

*Republic of Iraq  
Ministry of Higher Education  
and Scientific Research  
Al-Nahrain University  
College of Science*



# *Data Mining Using Association Rules with Fuzzy Logic*

*A Thesis*

*Submitted to the College of Science, Al-Nahrain University  
In Partial Fulfillment of the Requirements for  
The Degree of Master of Science in Computer Science*

By

***Amany Mohammad Abood***

**(B.Sc. 2005)**

Supervised by

***Dr. Sawsan K. Thamer***

2008

1429

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا  
عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ

الْحَكِيمُ

صدق الله العظيم

البقرة - 32

## Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Computer Science/College of Science/Al-Nahrain University, by Amany Mohammad Abood as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:

Name: **Sawsan Thamer**

Title: **Lecturer**

Date: / / **2008**

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:

Name: **Dr. Taha S. Bashaga**

Title: **Head of the Department of Computer Science, Al-Nahrain University.**

Date: / / **2008**

## *Certification of the Examination Committee*

We chairman and members of the examination committee, certify that we have studies this thesis "**Data Mining Using Association Rules with Fuzzy Logic**" presented by the student **Amany Mohammad Abood** and examined her in its contents and that we have found it worthy to be accepted for the degree of Master of Science in Computer Science with **good degree**.

Signature:

Name: **Dr. Loay E. George**

Title : **Assist. Prof.**

Date : / / **2009**

**(Chairman)**

Signature:

Name: **Dr. Ban N. Thanoon**

Title : **Assist. Prof.**

Date : / / **2009**

**(Member)**

Signature:

Name: **Osama A. Awad**

Title : **Lecturer**

Date : / / **2009**

**(Member)**

Signature:

Name: **Sawsan K. Thamer**

Title: **Lecturer**

Date: / / **2009**

**(Supervisor)**

Approved by the Dean of the Collage of Science, Al-Nahrain University.

Signature:

Name: **Dr. LAITH ABDUL AZIZ AL-ANI**

Title : **Assistant Professor**

Date : / / **2009**

**(Dean of Collage of Science)**

## ***Acknowledgment***

*First of all great thanks are due to God who helped me and gave me the ability to achieve this research from first to last step.*

*I would like to express my sincere appreciation to my supervisor, **Dr. Sawzan K. Thamer** for her guidance, assistance and encouragement during the course of this project.*

*Grateful thanks for the Dean of the College, the Head of the Department **Dr. Taha S. Bashaga** for the continuous support during the period of my studies.*

*Deep gratitude and special thanks to my **husband** and my family: my **parents, my sister and brothers** for their encouragements and supporting to succeed in doing this work.*

*Special thanks to my **Cousin** for supporting and giving me advises.*

*Amany*



# *Dedication*

*Dedicated To my*

*Husband and Son...*

*Father and Mother...*

*Brothers and Sister...*

*Cousin...*

*To all friends and every one who helps and supports me*

*Amany*

# Abstract

The growth of massive data stores has led to the development of a number of automated processors that work to discover relationships in and between the data in those stores. These processors are often referred to by a number of names including data mining, knowledge discovery, pattern recognition, artificial and machine learning.

Data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. It is used to automatically extract structured knowledge from large datasets.

The application of fuzzy logic with data mining makes information understandable to human.

Data mining can have many methods like association rules, classification, clustering. One of the methods of implementing association rules is Apriori algorithm.

In this thesis, A Fuzzy Apriori System is built; it uses Apriori algorithm alone, then Apriori algorithm with the application of fuzzy logic to find association rules. It will find the relationships among items stored in a supermarket to present knowledge about what are the most soled items and the relations among items.

From the experimental results, it was found that fuzzy functions filters the results and make number of extracted rules less than the number of rules extracted by applying Apriori algorithm only.

# *Table of Content*



# *Table of Contents*

## **Abstract**

## **Table of Contents**

## **List of Abbreviations**

## **List of Symbols**

## **Chapter One: Overview**

1.1 Introduction.....	1
1.2 Data Mining .....	2
1.3 Data Mining and Fuzzy Logic.....	2
1.4 Literature Survey.....	3
1.5 Aim of thesis.....	7
1.6 Thesis Layout.....	7

## **Chapter Two: Data Mining and Fuzzy Logic**

2.1 Introduction.....	9
2.2 Definition of Data Mining.....	9
2.3 Data Mining Usage .....	10
2.4 Limitations of Data Mining .....	11
2.5 Data Mining Process.....	12
2.6 Data Mining and Knowledge Discovery in Database.....	13
2.7 Architecture of a typical data mining system.....	15
2.8 Tasks Accomplished by Data Mining.....	16
2.9 Data Mining Methods.....	18
2.9.1 Supervised Versus Unsupervised Methods.....	18
2.9.2 Association Rules.....	19
2.9.3 Apriori Algorithm.....	22
2.9.4 Support, Confidence, Frequent Itemsets, and the Apriori Property.....	22

2.10 Fuzzy Sets.....	23
2.11 Fuzzy Logic.....	24
2.12 Fuzzy Logic with Data Mining.....	25
2.13 Mining Fuzzy Association Rules.....	26
2.14 Association Rule Algorithm Principles.....	26
2.15 Apriori Algorithm.....	29

### **Chapter Three: Implementation of a Fuzzy Apriori System**

3.1 introduction .....	32
3.2 Apriori Algorithm.....	33
3.3.1 Example.....	34
3.4 Fuzzy Apriori Algorithm 1.....	39
3.5 Fuzzy Apriori Algorithm 2.....	42

### **Chapter Four: Experiments and Results**

4.1 Introduction.....	48
4.2 System execution.....	48
4.2.1 Implementing Apriori Algorithm (without fuzzy).....	50
4.2.2 Implementing Apriori Algorithm (with fuzzy function 1).....	57
4.2.3 Implementing Apriori Algorithm (with fuzzy function 2).....	59
4.3 Implementation Results.....	63

### **Chapter Five: Conclusions and Future Work**

5.1 Conclusions.....	66
5.2 Suggestions for future works.....	67

## *List of Abbreviations*

<b>Abbreviation</b>	<b>Meaning</b>
AR_Map	Association Rule_Map
ARM	Association Rule Minig
EAR4DAR	Extracting Association Rules for Distributed Association Ruls
FCBA	Fuzzy Classification Based on Association
GA	Genetic Algorithm
KDD	Knowledge Discovery in Database

## List of Symbols

Symbol	Meaning
$A(x)$	Fuzzy Set
$\delta$	User Threshold $> 2$
$\beta_i$	User Threshold to indicate support value, $i$ is $> 2$
C	Confidence
D	Database
I	Itemset
LHS	Left Hand Side
M	Number of Qualified Transactions
Min_Sup	Minimum Support
Min_Conf	Minimum Confidence
$\mu A(x)$	Membership Function
$\mu$	Mid Value, Membership Value
Prob	Probability
RHS	Right Hand Side
S	Support
sub	subset
$\Sigma$	Summation Function
T	Transaction
Thr	Threshold
U	Universe and Dicourse

# *Chapter One*

## *Overview*

# Chapter One

## Overview

### 1.1 Introduction

Progress in digital data acquisition and storage technology has resulted in the growth of huge databases. This has occurred in all areas of human endeavor, from the mundane (such as supermarket transaction data, credit card usage records, telephone call details, and government statistics) to the more exotic (such as images of astronomical bodies, molecular databases, and medical records). After that interest was grown toward tapping these data, and extracting from them information that might be of value to the owner of the database. The discipline concerned with this task has become known as *data mining*. Data mining is the analysis of large observational data sets to find unsuspected relationships, and to summarize the data in novel ways that are both understandable and useful to the data owner [Dav01].

The amount of data stored in databases continues to grow fast. Intuitively, this large amount of stored data contains valuable hidden knowledge, which could be used to improve the decision-making process of an organization. For instance, data about previous sales might contain interesting relationships between products and customers. The discovery of such relationships can be very useful to increase the sales of a company. However, the number of human data analysts grows at a much smaller rate than the amount of stored data. Thus, there is a clear need for semi automatic

methods for extracting knowledge from data. This need has led to the emergence of a field called data mining and knowledge discovery. This is an interdisciplinary field, using methods of several research areas (specially machine learning and statistics) to extract high level knowledge from real-world data sets. Data mining is the core step of a broader process, called knowledge discovery in databases, or knowledge discovery, for short [Ale01].

## **1.2 Data Mining**

Data mining is the process of extracting meaningful information from large quantities of data. It involves uncovering patterns in the data and is often tied to data warehousing because it makes such large amounts of data usable. Data elements are grouped into distinct categories so that predictions can be made about other pieces of data. For example, a bank may wish to ascertain the characteristics that typify customers who pay back loans. Although this could be done with database queries, the bank would first have to know what customer attributes to query for. Data mining can be used to identify what those attributes are and then make predictions about future customer behavior [Sar05].

## **1.3 Data Mining and Fuzzy Logic**

Knowledge discovery, whose objective is to obtain useful knowledge from data stored in large repositories, is recognized as a basic necessity in many areas, especially those related to business. Since data represent a certain real-world domain, patterns that hold in data show interesting relations that can be used to improve human understanding of that domain. Data mining is the step in the knowledge discovery process that attempts to

discover novel and meaningful patterns in data. The theory of fuzzy sets can certainly help data mining to reach this goal. It is widely recognized that many real world relations are intrinsically fuzzy. For instance, fuzzy clustering generally provides a more suitable partition of a set of objects than crisp clustering do. Moreover, fuzzy sets are an optimal tool to model imprecise terms and relations as commonly employed by humans in communication and understanding. As a consequence, the theory of fuzzy sets is an excellent basis to provide knowledge expressed in a meaningful way [Mig03].

## 1.4 Literature Survey

Many studies and researches have been introduced in the field of data mining, data mining with fuzzy logic, the following are some of these researches:

### 1- A Fuzzy Data Mining Algorithm for Quantitative Values, [Tzu99]

This work attempted to propose a new data-mining algorithm to enhance the capability of exploring interesting knowledge from transactions with quantitative values. The proposed algorithm integrated the fuzzy set concepts and the apriori mining algorithm to find interesting fuzzy association rules from given transaction data. The rules thus mined exhibit quantitative regularity in databases and can be used to provide some suggestions to appropriate supervisors. The proposed algorithm can also solve conventional transaction-data problems by using degraded membership functions.



