            UInd = UInd & Dgv1.Item(0, c2).Value &
", "
                            rtbResult.Text = rtbResult.Text &
Dgv1.Item(0, c2).Value & ", "
                            Dgv1.Item(a, c2).Tag = "False"
                        End If
                    Next
                    UInd = UInd & " }, " & vbNewLine
                    rtbResult.Text = rtbResult.Text & "}" &
vbNewLine & vbNewLine
                End If
            Next
            b = b + 1
        Next
        rtbResult.Text = rtbResult.Text & "U/IND(" &
Dgv1.Columns(a).HeaderText & ")" & " = { " & vbNewLine & UInd & " }" &
vbNewLine & vbNewLine
        rtbResult.Text = rtbResult.Text & "
++++++++++++++++++++++++++++++++++++++++++++++++++++++++" & vbNewLine
& vbNewLine
    Next
End Sub

```

Function 3: Dependency of Attributes on Each Attribute

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```

```
    Dim tag As Integer
```

```
    For e1 = 1 To Dgv1.Columns.Count - 1
        For e2 = 0 To Dgv1.Rows.Count - 1
            Dgv1.Item(e1, e2).Tag = "True"           'tag all items
to "True".
        Next
    Next
    For a = 1 To Dgv1.Columns.Count - 1
        maxAttVar(a - 1) = 1
        '1D array to store the total number of variable in each columns |
        firstly set to default value 1 | e.g; for column cap-shape, variable =
        convex & bell, maxAttVar(column cap-shape) = 2.
        attVariable(a - 1, (maxAttVar(a - 1) - 1)) = Dgv1.Item(a,
0).Value           '2D array to store the variables for each column
| attVariable(column?, variable array) | every 1st array,
attVariable(column?, 0), is equal to the 1st item in each columns.
        For b1 = 1 To Dgv1.Rows.Count - 1
            For b2 = 0 To (b1 - 1)
                If Dgv1.Item(a, b1).Value = Dgv1.Item(a, b2).Value
Then
                    Dgv1.Item(a, b1).Tag = "False"
                'tag 1st occurrence of each variable item as "True", other as "False".
                End If
            Next
            If Dgv1.Item(a, b1).Tag <> "False" Then
                maxAttVar(a - 1) = maxAttVar(a - 1) + 1
            'count how many "True" in each row, then store in the
            maxAttVar(column?) array.
                If maxAttVar(a - 1) > wholeMaxValue Then
                    wholeMaxValue = maxAttVar(a - 1)
                'count how many variable in a row.
                    ReDim Preserve attVariable(Dgv1.Columns.Count
- 1, wholeMaxValue) 'ReDim, re-declaring bcause need to change the
array size | size of attVariable(column?, variable?) depends on the
how many variable in each row.
                End If
```

```

        attVariable(a - 1, (maxAttVar(a - 1) - 1)) =
Dgv1.Item(a, b1).Value 'set the values of attVariable(column?,
variable?) array.
    End If
Next
tag = 1
For b1 = 0 To Dgv1.Rows.Count - 1
    If Dgv1.Item(a, b1).Tag = True Then
        Dgv1.Item(a, b1).Tag = CStr(tag)
'for each "True" tag found, change tag to 1, 2, 3, ..., n.
        tag = tag + 1
    End If
Next
For tag = 1 To maxAttVar(a - 1)
    For b1 = 0 To Dgv1.Rows.Count - 1
        If Dgv1.Item(a, b1).Tag = CStr(tag) Then
            For b2 = (b1 + 1) To Dgv1.Rows.Count - 1
                If Dgv1.Item(a, b2).Value = Dgv1.Item(a,
b1).Value And Dgv1.Item(a, b2).Tag <> CStr(tag) Then
                    Dgv1.Item(a, b2).Tag = CStr(tag)
'tag each item as 1, 2, ..n based on the variable.
                End If
            Next
        End If
    Next
Next
Next
Next
Next