## 2- Mining fuzzy association rules for classification problems, [Rue02]

This work proposed a learning algorithm, which can be viewed as a knowledge acquisition tool, to effectively discover fuzzy association rules for classification problems. The consequence part of each rule is one class label. The proposed learning algorithm consists of two phases: one to generate large fuzzy grids from training samples by fuzzy partitioning in each attribute, and the other to generate fuzzy association rules for classification problems by large fuzzy grids. The proposed learning algorithm is implemented by scanning training samples stored in a database only once and applying a sequence of Boolean operations to generate fuzzy grids and fuzzy rules; therefore, it can be easily extended to discover other types of fuzzy association rules for market basket analysis that can help managers design different store layouts and help retailers to plan which items to put on sale. The simulation results from the iris data demonstrate that the proposed learning algorithm can effectively derive fuzzy association rules for classification problems.

## 3- Fuzzy Association Rules: General Model and Applications, [Mig03]

This work developed a general model to discover association rules among items in a (crisp) set of fuzzy transactions. This general model can be particularized in several ways; each particular instance corresponds to a certain kind of pattern and/or repository of data. They describe some applications of this scheme, paying special attention to the discovery of fuzzy association rules in relational databases. The proposed model has been tested on some of the applications, specifically to discover fuzzy association rules in relational databases that contain quantitative data. The model can be employed in mining distinct types of patterns, from ordinary association

rules to fuzzy and approximate functional dependencies and gradual rules. They will be used in multimedia data mining and web mining.

#### 4- Fuzzy Classification Based on Fuzzy Association Rule Mining, [Wei04]

He investigated the way to integrate fuzzy association rule mining and fuzzy classification. First, the framework of fuzzy association rule mining is presented which incorporates fuzzy set modeling in an association rule mining technique. He studied the impact of different fuzzy aggregation operators on the rule mining result. The selection of the operator should depend on the application context. Based on the framework of fuzzy association rule mining, he proposed a heuristic method to construct the fuzzy classifier based on the set of fuzzy class association rules. He called this method the FCBA approach, where FCBA stands for Fuzzy Classification Based on Association. The objective is to build a classifier with strong classification ability. In the FCBA approach, the composite criteria of fuzzy support and fuzzy confidence is used as the rule weight to indicate the significance of the rule.

#### 5- Extracting Association Rules for Distributed Association Rules, [Raw07]

In this thesis two proposed algorithms were introduced, they focused on the principle of mining knowledge over geographical distributed systems: The main proposed algorithm (extracting Association Rules for Distributed Association Rules (EAR4DAR) algorithm, aims to extract association rules for distributed association rules) instead of extract association rules from huge quantity of distributed data at several sites, and that is through collecting the local association rules from each site and storing them, these local association rules turn in series of operations to produce global

association rules over distributed systems. Secondary Proposed Algorithm: Association Rules\_map (AR\_map) algorithm aims to get association rules by using AND logic operation which is a suitable tool to represent association relations between items, since it's giving induction for finding relation or not. These new algorithms are saving the cost which is required to communicate over the network, cost of central storage requirements, and rate of required time for execution.

#### 6- Utilize Fuzzy Data Mining to Find the Living Pattern of Customers in Hotels, [Zho07]

This study of finding the living pattern of customers in hotels adopts fuzzy data mining that combine Apriori algorithm with different min-support and min-confidence and fuzzy set theory to copy with the criteria, the yielded rules are not only useful to the hotel decision-makers, but also to those who want to do the business. This study has yielded some association rules from quantitative data sets and taken the rules to the decision-makers of the hotels who are interested in, they have achieved some effects from the association rules. Someone who wants to operate a hotel, also gets some useful information from the rules. This approach can be used in other facets, such as the discovery of the shopping pattern of consumers in the supermarket according to different categories of commodities, etc.

#### 7- An Evolutionary Data Mining Model for Fuzzy Concept Extraction, [Moh08]

In this work a method is proposed for extracting useful information from a relational database using a hybrid of genetic algorithm and fuzzy data mining approach to extract user desired information. The genetic algorithm

is employed to find a compact set of useful fuzzy concepts with a good fuzzy support for the output of fuzzy data mining process. The output of the common fuzzy mining system is constant. But sometimes users want more information from database, perhaps information with higher dimensions. So genetic algorithm is used to find information with more attributes. The genetic algorithm has one input which decides the dimension of output of system. The output of genetic algorithm is the number of linguistic values for each attribute in the database. This input is used to get fuzzy information with lower/higher attributes.

## 1.5 Aim of Thesis

The aim of this thesis is to build a system that uses data mining techniques to access dataset by applying specific algorithms for extracting desirable knowledge or interesting patterns from existing datasets for specific purposes, market basket data is used for this work. To improve the data mining work, fuzzy logic is used to increase the flexibility for supporting supermarket managers in making decisions, it will reduce the number of association rules according to specified threshold for each algorithm.

## 1.6 Thesis Layout

In the following a summary of the contents of the subsequent chapters of this thesis is given:

- **Chapter two:** this chapter presents the concepts of data mining in details, illustrates its uses and applications, etc. And it presents the relevant disciplines of fuzzy logic and association rules.

- **Chapter three:** this chapter describes the Apriori algorithm and fuzzy Apriori algorithm.
- **Chapter four:** this chapter presents the use of Apriori system, fuzzy Apriori system, and all the algorithms used to implement the proposed system, beside to the implementation interfaces.
- **Chapter five:** this chapter explores the derived conclusions, and the suggestion for future works.

# *Chapter Two*

## *Data Mining and Fuzzy Logic*

# *Chapter Two*

# *Data Mining and*

# *Fuzzy Logic*

## **2.1 Introduction**

Data mining consists of finding interesting trends or patterns in large datasets, in order to guide decisions about future activities. There is a general expectation that data mining tools should be able to identify these patterns in the data with minimal user input. The patterns identified by such tools can give a data analyst useful and unexpected insights that can be more carefully investigated subsequently, perhaps using other decision support tools [Jef06].

Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge.

The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration [Jia06].

## **2.2 Definition of Data Mining**

Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques [Dan05].

It uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions. The first and simplest analytical step in data mining is to describe the data — summarize its statistical attributes (such as means and standard deviations), visually review it using charts and graphs, and look for potentially meaningful links among variables [Two99].

### **2.3 Data Mining Usage**

Data mining is used for a variety of purposes in both the private and public sectors. Industries such as banking, insurance, medicine, and retailing commonly use data mining to reduce costs, enhance research, and increase sales. For example, the insurance and banking industries use data mining applications to detect fraud and assist in risk assessment (e.g., credit scoring). Using customer data collected over several years, companies can develop models that predict whether a customer is a good credit risk, or whether an accident claim may be fraudulent and should be investigated more closely. The medical community sometimes uses data mining to help predict the effectiveness of a procedure or medicine. Pharmaceutical firms use data mining of chemical compounds and genetic material to help guide research on new treatments for diseases. Retailers can use information collected through affinity programs (e.g., shoppers' club cards, frequent flyer points, contests) to assess the effectiveness of product selection and placement decision, coupon offers, and which products are often purchased together. Companies such as telephone service providers and music clubs can use data mining to create a churn analysis," to assess which customers are likely to remain as subscribers and which ones are likely to switch to a competitor [Jef06, Zhe01, Fan06].



In public sector, data mining applications were initially used as means to detect fraud and waste, but they have grown also to be used for purposes such as measuring and improving program performance. It has been reported that data mining has helped the federal government recover millions of dollars in fraudulent medicare payments [Pet98].

## 2.4 Limitations of Data Mining

While data mining products can be very powerful tools, they are not self-sufficient applications. To be successful, data mining required skilled technical and analytical specialists who can structure the analysis and interpret the output that is created. Consequently, the limitations of data mining are primarily data or personnel-related, rather than technology-related. Although data mining can help reveal patterns and relationships, it does not tell the user the value or significance of these patterns. These types of determinations must be made by the user. Similarly, the validity of the patterns discovered is depending on how they compare to “real world” circumstances. For example, to assess the validity of data mining application designed to identify potential terrorist suspects in a large pool individuals, the user may test the model using data that include information about known terrorists. However, while possibly re-affirming a particular profile, it does not necessarily mean that the application will identify a suspect whose behavior significantly deviates from the original model. Another limitation of data mining is that while it can identify connections between behaviors and/or variables, it does not necessary identify a casual relationship. For example, an application may identify that a pattern of behavior, such as the propensity to purchase airline tickets just shortly before the flight is scheduled to depart, is related to characteristics such as income, level of

education, and internet use. However, that does not necessarily indicate that the ticket purchasing behavior is caused by one or more of these variables. In fact, the individual's behavior could be affected by some additional variable(s) such as occupation (the need to make trips on short notice), family status (a stick relative needing care), or a hobby (taking advantage of last minute discounts to visit new destinations) [Jef06, Fan06].

## 2.5 Data Mining Process

A data mining application usually starts with an understanding of the application domain by **data analysts (data miners)**, who then identify suitable data sources and the target data. With the data, data mining can be performed, which is usually carried out in three main steps [Bin07]:

- § **Pre-processing:** The raw data is usually not suitable for mining due to various reasons. It may need to be cleaned in order to remove noises or abnormalities. The data may also be too large and/or involve many irrelevant attributes, which call for data reduction through sampling and attribute selection. Details about data pre-processing can be found in any standard data mining textbook.
- § **Data mining:** The processed data is then fed to a data mining algorithm which will produce patterns or knowledge.
- § **Post-processing:** In many applications, not all discovered patterns are useful. This step identifies those useful ones for applications. Various evaluation and visualization techniques are used to make the decision.

The whole process (also called the **data mining process**) is almost always iterative. It usually takes many rounds to achieve final satisfactory results, which are then incorporated into real world operational tasks.

## 2.6 Data Mining and Knowledge Discovery in Database

Many people treat data mining as a synonym for another popularly used term, "Knowledge Discovery in Databases", or KDD. Alternatively, others view data mining as simply an essential step in the process of knowledge discovery in databases [Jia00].

Knowledge Discovery in Databases is the process of extracting interesting, non-trivial, implicit, previously unknown and potentially useful information or patterns from data in large databases. Data Mining is the most important step in the KDD process and involves the application of data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data. The KDD process entails the application of one or more Data Mining techniques to a dataset, in order to extract specific patterns and to evaluate them on the data [And05].

Knowledge discovery as a process is depicted in Figure (2.1) it consists of an iterative sequence of the following steps:

1. Data cleaning (to remove noise and inconsistent data)
2. Data integration (where multiple data sources may be combined)
3. Data selection (where data relevant to the analysis task are retrieved from the database)
4. Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
5. Data mining (an essential process where intelligent methods are applied in order to extract data patterns)

6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures)
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

Steps 1 to 4 are different forms of data preprocessing, where the data are prepared for mining. The data mining step may interact with the user or a knowledge base. The interesting patterns are presented to the user and may be stored as new knowledge in the knowledge base. Note that according to this view, data mining is only one step in the entire process, albeit an essential one because it uncovers hidden patterns for evaluation [Jia06, Ode08].

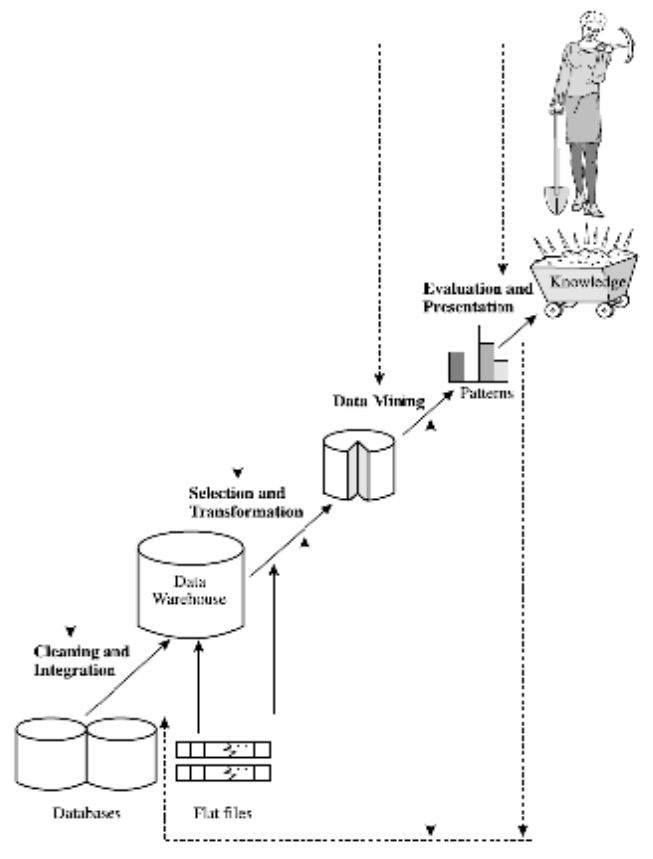


Figure (2.1) Data mining as a step in the process of knowledge discovery.

## 2.7 Architecture of a typical data mining system

The architecture of a typical data mining system may have the following major components (Figure 2.2) [Jia00]:

1. Database, data warehouse, or other information repository. This is one or a set of databases, data warehouses, spread sheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.
2. Database or data warehouse server. The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.
3. Knowledge base. This is the domain knowledge that is used to guide the search, or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).
4. Data mining engine. This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association analysis, classification, evolution and deviation analysis.

5. Pattern evaluation module. This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search towards interesting patterns. It may access interestingness thresholds stored in the knowledge base. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.
6. Graphical user interface. This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

## 2.8 Tasks Accomplished by Data Mining

The main tasks that data mining is usually called upon to accomplish are listed below [Dan05, Tay04]:

- **Description:** Descriptions of patterns and trends often suggest possible explanations for such patterns and trends. Data mining models should be as transparent as possible. That is, the results of the data mining model should describe clear patterns that are amenable to

intuitive interpretation and explanation. Some data mining methods are more suited than others to transparent interpretation. For example, decision trees provide an intuitive and human friendly explanation of their results. On the other hand, neural networks are comparatively opaque to nonspecialists, due to the nonlinearity and complexity of the model. High-quality description can often be accomplished by exploratory data analysis, a graphical method of exploring data in search of patterns and trends.

- **Estimation:** Estimation is similar to classification except that the target variable is numerical rather than categorical. Models are built using “complete” records, which provide the value of the target variable as well as the predictors. Then, for new observations, estimates of the value of the target variable are made, based on the values of the predictors.
- **Prediction:** Prediction is similar to classification and estimation, except that for prediction, the results lie in the future. Any of the methods and techniques used for classification and estimation may also be used, under appropriate circumstances, for prediction. These include the traditional statistical methods of point estimation and confidence interval estimations, simple linear regression and correlation, and multiple regression, as well as data mining and knowledge discovery methods such as neural network, decision tree, and  $k$ -nearest neighbor methods.
- **Classification:** In classification, there is a target categorical variable, such as income bracket, which, for example, could be partitioned into three classes or categories: high income, middle income, and low income. The data mining model examines a large set of records, each

record containing information on the target variable as well as a set of input or predictor variables.

- **Clustering:** Clustering refers to the grouping of records, observations, or cases into classes of similar objects. A cluster is a collection of records that are similar to one another, and dissimilar to records in other clusters. Clustering differs from classification in that there is no target variable for clustering. The clustering task does not try to classify, estimate, or predict the value of a target variable. Instead, clustering algorithms seek to segment the entire data set into relatively homogeneous subgroups or clusters, where the similarity of the records within the cluster is maximized and the similarity to records outside the cluster is minimized.
- **Dependency modeling:** describes significant dependencies among variables.

## 2.9 Data Mining Methods

Data mining methods may be categorized as either supervised or unsupervised; the following paragraph will illustrate the two methods:

### 2.9.1 Supervised Versus Unsupervised Methods

In unsupervised methods, no target variable is identified as such. Instead, the data mining algorithm searches for patterns and structure among all the variables. The most common unsupervised data mining method is clustering. For example, political consultants may analyze congressional districts using clustering methods, to uncover the locations of voter clusters that may be responsive to a particular candidate's message. In this case, all appropriate variables (e.g., income, race, gender) would be input to the



clustering algorithm, with no target variable specified, in order to develop accurate voter profiles for fund-raising and advertising purposes.

Another data mining method, which may be supervised or unsupervised, is association rule mining. In market basket analysis, for example, one may simply be interested in “which items are purchased together,” in which case no target variable would be identified. The problem here, of course, is that there are so many items for sale, that searching for all possible associations may present a daunting task, due to the resulting combinatorial explosion. Nevertheless, certain algorithms, such as the Apriori algorithm, attack this problem cleverly [Dan05].

Data mining have many methods like classification, clustering, association ... etc. In this thesis association rule mining will be used.

### **2.9.2 Association Rules**

The association task for data mining is the job of finding which attributes “go together.” Most prevalent in the business world, where it is known as affinity analysis or market basket analysis, the task of association seeks to uncover rules for quantifying the relationship between two or more attributes. Association rules are of the form “If antecedent, then consequent,” together with a measure of the support and confidence associated with the rule. For example, a particular supermarket may find that of the 1000 customers, 200 bought diapers, and of those 200 who bought diapers, 50 bought milk. Thus, the association rule would be “If buy diapers, then buy milk” with a support of  $200/1000 = 20\%$  and a confidence of  $50/200 = 25\%$  [Dan05].

Association rule mining finds interesting association or correlation relationships among a large set of data items. With massive amounts of data continuously being collected and stored in databases, many industries are becoming interested in mining association rules from their databases. For example, the discovery of interesting association relationships among huge amounts of business transaction records can help catalog design, cross-marketing, lossleader analysis, and other business decision making processes [Jia00].

Association rule discovery aims to find rules with strong associations between items from the database. It focuses on detecting relationships between items. A rule is of the form  $A \Rightarrow B$  where  $A$  is known as the antecedent and  $B$  is the consequent of the rule. Both  $A$  and  $B$  are *itemsets* from the database of transactions. An itemset can be a single item (Example: water) or a set of items (Example: water and chips). The rule implies that if an itemset  $A$  occurs in a transaction then itemset  $B$  is likely to occur in the same transaction of the database [Dha03].

The search space of rules generated from the database can also be very large, therefore to mine association rules from the database, constraints are defined. For example, 1000 items in the database have  $2^{1000}$  possible combinations of itemsets which results in a large number of rules to explore. The *minimum support* constraint is used to limit the number of itemsets that can be considered for rules to be generated. The *support* of an itemset is the frequency with which the itemset occurs in the database. For example, if 25 transactions out of 100 transactions (assuming that a set from the database consists of 100 transactions) contain Pepsi, then the support of Pepsi is 0.25. The itemsets which satisfy the minimum support constraint are *frequent itemsets*. From these itemsets the rules are developed. If the minimum

support is defined as 0.2 by the user, then Pepsi is a frequent item in the previous example as  $\text{support}(\text{pepsi}) \geq \text{minimum support}$ .

For example, consider the rule  $\text{pepsi} \Rightarrow \text{chips}$ .

- $\text{Support}(\text{pepsi}) = 0.4$ , implies that 40 percent of all customer transactions contain pepsi.
- $\text{Support}(\text{pepsi} \Rightarrow \text{chips}) = 0.2$  implies that 20 percent of all customer transactions contain pepsi and chips together.

From the rule set that is developed the user can choose to apply further constraints. The result is several rules will be pruned from the space of rules. Confidence is usually the measure of interest for generating association rules in association rule discovery. The final set of rules is referred to as interesting rules to the user. Some measures of interest which the user can specify are [Dha03]:

$$\text{Confidence}(A \Rightarrow B) = \text{support}(A \Rightarrow B) / \text{support}(A) \text{ (equation 2.1)}$$

Discovering association rules is a three part process [Dha03]:

1. Search the data space to find all the itemsets (can be a single item) whose support is greater than the user specified minimum support. These itemsets are the frequent itemsets.
2. Generate interesting rules based on the frequent itemsets. A rule is said to be interesting if its confidence is above a user's specified minimum confidence.
3. Remove (prune) all rules which are not interesting from the rule set

### 2.9.3 Apriori Algorithm

One of the algorithms used in discovering association rules is the Apriori algorithm:

The Apriori algorithm was proposed by Agrawal and Srikant. The algorithm has proved to be an efficient algorithm for mining association rules and has become the standard algorithm used for association rule discovery. Apriori follows two step process to generate rules [Dha03]:

1. The first step is to find all frequent itemsets. The itemset frequency information (support) is maintained. This step will limit the number of itemsets which are considered for the antecedent and consequent of the rules.
2. From these frequent itemsets, association rules are generated. All the items in the database are tested for minimum support. The frequent 1-itemsets found, known as a seed set, can be used to construct a candidate set of itemsets. Sets with  $k-1$  items which are frequent can be joined to construct candidate sets with  $k$  items. Then the candidate set ( $k$  itemset) is tested to see if it satisfies minimum support and if it does it becomes a seed for the next pass. This iterative process continues until no frequent itemsets are found.