Dim count, count2, countOfDA, sameObjectCount As Integer
Dim DA As Double
rtbResult.Text = "
++++++" & vbNewLine
For a = 1 To Dgv1.Columns.Count - 1
    rtbResult.Text = rtbResult.Text & vbNewLine & "Attribute "
& Dgv1.Columns(a).HeaderText & " :" & vbNewLine & vbNewLine
    For a2 = 1 To Dgv1.Columns.Count - 1
        If a <> a2 Then
            rtbResult.Text = rtbResult.Text & vbNewLine &
Dgv1.Columns(a2).HeaderText & " => " & Dgv1.Columns(a).HeaderText &
", k = ( "

            countOfDA = 0
            For tag = 1 To maxAttVar(a - 1)
                count = 0
                For b1 = 0 To Dgv1.Rows.Count - 1
                    If Dgv1.Item(a, b1).Tag = CStr(tag) Then

```



```

        count = count + 1
'counting the number of occurrence for each attribute variable in
column(a)
        End If
    Next
    For b1 = 0 To Dgv1.Rows.Count - 1
        Dgv1.Item(0, b1).Tag = "None"
'tag all item in 1st column (or U) as "None"
    Next
        'comparing each set in column(a) to each set
in column(a2) to find the dependency attribute
    For tag2 = 1 To maxAttVar(a2 - 1)
        count2 = 0
        For b1 = 0 To Dgv1.Rows.Count - 1
            If Dgv1.Item(a2, b1).Tag = CStr(tag2)
Then
                count2 = count2 + 1
'counting the number of occurrence for each attribute variable in
column(a2)
                End If
            Next
            sameObjectCount = 0

            For e1 = 0 To Dgv1.Rows.Count - 1
                If Dgv1.Item(a, e1).Tag = CStr(tag)
Then
                    For e2 = 0 To Dgv1.Rows.Count - 1
                        If Dgv1.Item(a2, e2).Tag =
CStr(tag2) And Dgv1.Item(0, e2).Value = Dgv1.Item(0, e1).Value Then
                            sameObjectCount =
sameObjectCount + 1                'comparing sets to find out the if both
set contain the same object, for every occurrence of same object found,
sameObjectCount+1
                                End If
                            Next
                        End If
                    Next
                End If
            Next

            If sameObjectCount = count2 Then
                For e2 = 0 To Dgv1.Rows.Count - 1
                    If Dgv1.Item(a2, e2).Tag =
CStr(tag2) Then
                        Dgv1.Item(0, e2).Tag = "DA"
'tag object as "DA" if sameObjectCount is the same as the number of
occurrence of the respective attribute variable in (a2)
                    End If
                End If
            End If
        End If
    Next
End If

```


Function 4: Maximum Dependency of Attributes

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    'sorting
    Dim biggestIndex, biggestCount As Integer
    Dim temp, biggest As Double
    For a = 1 To Dgv1.Columns.Count - 1
        For a2 = 1 To Dgv1.Columns.Count - 1
            biggestIndex = a2
            For b = a2 + 1 To Dgv1.Columns.Count - 1
                If Dependency(a - 1, b - 1) >= Dependency(a - 1,
biggestIndex - 1) Then
                    biggestIndex = b
                End If
            Next
            temp = Dependency(a - 1, a2 - 1)
            Dependency(a - 1, a2 - 1) = Dependency(a - 1,
biggestIndex - 1)
            Dependency(a - 1, biggestIndex - 1) = temp
        Next
    Next

    'write in rtbresult
    rtbResult.Text = ""
    For a = 1 To Dgv1.ColumnCount - 1
        rtbResult.Text = rtbResult.Text & "    |    "
        For a2 = 1 To Dgv1.ColumnCount - 2
            rtbResult.Text = rtbResult.Text & Format(Dependency(a
- 1, a2 - 1), "0.00") & "    |    "
        Next
        rtbResult.Text = rtbResult.Text & "    -->" & vbCrLf &
Dgv1.Columns(a).HeaderText & vbCrLf
    Next

    'find the max value
    For a = 1 To Dgv1.Columns.Count - 1
        If Dependency(a - 1, 0) > biggest Then
            biggest = Dependency(a - 1, 0)
            maxColumn = a
        End If
    Next
    rtbResult.Text = rtbResult.Text & vbCrLf & "Max Value = " &
Format(biggest, "0.00") & vbCrLf