### 2.9.4 Support, Confidence, Frequent Itemsets, and the Apriori Property [Jia01]

Let  $D$  be the set of transactions, where each transaction  $T$  in  $D$  represents a set of items contained in Itemset  $I$ . Suppose that there is a particular set of items  $A$  (e.g., beans and squash), and another set of items  $B$  (e.g., asparagus). Then an association rule takes the form if  $A$ , then  $B$  (i.e.,  $A$

$\Rightarrow B$ ), where the antecedent  $A$  and the consequent  $B$  are proper subsets of  $I$ , and  $A$  and  $B$  are mutually exclusive. This definition would exclude, for example, trivial rules such as if beans and squash, then beans.

The support  $S$  for a particular association rule  $A \Rightarrow B$  is the proportion of transactions in  $D$  that contain both  $A$  and  $B$ . That is,

$$S = P(A \cap B) = \frac{\text{number of transactions containing both } A \text{ and } B}{\text{total number of transactions}} \quad (\text{equation 2.2})$$

The confidence  $C$  of the association rule  $A \Rightarrow B$  is a measure of the accuracy of the rule, as determined by the percentage of transactions in  $D$  containing  $A$  that also contain  $B$ . In other words,

$$C = P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{\text{number of transactions containing both } A \text{ and } B}{\text{number of transactions containing } A} \quad (\text{equation 2.3})$$

## 2.10 Fuzzy Sets

This concept is extended by fuzzy set theory, which allows degrees of membership of elements to sets. In classical sets, elements could belong fully (i.e. have a membership of 1) or not at all (a membership of 0). Fuzzy set theory relaxes this restriction by allowing memberships to take values anywhere in the range  $[0-1]$ . A fuzzy set can be defined as a set of ordered pairs  $A = \{x, \mu_A(x) \mid x \in U\}$ . The function  $\mu_A(x)$  is called the membership function for  $A$ , mapping each element of the universe  $U$  to a membership degree in the range  $[0-1]$ . The universe may be discrete or continuous. Any fuzzy set containing at least one element with a membership degree of 1 is called normal [Ric05].

## 2.11 Fuzzy Logic

Fuzzy logic, the logic based upon which fuzzy systems operate, is much closer in spirit to human thinking and natural language than conventional digital logic. Basically, it provides an effective means of capturing the approximate and inexact nature of the real-world knowledge. Fuzzy logic was invented by Lotfi Zadeh in 1964. According to Zadeh, the essential characteristics of fuzzy logic are [Lip05]:

- In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
- Any logic system can be fuzzified.
- In fuzzy logic, knowledge is interpreted as a collection of elastic or equivalent, fuzzy constraints on a collection of variables.
- Inference is viewed as a process of propagation of elastic constraints.

Fuzzy logic states that everything is to a matter of degree (i.e. between 0 and 1) and fuzzy sets are properties (e.g., low, medium, high) whose elements belong to the sets only in a degree [Lip05].

In contrast to a classical set, which has a crisp boundary, the boundary of a fuzzy set is blurred. Almost all the labels given to groups of objects are fuzzy. For example, friends, pretty faces, tall trees etc. An object may belong to the set of objects with a certain label, with a certain membership value. In traditional set theory, this membership value only has two possible values, 1 and 0, representing the case where the object belongs to or does not belong to the set, respectively. A fuzzy term is used such as ‘big’ to label a particular group, because they share the property of objects within this group (i.e., they are big). The objects within this group will have different

membership values varying from 0 to 1 qualifying the degree to which they satisfy the concept 'big'. An object with membership of 0.8 is more likely to be described as 'big' than an object with membership of 0.4 [Zen05].

## 2.12 Fuzzy Logic with Data Mining

Data mining is the process of extracting nontrivial relationships in the database. However, associations that are qualitative are very difficult to utilize effectively by applying conventional rule induction algorithms. Since fuzzy logic modeling is a probability based modeling, it has many advantages over the conventional rule induction algorithms. The first advantage is that it allows processing of very large data sets which require efficient algorithms. Fuzzy logic-based rule induction can handle noise and uncertainty in data values well. Most of the databases, in general, are not designed or created for data mining. Selecting and extracting useful attributes of target objects becomes hard. Not all of the attributes needed for successful extraction can be contained in the database. In this case, domain knowledge and user analysis becomes a necessity. In these cases, techniques such as neural networks tend to do badly since the domain knowledge cannot be incorporated into the neural networks. Fuzzy logic based models utilize the domain knowledge in coming up with rules of data selection and extraction. An example of a fuzzy logic based data mining algorithm is "if 'age' is 'young' and 'sex' is 'female' then 'probability of purchase is high"[Rah98].

Fuzzy logic provides essential tool to utilize qualitative knowledge in the data mining process, it allows a focused search in the database that can

be defined "qualitatively". It also defines the associations among objects within the data set which can be expressed in a qualitative format [Rah98].

### **2.13 Mining Fuzzy Association Rules**

Association Rule Mining is an important and well established data mining topic. The objective of association rule mining is to identify patterns expressed in the form of Association Rules in transaction data sets. The attributes in association rule mining data sets are usually binary valued but association rule mining has also been applied to quantitative and categorical (non-binary) data. With the latter, values can be split into linguistically labeled ranges (for example "low", "medium", "high" etc) such that each range represents a binary valued attribute. Values can be assigned to these range attributes using crisp or fuzzy boundaries. The application of association rule mining using the latter is referred to as fuzzy association rule mining. Fuzzy ARM has been shown to produce more expressive association rules than the "crisp" methods [Sul08].

Association rule mining searches for interesting relationship among items in a large data set. Market basket analysis, a typical example of association rule mining, analyzes buying habit of customers by finding association between the different items that customers put in their shopping cart (basket) [Rol06].

### **2.14 Association Rule Algorithm Principles**

An association is said to exist between two sets of items when a transaction containing one set is likely to also contain the other. One example is the analysis of supermarket basket data where associations like



“28% of all customers who buy cheese also buy milk” may be discovered or mined. An association is a rule latent in and possibly mined from databases, by which one attribute set can be inferred from another [Dar05].

Association rules can be viewed as a technique to retrieve patterns from very large databases, for the purpose of obtaining useful information. Association-rule mining focuses on finding rules that have a minimum specified support and confidence level. The ‘support’ is the amount of times the rule appears with respect to the entire datasets, while ‘confidence’ is the degree or measure of times the consequent occurs with respect to the antecedents. Association-rule mining can be divided into two phases: picking out all items (itemsets) that have a support level above the minimum required support level, and discovering interesting rules from such itemsets [Zim02].

Association rules extracts relationships between attributes in datasets which may not have class labels. Association rule extraction techniques are usually used to discover relationships between items in transaction data [Lip05].

An association rule is a rule which implies certain association relationships among a set of objects (such as “occur together” or “one implies the other”) in a database. An association rule is an expression of the form  $X \Rightarrow Y$ , where  $X = \{x_1, x_2, \dots, x_n\}$  and  $Y = \{y_1, y_2, \dots, y_m\}$  are sets of items,  $X$  is called the antecedent,  $Y$  is called the consequent. The antecedent and consequent frequently are called, respectively, the Left-Hand Side (or LHS) and the Right-Hand Side (or RHS). The intuitive meaning of such rule is that transactions of the database, which contains  $X$ , tends to contain  $Y$  [Pie98, Yik02].

Association rules identify relationships between attributes and items in a database such as the presence or absence of one pattern implies the presence or absence of another pattern, mining of such rules is one of the most popular pattern discovery methods in KDD. The mining of such rules is quite intuitive: given database  $D$  of transaction  $T$  where each transaction  $T \in D$  is a set of items  $X \Rightarrow Y$  which expresses that whenever a transaction  $T$  contains  $X$ , the  $T$  probably contains  $Y$ . Also the probability of rule strength is defined as the percentage of transactions containing  $Y$  in addition to  $X$ . The prevalence of rule is the percentage of transactions that hold all the items in the union. If prevalence is low, it implies that there is no overwhelming evidence that items in  $X \cap Y$  occur together. The rule  $X \Rightarrow Y$  has support  $S$  in  $D$  if the fractions of the transactions in  $D$  contain  $(X \cap Y)$ . The problem of mining association rules is to generate all association rules that have certain user-specified minimum support called (min-sup) and confidence called (min-conf) [Rak96].

The **support** ( $S$ ) of an association rule is the ratio (in percent) of the records that contain  $(X \cap Y)$  to the total number of the records in database. Therefore, if the support of a rule is 5% then it means that 5% of the total records contain  $(X \cap Y)$ . Support is the statistical significant of a rule [Ser97].

Support  $(X \Rightarrow Y) = \text{frequent}(X) / \text{total number of records in the database}$   
(equation 2.4)

The **confidence** ( $C$ ) of a rule indicate the degree of correlation in the database between  $X$  and  $Y$ .

Confidence ( $X \Rightarrow Y$ ) = frequent ( $X \cap Y$ )/frequent( $X$ ) (equation 2.5)

**Confidence** is also a measure of rules strength. Mining consists of finding all rules that meet the user-specified threshold support and confidence.

## 2.15 Apriori Algorithm

Apriori is an influential algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level-wise search, where  $k$ -itemsets are used to explore  $(k+1)$ -itemsets. First, the set of frequent 1-itemsets is found. This set is denoted  $L_1$ .  $L_1$  is used to find  $L_2$ , the frequent 2-itemsets, which is used to find  $L_3$ , and so on, until no more frequent  $k$ -itemsets can be found. The finding of each  $L_k$  requires one full scan of the database.

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

**The Apriori property.** All non-empty subsets of a frequent itemset must also be frequent; this property is based on the following observation: If an itemset  $I$  does not satisfy the minimum support threshold,  $S$ , then  $I$  is not frequent, i.e.,  $\text{Prob}\{I\} < S$ . If an item  $A$  is added to the itemset  $I$ , then the resulting itemset (i.e.,  $I \cap A$ ) cannot occur more frequently than  $I$ . Therefore,  $I \cap A$  is not frequent either, i.e.,  $\text{Prob}\{I \cap A\} < S$ .

This property belongs to a special category of properties called anti-monotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well. It is called anti-monotone because the property is monotonic in the context of a failing test [Jia00].

In general, to find  $L_k$ , a two-step process is followed, consisting of join and prune actions [Jia06]:

1. **The join step:** To find  $L_k$ , a set of candidate  $k$ -itemsets is generated by joining  $L_{k-1}$  with itself. This set of candidates is denoted  $C_k$ . Let  $l_1$  and  $l_2$  be itemsets in  $L_{k-1}$ . The notation  $l_i[j]$  refers to the  $j$ th item in  $l_i$  (e.g.,  $l_1[k-2]$  refers to the second to the last item in  $l_1$ ). By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the  $(k-1)$ -itemset,  $l_i$ , this means that the items are sorted such that  $l_i[1] < l_i[2] < \dots < l_i[k-1]$ . The join,  $L_{k-1}$  on  $L_{k-1}$ , is performed, where members of  $L_{k-1}$  are joinable if their first  $(k-2)$  items are in common. That is, members  $l_1$  and  $l_2$  of  $L_{k-1}$  are joined if  $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ . The condition  $l_1[k-1] < l_2[k-1]$  simply ensures that no duplicates are generated. The resulting itemset formed by joining  $l_1$  and  $l_2$  is  $l_1[1], l_1[2], \dots, l_1[k-2], l_1[k-1], l_2[k-1]$ .
2. **The prune step:**  $C_k$  is a superset of  $L_k$ , that is, its members may or may not be frequent, but all of the frequent  $k$ -itemsets are included in  $C_k$ . A scan of the database to determine the count of each candidate in  $C_k$  would result in the determination of  $L_k$  (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to  $L_k$ ).  $C_k$ , however, can be huge, and so this could involve heavy computation. To reduce the size of  $C_k$ , the Apriori property is used as follows. Any  $(k-1)$ -itemset that is not

frequent cannot be a subset of a frequent  $k$ -itemset. Hence, if any  $(k-1)$ -subset of a candidate  $k$ -itemset is not in  $L_{k-1}$ , then the candidate cannot be frequent either and so can be removed from  $C_k$ .

*Chapter Three*

*Implementation of  
a Fuzzy Apriori  
System*

# Chapter Three

## Implementation of a Fuzzy Apriori System

### 3.1 Introduction

Association rules are one of the promising aspects of data mining as knowledge discovery tool and have been widely explored to date, they allow to capture all possible rules that explain the presence of some attributes according to the presence of other attributes. This chapter will explain apriori association rule in detail with example, and two methods are used to implement fuzzy apriori association rule principles.

### 3.2 Apriori Algorithm

Algorithm (3-1) shows the Apriori algorithm steps, it has two sub procedures `apriori_gen` procedure and `has_infrequent_subset` in the `apriori_gen` procedure

**Algorithm (3-1)** Apriori Find Frequent itemset using an iterative level-wise approach based on candidate generation.

**Input:** database  $D$  of transaction; minimum support threshold,  $min\_sup$ .

**Output:** Association Rules.

continue

**Begin**

$L_1 = \text{find\_frequent\_1\_itemset}(D);$

**For** ( $K=2; L_{K-1} \neq \emptyset; K++$ )

$C_k = \text{apriori\_gen}(L_{K-1}, \text{min\_sup});$

**For** each transaction  $t \in D$  //scan  $D$  counts

$C_t = \text{subset}(C_k, t);$  //get the subsets of  $t$  that are candidates

**For** each candidate  $c \in C_t$

$c.\text{count}++;$

$L_k = \{c \in C_k \mid c.\text{count} \geq \text{min\_sup}\}$

return  $L = \bigcup_k L_k;$

**End.**

**Procedure apriori\_gen**

**Input:**  $L_{k-1}$ : frequent( $k-1$ )-itemsets;  $\text{min\_sup}$ : minimum support

**Output:**  $C_k$

**Begin**

**For** each itemset  $I1 \in L_{k-1}$

**For** each itemset  $I2 \in L_{k-1}$

**IF** ( $I1[1] = I2[1] \wedge I1[2] = I2[2] \wedge \dots \wedge I1[k-2] = I2[k-2] \wedge I1[k-1] <$

$I2[k-1]$ ) then  $c = I1 \text{ join } I2;$  //join step generate candidates

**IF**  $\text{has\_infrequent\_subset}(c, L_{k-1})$  then

Delete  $c;$  //prune step: remove unfruitful candidate

**Else** add  $c$  to  $C_k;$

Return  $C_k$

**End.**



**Procedure has\_infrequent\_subset**

**Input:**  $c$ : candidate  $k$ -itemset;  $L_{k-1}$ : frequent  $(k-1)$ -itemsets); // use prior knowledge

**Output:** flag

**Begin**

**For** each  $(k - 1)$ -subset  $s$  of  $c$

    if  $s \notin L_{k-1}$  then

      return TRUE;

**3.3 Example:**

Let's look at a concrete example of Apriori, based on transaction database,  $D$ , of table (3-1). There are four transactions in this database, i.e.,  $|D| = 4$ . Use this data to illustrate the Apriori algorithm for finding frequent itemsets in  $D$  to get association rules.

**Table (3-1) Transactional Data**

TID	List of item_IDs
T1	I1,I2,I5
T2	I2,I3,I4
T3	I3,I4
T4	I1,I2,I3,I4

**First: the Apriori Algorithm finds frequent itemsets**

In the first iteration of the algorithm, the algorithm simply scans all of the transactions in order to count the number of occurrences of each item; each item is a member of the set of candidate 1-itemsets,  $C_1$ .

**Table (3-2) C1**

Itemset	Sup.
I1	2
I2	3
I3	3
I4	3
I5	1

Suppose that the minimum transaction support count required is 2 (i.e., min sup = 50%).

**Table (3-3) L1**

Itemset	Sup.
I1	2
I2	3
I3	3
I4	3

The set of frequent 1-itemsets, L1, can then be determined. It consists of the candidate 1-itemsets having minimum support.

To discover the set of frequent 2-itemsets, L2, the algorithm uses L1 join L1 to generate a candidate set of 2-itemsets

Table C2 is constructed; it consists of set of 2- itemsets by joining L1 with itself.

Next, the transactions in the Database (D) are scanned and the support count of each candidate itemset in C2 is accumulated, as shown in Table (3-4).

**Table (3-4) C2**

Itemset	Sup.
I1,I2	2
I1,I3	1
I1,I4	1
I2,I3	2
I2,I4	2
I3,I4	3

The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 satisfying minimum support

**Table (3-5) L2**

Itemset	Sup.
I1,I2	2
I2,I3	2
I2,I4	2
I3,I4	3

The generation of the set of candidate 3-itemsets, C3, is detailed in **Table (3-6)** bellow. First, let  $C3 = L2 \text{ join } L2 = \{I1; I2; I3\}; \{I1; I2; I4\}; \{I2; I3; I4\}$ . Based on the Apriori property that all subsets of a frequent itemset must also be frequent, it can be determined that the candidates  $\{I1, I2, I3\}$  and  $\{I1, I2, I4\}$  cannot possibly be frequent. Therefore they will be removed from C3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D to determine L3. Note that since the Apriori algorithm uses a level-wise search strategy, then given a k-itemset, it's only needed to check if its (k-1)-subsets are frequent.

**Table (3-6) C3**

Itemset	Sup
I1,I2,I3	1
I1,I2,I4	1
I2,I3,I4	2

The transactions in D are scanned in order to determine L3, consisting of those candidate 3-itemsets in C3 having minimum support above 2, as shown in Table (3-7).

**Table (3-7) C3**

Itemset	Sup
I2,I3,I4	2

No more frequent itemsets can be found (since here,  $C4 = \emptyset$ ), and so the algorithm terminates, having found all of the frequent itemsets, table L3 is the same as table C3.

### **Second: Generating Association Rules from Frequent Itemsets**

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

$$\text{Confidence } (A \Rightarrow B) = P(B | A) = \text{support}(A \cup B) / \text{support}(A) \text{ (equation 3.1)}$$

Where  $\text{support}(A \cup B)$  is the number of transactions containing the itemsets  $A \cup B$ , and  $\text{support}(A)$  is the number of transactions containing the itemset A.

Based on this equation, association rules can be generated as follows:

- For each frequent itemset, I, generate all non-empty subsets of I.
- For every non-empty subset sub, of I, output the rule “s  $\Rightarrow$  (I - sub)” if  $\text{support}(I)/\text{support}(\text{sub}) \geq \text{min\_conf}$ , where min\_conf is the minimum confidence threshold.

**Example:**

This example based on the transactional data for table (3-1). Suppose the data contains the frequent itemset  $I = \{I_2, I_3, I_4\}$ , what are the association rules that can be generated from I? The non-empty subsets of I are  $\{I_2, I_3\}$ ,  $\{I_2, I_4\}$ ,  $\{I_3, I_4\}$ ,  $\{I_2\}$ ,  $\{I_3\}$ , and  $\{I_4\}$ . The resulting association rules are as shown below, each listed with its confidence.

$I_2 \wedge I_3 \Rightarrow I_4$ , confidence =  $2/2 = 100\%$  because support of  $I_2 \& I_3 = 2$  (from table (3-5)) and support of  $I_3 = 2$  then  $2/2 = 100\%$ , and it will be implemented on the rest of the following rules.

$I_2 \wedge I_4 \Rightarrow I_3$ , confidence =  $2/2 = 100\%$

$I_3 \wedge I_4 \Rightarrow I_2$ , confidence =  $2/3 = 67\%$

$I_2 \Rightarrow I_3 \wedge I_4$ , confidence =  $2/3 = 67\%$

$I_3 \Rightarrow I_2 \wedge I_4$ , confidence =  $2/3 = 67\%$

$I_4 \Rightarrow I_2 \wedge I_3$ , confidence =  $2/3 = 67\%$

If the minimum confidence threshold is, say, 70%, then only the first and second rules above are output, since these are the only ones generated that are strong.

### 3.4 Fuzzy Apriori Algorithm 1

To mine the fuzzy association rule, first find out all sets of items that have transaction support above minimum support. Itemsets with minimum support are called frequent itemsets. The fuzzy support and confidence value is computed by the following equations[Ger02]:

$$NewMinSup = (1 - MinSup) * \frac{numberoftransactions - currenttransactions}{numberoftransactions} * k + Min\_Sup$$

(equation 3.2)

$$NewMinConf = (1 - MinConf) * \frac{numberoftransactions - currenttransactions}{numberoftransactions} * k + Min\_Conf$$

(equation 3.3)

Where: Min\_Sup is Minimum Support, Min\_Conf is Minimum Confidence, numberoftransactions is the total number of transactions in the database, currenttransactions is the number of itemsets in the LHS and RHS of the rule (i.e. the antecedent and consequent of the rule), K is user threshold used to reduce the value of NewMinSup and NewMinConf.

If K is equal to 0, all the rules will have the same confidence and support (i.e. the basic Apriori algorithm). If K = 1, then rules with few itemsets in its LHS and RHS must have much more confidence and support to be selected by the algorithm.