```

```

'count how many occurrence of the biggest value
biggestCount = 0
For a = 1 To Dgv1.Columns.Count - 1
    If biggest = Dependency(a - 1, 0) Then
        biggestCount += 1
    End If
Next

'in case of more than 1 of the same value of biggest
For a1 = 2 To Dgv1.ColumnCount - 2
    If biggestCount > 1 Then
        biggest = 0
        For a = 1 To Dgv1.Columns.Count - 1
            If Dependency(a - 1, a1 - 1) > biggest Then
                biggest = Dependency(a - 1, a1 - 1)
                maxColumn = a
            End If
        Next
    End If
Next

        biggestCount = 0
        For a = 1 To Dgv1.Columns.Count - 1
            If biggest = Dependency(a - 1, a1 - 1) Then
                biggestCount += 1
            End If
        Next
    End If
Next

Dim rowStringMax As String = ""
Dim rowStringOther As String
If biggestCount = 1 Then
    rtbResult.Text = rtbResult.Text & vbNewLine & "MDA = " &
biggest
    rtbResult.Text = rtbResult.Text & vbNewLine & "Splitting
Attribute = " & Dgv1.Columns(maxColumn).HeaderText
ElseIf biggestCount > 1 Then
    rtbResult.Text = rtbResult.Text & vbNewLine & "There are
more than 1 spitting attribute choices:"
    rtbResult.Text = rtbResult.Text & vbNewLine & "MDA = " &
Format(Dependency(maxColumn - 1, 0), "0.00")
    rtbResult.Text = rtbResult.Text & vbNewLine & "Splitting
Attribute = "
        For a = 1 To Dgv1.ColumnCount - 2
            rowStringMax = rowStringMax &
CStr(Dependency(maxColumn - 1, a - 1))

```

```
        Next
        For a1 = 1 To Dgv1.ColumnCount - 1
            rowStringOther = ""
            For a = 1 To Dgv1.ColumnCount - 2
                rowStringOther = rowStringOther &
CStr(Dependency(a1 - 1, a - 1))
            Next
            If rowStringOther = rowStringMax Then
                rtbResult.Text = rtbResult.Text &
Dgv1.Columns(a1).HeaderText & ", "
            End If
        Next
    End If

End Sub
```

Function 5: Cluster Using Maximum Dependency Attribute

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    rtbResult.Text = "Clustering result based on the Splitting
attribute " & Dgv1.Columns(maxColumn).HeaderText & vbNewLine &
vbNewLine

    For a = 0 To Dgv1.Rows.Count - 1
        Dgv1.Item(maxColumn, a).Tag = "True"
    Next

    For a = 0 To Dgv1.RowCount - 1
        If Dgv1.Item(maxColumn, a).Tag = "True" Then
            rtbResult.Text = rtbResult.Text & "-->" & vbTab & "{ "
& Dgv1.Item(0, a).Value & ", "
            For b = a + 1 To Dgv1.Rows.Count - 1
                If Dgv1.Item(maxColumn, a).Value =
Dgv1.Item(maxColumn, b).Value Then
                    rtbResult.Text = rtbResult.Text & Dgv1.Item(0,
b).Value & ", "
                    Dgv1.Item(maxColumn, b).Tag = "False"
                End If
            Next
            rtbResult.Text = rtbResult.Text & "}" & vbNewLine
        End If
    Next
End Sub
```