For example if there is 11 transactions and 3 items in the database and If min\_Sup = 0.2 and min\_Conf = 0.8, then with the basic Apriori algorithm (i.e. with K=0) this rule would be selected. However, by using K=0.5 for example, this rule does not satisfy the new Apriori constraints:

$$\text{NewMinSup} = (1-0.2) \frac{11-3}{11} (0.5) + 0.2 = 0.49$$

$$\text{NewMinConf} = (1-0.8) \frac{11-3}{11} (0.5) + 0.8 = 0.872$$

By taking the first example explained previously, the construction of C tables and L tables will be depending on the new computed support value (Min\_Sup), different number of association rules will be presented depending on the new computed minimum confidence (Min\_Conf).

Algorithm (3-2) shows the first proposed Fuzzy Apriori Algorithm steps. It has two sub procedures apriori\_gen procedure and has\_infrequent\_subset in the apriori\_gen procedure:

**Algorithm (3-2)** FuzzyApriori1 Find Frequent itemset using an iterative level-wise approach based on candidate generation, new minimum support and new minimum confidence.

**Input:** database  $D$  of transaction; minimum support threshold,  $min\_sup$ , user threshold  $K$ .

**Output:** Fuzzy Association Rules.

**Begin**

$L_1 = \text{find\_frequent\_1\_itemset}(D)$ ;

**For** ( $K=2$ ;  $L_{K-1} \neq \emptyset$ ;  $K++$ )

$C_k = \text{apriori\_gen}(L_{K-1}, min\_sup)$ ;

**For** each transaction  $t \in D$  //scan  $D$  counts

compute new fuzzy support  $new\_min\_sup$

compute new fuzzy confidence  $new\_min\_conf$

continue

```

For each candidate  $c \in C_t$ 
     $c.count = c.count + new\_min\_sup$ ;
 $L_k = \{c \in C_k \mid c.count \geq min\_sup\}$ 
return  $L = U_k L_k$ ;

```

**End.**

The following procedure explains apriori\_gen procedure of Apriori algorithm

**Procedure apriori\_gen**

**Input:** ( $L_{k-1}$ : frequent( $k-1$ )-itemsets;  $min\_sup$ : minimum support)

**Output:**  $C_k$

**Begin**

```

For each itemset  $I1 \in L_{k-1}$ 
    For each itemset  $I2 \in L_{k-1}$ 
        IF ( $I1[1] = I2[1] \wedge I1[2] = I2[2] \wedge \dots \wedge I1[k-2] = I2[k-2] \wedge I1[k-1] <$ 
             $I2[k-1]$ ) then  $c = I1 \text{ join } I2$ ; //join step generate candidates
        IF  $has\_infrequent\_subset(c, L_{k-1})$  then
            Delete  $c$ ; //prune step: remove unfruitful candidate
        Else add  $c$  to  $C_k$ ;
Return  $C_k$ 

```

**End.**



The following procedure explains has\_infrequent\_subset procedure

<p><b>Procedure has_infrequent_subset</b></p> <p><b>Input:</b> (<i>c</i>: candidate <i>k</i>-itemset; <math>L_{k-1}</math>: frequent (<i>k</i>-1)-itemsets); // use prior knowledge</p> <p><b>Output:</b> flag</p>
<p><b>Begin</b></p> <p>  <b>For</b> each (<i>k</i> - 1)-subset <i>s</i> of <i>c</i></p> <p>    if <math>s \notin L_{k-1}</math> then</p> <p>      return TRUE;</p> <p>  return FALSE;</p> <p><b>End.</b></p>

### 3.5 Fuzzy Apriori Algorithm 2

First of all, the transaction database is filtered into M database of transactions depending on a user threshold. The construction of  $C_i$  and  $L_i$ ,  $\{i=1..6\}$  tables are the same as explained previously except that there is minimum support value  $\beta$  for each level used for comparison with item support, item support value is calculated for each item in its transaction then final item support is computed by summing all support values of all transactions. Algorithm (3-3) illustrates the second proposed Fuzzy Apriori Algorithm

<p><b>Algorithm (3-3)</b> FuzzyApriori2 Find Frequent itemset using an iterative level-wise approach based on candidate generation, new minimum support and new minimum confidence.</p> <p><b>Input:</b> database <i>D</i> of transaction; minimum support threshold, <math>\beta</math>, user threshold <i>K</i>, maximum item threshold, <math>\delta</math></p> <p style="text-align: right;">continue</p>
---

**Output:** Fuzzy Association Rules.

**Step 1:** Convert all transactions in database D into M; M is filtered from D where number of items in its transactions is no greater than  $\delta$

**Step 2:** Set  $k=1$  where  $k$  is an index variable to determine the number of combination items in itemsets called  $k$  itemsets.

**Step 3:** Determine minimum support for  $k$ -itemsets, denoted by  $\beta_k$

**Step 4:** Compute fuzzy support according to its appearance in its transaction.

**Step 5:** Compute final fuzzy support by summing all fuzzy support for all transactions.

**Step 6:**  $I_k$  will be stored in the set of frequent  $k$ -itemsets,  $L_k$  if and only if  $support(I_k) \geq \beta_k$ .

**Step 7:** Set  $k=k+1$ , and if  $k > \delta$ , then go to Step 9.

**Step 8:** Looking for possible candidate  $k$ -itemsets from  $L_{k-1}$  by the following rules: A  $k$ -itemset,  $I_k$ , will be considered as a candidate  $k$ -itemset if and only if  $I_k$  satisfied:  $\forall F \subset I_k, |F| = k - 1 \Rightarrow F \in L_{k-1}$  If there is not found any candidate  $k$ -itemset then go to Step 9. Otherwise, the process is going to Step 3.

**Step 9:** Confidence value is computed same way as Apriori algorithm.

### Example:

**Step 1:** Table (3-9) represents a transactional database D, suppose  $\delta = 3$ , A Qualified Data Transaction (M) is constructed by filtering transactions in Table (3-9) where number of items in its transactions is no greater than  $\delta$

**Table (3-8) Transactional Data**

TID	List of item_IDs
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3
T10	I1,I2,I3,I6

**Table (3-9) A Qualified Data Transaction (M)**

TID	List of item_IDs
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T9	I1,I2,I3

**Step 2:** The process is started by looking for support of 1-itemsets for which  $k$  is set equal to 1.

**Step 3:** Since  $\delta=3$ , then  $k \in \{1,2,3\}$ . It is arbitrarily given  $\beta_1=\beta_2=0.5$ ,  $\beta_3=0.2$ . That means the system just considers support of  $k$ -itemsets that is greater than 0.5, for  $k=1,2$ , and greater than 0.2, for  $k=3$ .

**Step 4:** Compute fuzzy support:

**1-itemsets:**

$$\{i1\}=\{0.33/T1, 0.33/T4, 0.5/T5, 0.5/T7, 0.33/T9\},$$

Explanation:

$i1$  in  $T1$  appears three times, then  $\frac{1}{3} = 0.33$ ,  $i1$  in  $T4$  appears three times, then

$\frac{1}{3} = 0.33$ ,  $i1$  in  $T5$  appears two times, then  $\frac{1}{2} = 0.5$ ,  $i1$  in  $T7$  appears two

times, then  $\frac{1}{2} = 0.5$ ,  $i1$  in  $T9$  appears three times, then  $\frac{1}{3} = 0.33$ .

And the same calculations will be applied to the rest of items:

$$\{i2\}=\{0.33/T1, 0.5/T2, 0.5/T3, 0.33/T4, 0.5/T6, 0.33/T9\},$$

$$\{i3\}=\{0.5/T3, 0.5/T5, 0.5/T6, 0.5/T7, 0.33/T9\},$$

$$\{i4\}=\{0.5/T2, 0.33/T4\},$$

$$\{i5\}=\{0.33/T1\}.$$

From Step-5 and Step-6 of algorithm (3-3),  $\{i5\}$  cannot be considered for further process because  $\text{sup}(\{i5\}) < 0.5$ .

**2-itemsets:**

$$\{i1, i2\}=\{0.33/T1, 0.33/T4, 0.33/T9\},$$

$$\{i2, i4\}=\{0.5/T2, 0.33/T4\},$$

$$\{i2, i3\}=\{0.5/T3, 0.5/T6, 0.33/T9\},$$

$$\{i1, i4\}=\{0.33/T4\},$$

$\{i1, i3\} = \{0.5/T5, 0.5/T7, 0.33/T9\}$ .

From Step-5 and Step-6 of algorithm (3-3),  $\{i1, i4\}$  cannot be considered for further process because  $\text{sup}(\{i1, i4\}) < 0.5$ .

### **3-itemsets:**

$\{i1, i2, i3\} = \{0.33/T9\}$ .

**Step 5:** Support of each k-itemset is calculate as given in the following results:

<u>1-itemsets</u>	<u>2-itemsets</u>	<u>3-itemsets</u>
support( $\{i1\}$ ) = 1.99,	support( $\{i1, i2\}$ )=0.99,	support( $\{i1, i2, i3\}$ )=0.33
support( $\{i2\}$ ) = 2.49,	support( $\{i2, i4\}$ )=0.83	
support( $\{i3\}$ ) = 2.33,	support( $\{i2, i3\}$ )=1.33	
support( $\{i4\}$ ) = 0.83,	support( $\{i1, i4\}$ )=0.33	
support( $\{i5\}$ ) = 0.33,	support( $\{i1, i3\}$ )=1.33	

Table (3-10) L1

1-Itemsets	Sup.
I1	1.99
I2	2.49
I3	2.33
I4	0.83

Table (3-11) L2

2-Itemsets	Sup.
I1,I2	0.99
I2,I4	0.83
I2,I3	1.33
I1,I3	1.33

Table (3-12) L3

3-Itemsets	Sup.
I1,I2,I3	0.33

**Step-6:**

From the results as performed by Step-4 and 5, the sets of frequent 1-itemsets, 2-itemsets and 3-itemsets are given in Table (3-11), (3-12) and (3-13), respectively.

**Step 7:** Set  $k=k+1$ , and if  $k > \delta$ , then go to Step 9.

Step-8: This step is looking for possible/candidate k-itemsets from  $L_{k-1}$ . If there is no any more candidate k-itemset then go to Step-9. Otherwise, the process is going to Step-3.

**Step-9:** The step is to calculate every confidence of each possible association rules as follows:

$$\text{Conf}(i_1 \Rightarrow i_2) = \frac{0.99}{1.99} = 0.5,$$

$$\text{Conf}(i_2 \Rightarrow i_4) = \frac{0.83}{2.49} = 0.33,$$

•

•

•

$$\text{Conf}(i_1 \Rightarrow i_2, i_3) = \frac{0.33}{1.99} = 0.17,$$

•

•

•

# *Chapter Four*

## *Experiments and Results*

# Chapter Four

## Experiments and Results

### 4.1 Introduction

In this chapter, the work of the data mining system is explained with experimental results; the system was implemented using Microsoft access and is connected with visual basic 6.

### 4.2 System Interface

Figure (4.1) illustrates the data mining system to find the associations among items in the supermarket, it have the items table to indicate the items in the supermarket, the transaction table to indicate the buying menus of the customers.

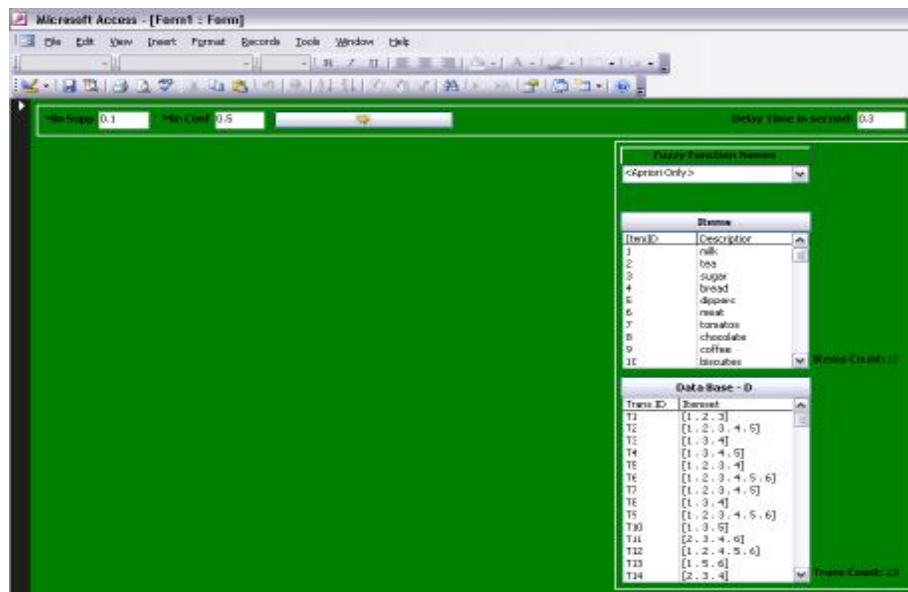


Figure (4.1) main form





Figure (4.2) Min Supp, Min Conf, Find Ass. rules

Where Min Supp refers to minimum support threshold, Min Conf refers to minimum confidence threshold. Both of them are specified by the user, Find Association Rules is a command of execution.

Items	
ItemID	Description
1	milk
2	tea
3	sugar
4	bread
5	dippers
6	meat
7	tomatos
8	chocolate
9	coffee

Items Count: 12

Figure (4.3) items table

The (Items) table represents the set of items available in the supermarket.

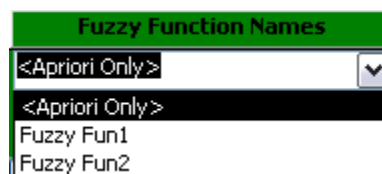


Figure (4.4) fuzzy function names

Figure (4.4) presents Apriori algorithm only and fuzzy functions with Apriori algorithm to make new calculations.

(Data Base-D) table which represents the transactions between items, i.e. the items that purchased together by a customer (baskets), it's represented by numbers for simplicity, for example: the first transaction means that milk, tea and sugar are purchased together by a customer, transaction two means that milk, tea, sugar, bread and dippers are purchased together.

Trans ID	Itemset
T1	[1 . 2 . 3]
T2	[1 . 2 . 3 . 4 . 5]
T3	[1 . 3 . 4]
T4	[1 . 3 . 4 . 5]
T5	[1 . 2 . 3 . 4]
T6	[1 . 2 . 3 . 4 . 5 . 6]
T7	[1 . 2 . 3 . 4 . 5]
T8	[1 . 3 . 4]
T9	[1 . 2 . 3 . 4 . 5 . 6]
T10	[1 . 3 . 5]
T11	[2 . 3 . 4 . 6]
T12	[1 . 2 . 4 . 5 . 6]
T13	[1 . 5 . 6]
T14	[2 . 3 . 4]

Figure (4.5) Data Base - D

### 4.2.1 Implementing Apriori Algorithm

From the selection shown in figure (4.4) a user can choose <Apriori Only> to run Apriori algorithm only.

If the user enters minimum support value=0.9 in the (Min Sup) button, and minimum confidence value=0.8 in the (Min Conf) button then no association rules occurs because no item have minimum support equal or greater than 0.9

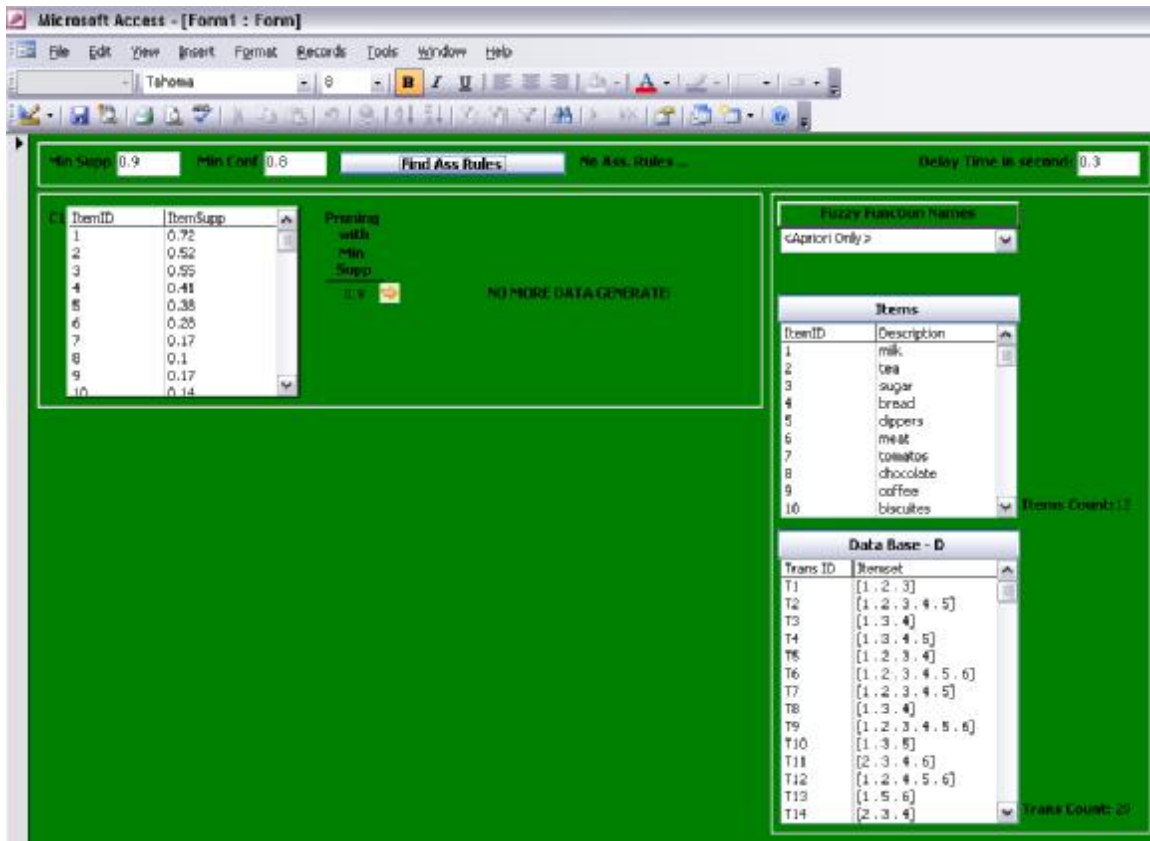
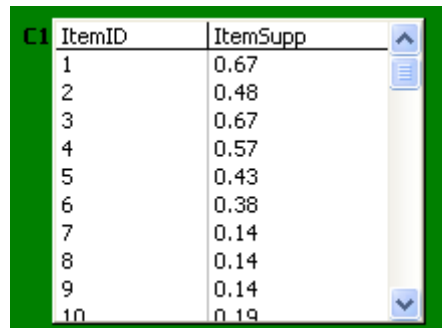


Figure (4.6)

If the user enters minimum support value=0.2 in the (Min Sup) button, and minimum confidence value=0.8 in the (Min Conf) button, and then by clicking on (Find Ass Rules) button of figure (4.2), the system will reach to level 5 of execution here (K=5) as follows:

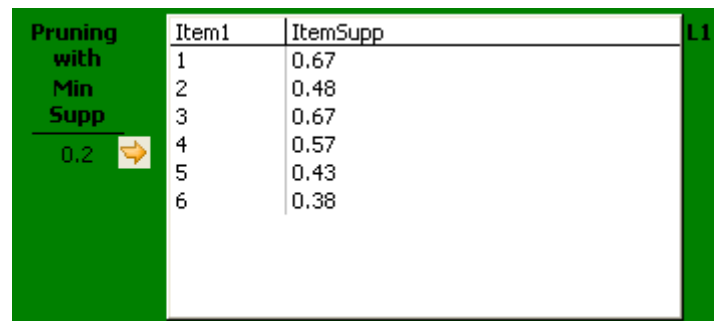
1. Generating table C1: by scanning all transactions in the database then C1 has two fields: ItemID field (which represents all items exists in the database), and the ItemSupp field (which represents the number of times that the item occurs in the database)



ItemID	ItemSupp
1	0.67
2	0.48
3	0.67
4	0.57
5	0.43
6	0.38
7	0.14
8	0.14
9	0.14
10	0.19

Figure (4.7)

2. Generating table L1: by scanning table C1, L1 contains only the items having ItemSupp above or equal (Min Supp) predetermined previously, i.e. only the items with support greater than or equal to 0.2 remains in L1, all items having ItemSupp less than 0.2 are pruned

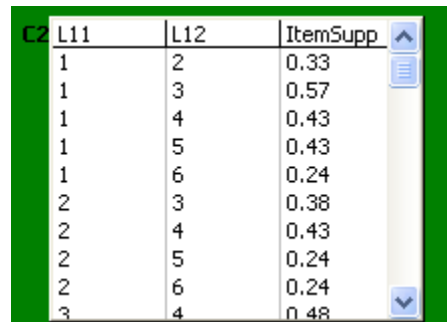


Item1	ItemSupp
1	0.67
2	0.48
3	0.67
4	0.57
5	0.43
6	0.38

Pruning with Min Supp 0.2

Figure (4.8)

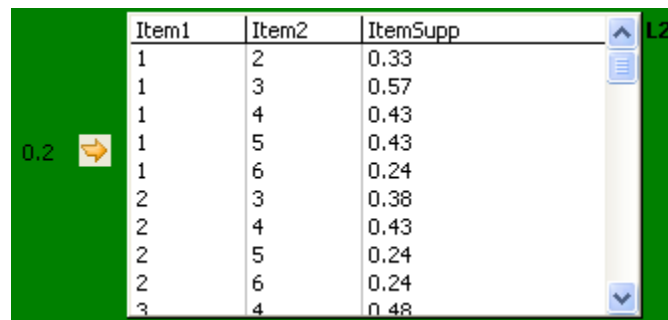
3. Generating table C2: by joining L1 with L1 without repeating the items, the result is a set of two items. The joining is as follows: join first item with the second then the first with the third then... till the end of items, and now join the second item with the third...till the end of items, and so on until all the items completed, ItemSupp is counted by referencing to the database and counting the number of times the two item set occurs in the database



C2	L11	L12	ItemSupp
1	2	0.33	
1	3	0.57	
1	4	0.43	
1	5	0.43	
1	6	0.24	
2	3	0.38	
2	4	0.43	
2	5	0.24	
2	6	0.24	
3	4	0.48	

Figure (4.9)

4. Generating table L2: by scanning table C2, L2 contains only the set of items having ItemSupp above or equal (Min Supp) predetermined previously, i.e. only the items with support greater than or equal to 0.2 remains in L2, all items having ItemSupp less than 0.2 are pruned



Item1	Item2	ItemSupp
1	2	0.33
1	3	0.57
1	4	0.43
1	5	0.43
1	6	0.24
2	3	0.38
2	4	0.43
2	5	0.24
2	6	0.24
3	4	0.48

Figure (4.10)

5. Generating table C3: by joining L2 with L2 without repeating the items, the result is three item sets. The joining is as follows: join first two item set with the second two item sets in a condition that they have one common element then the first two item sets with the third then... till the end of items, and now join the second two item sets with the third...till the end of items, and so on until all the items completed, ItemSupp is counted by referencing to the database and

counting the number of times the three item sets occurs in the database

C3	L21	L22	L23	ItemSupp
1	2	3	0.29	
1	2	4	0.29	
1	2	5	0.24	
1	2	6	0.14	
1	3	4	0.38	
1	3	5	0.33	
1	3	6	0.14	
1	4	5	0.29	
1	4	6	0.14	
1	5	6	0.19	

Figure (4.11)

6. Generating table L3: by scanning table C3, L3 contains only the set of items having ItemSupp above or equal (Min Supp) predetermined previously, i.e. only the items with support greater than or equal to 0.2 remains in L3, all items having ItemSupp less than 0.2 are pruned

Item1	Item2	Item3	ItemSupp
1	2	3	0.29
1	2	4	0.29
1	2	5	0.24
1	3	4	0.38
1	3	5	0.33
1	4	5	0.29
2	3	4	0.33
2	4	5	0.24
2	4	6	0.24
3	4	5	0.24

Figure (4.12)

7. Generating table C4: by joining L3 with L3 without repeating the items, the result is four item sets. The joining is as follows: join first three item set with the second three item set in a condition that they have two common element then the first three item set with the third

then... till the end of items, and now join the second three item set with the third...till the end of items, and so on until all the items completed, ItemSupp is counted by referencing to the database and counting the number of times the four item set occurs in the database

C4	L31	L32	L33	L34	ItemSupp
	1	2	3	4	0.24
	1	2	3	5	0.19
	1	2	4	5	0.24
	1	3	4	5	0.24
	2	4	5	6	0.14

Figure (4.13)

8. Generating table L4: by scanning table C4, L4 contains only the set of items having ItemSupp above or equal (Min Supp) predetermined previously, i.e. only the items with support greater than or equal to 0.2 remains in L4, all items having ItemSupp less than 0.2 are pruned

Item1	Item2	Item3	Item4	ItemSupp	L4
1	2	3	4	0.24	
1	2	4	5	0.24	
1	3	4	5	0.24	

0.2 →

Figure (4.14)

Now the association rules are generated are 42 rules, each rule has left hand side and right hand side and a confidence, The confidence of an

association rule is counted by dividing the whole transaction by the left hand side of the association rule

Total Ass. Rules Generated: 42			
Left HS		Right HS	Conf.
[1]	--->	[2, 3, 4]	0.36
[2]	--->	[1, 3, 4]	0.5
[3]	--->	[1, 2, 4]	0.36
[4]	--->	[1, 2, 3]	0.42
[1, 2]	--->	[3, 4]	0.73
[1, 3]	--->	[2, 4]	0.42
[1, 4]	--->	[2, 3]	0.56
[2, 3]	--->	[1, 4]	0.63
[2, 4]	--->	[1, 3]	0.56
[3, 4]	--->	[1, 2]	0.5
[1, 2, 3]	--->	[4]	0.83
[1, 2, 4]	--->	[3]	0.83
[1, 3, 4]	--->	[2]	0.63
[2, 3, 4]	--->	[1]	0.73
[1]	--->	[2, 4, 5]	0.36
[2]	--->	[1, 4, 5]	0.5

Figure (4.15)

Now from the 42 association rules generated the obtained rules are the ones which have confidence above the (Min Conf) predetermined previously, i.e. only the rules having minimum confidence above 0.8 remains, all other rules are pruned so the resulted rules are only eleven rules, as shown in figure (4.16).

Ass. Rules with Min Conf.: 0.8			
No.	Left HS	Right HS	Conf.
1	[1, 2, 3]	---> [4]	0.83
2	[1, 2, 4]	---> [3]	0.83
3	[2, 5]	---> [1, 4]	1
4	[4, 5]	---> [1, 2]	0.83
5	[1, 2, 4]	---> [5]	0.83
6	[1, 2, 5]	---> [4]	1
7	[1, 4, 5]	---> [2]	0.83
8	[2, 4, 5]	---> [1]	1
9	[4, 5]	---> [1, 3]	0.83
10	[1, 4, 5]	---> [3]	0.83
11	[3, 4, 5]	---> [1]	1

Figure (4.16)



### 4.2.2 Implementing Apriori Algorithm (with fuzzy function 1)

The user chooses Fuzzy Apriori F1 and enters a user threshold number in the (User Thr.) box, as shown in Figure (4.17)

Figure (4.17)

If the user enters a zero in the (User Thr.) box and enters the same values for (Min Sup) and (Min Conf), and clicks on (Find Ass Rules) of figure (4.2), the results obtained are the same as the previous execution, this means that fuzzy function did not work.

Now if the user enters 0.3 in the (User Thr.) box and enters the same values for (Min Sup) and (Min Conf), then clicks on (Find Ass Rules), the system will reach to level 3 of execution ( $K=3$ ). The construction of C1 and L1 is the same as explained previously, as shown in Figure (4.18).

ItemID	ItemSupp
1	0.67
2	0.48
3	0.67
4	0.57
5	0.43
6	0.38
7	0.14
8	0.14
9	0.14
10	0.19

**Pruning with New Min Sup**  
0.42

Item1	ItemSupp
1	0.67
2	0.48
3	0.67
4	0.57
5	0.43

Figure (4.18)

New minimum support is calculated using equation (3.2) and is it equal to 0.42. The construction of C2 is the same, the construction of L2

depends on the new minimum support 0.42, all items having ItemSupp greater than or equal to 0.42 remains all others are pruned, as shown in figure (4.19).

C2	L11	L12	ItemSupp
	1	2	0.33
	1	3	0.57
	1	4	0.43
	1	5	0.43
	2	3	0.38
	2	4	0.43
	2	5	0.24
	3	4	0.48
	3	5	0.33
	4	5	0.29

L2	Item1	Item2	ItemSupp
	1	3	0.57
	1	4	0.43
	1	5	0.43
	2	4	0.43
	3	4	0.48

Figure (4.19)

New minimum support is calculated using (equation 3.2) and is equal to 0.4. The construction of C3 is the same, the construction of L3 depends on the new minimum support 0.4, all items having ItemSupp greater than or equal to 0.4 remains all others are pruned, as shown in figure (4.20).

C3	L21	L22	L23	ItemSupp
	1	3	4	0.38
	1	3	5	0.33
	1	4	5	0.29

L3	Item1	Item2	Item3	ItemSupp
	1	3	4	0.38

Figure (4.20)

New minimum support is calculated using (equation 3.2) and is equal to 0.38. The algorithm stops here because  $C4 = \emptyset$ .

The number of association rules generated is (6 association rules), as shown in figure (4.21).

Total Ass. Rules Generated: 6			
Left HS		Right HS	Conf.
[1]	--->	[3 . 4]	0.57
[3]	--->	[1 . 4]	0.57
[4]	--->	[1 . 3]	0.67
[1 . 3]	--->	[4]	0.67
[1 . 4]	--->	[3]	0.88
[3 . 4]	--->	[1]	0.79

Figure (4.21)

New minimum confidence is calculated using (equation 3.3), it is found equal to 0.85 using (equation 3.4), from the 6 association rules generated, the obtained rules are only the ones that have confidence above the (New Min Conf), i.e. only the rules that have minimum confidence above 0.85 remains, all other rules are pruned so the resulted rules is only one rule, as shown in figure (4.22).

Ass. Rules with New Min Conf.:0.85			
No.	Left HS	Right HS	Conf.
1	[1 . 4]	---> [3]	0.88

Figure (4.22)

### 4.2.3 Implementing Apriori Algorithm (with fuzzy function 2)

The user chooses Fuzzy Apriori F2 and enters a user threshold number in the (User Thr.) box; User Thr. must have value between 2 and 6,

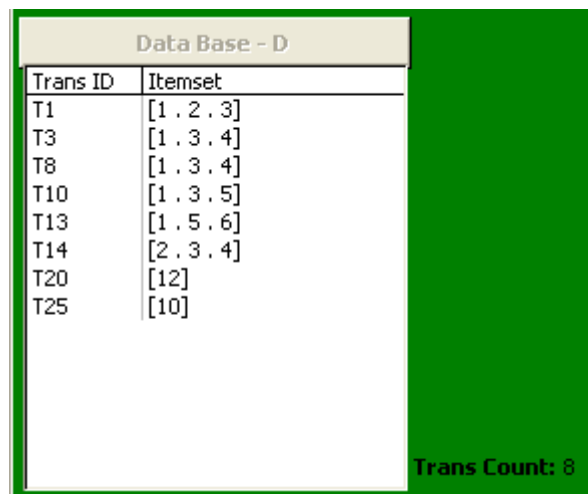
the user also enters  $\beta$  values to determine minimum support for each  $C_i$ ,  $i=\{1\dots6\}$  tables, as shown in figure (4.23).



Fuzzy Function Names	
Fuzzy Fun2	
User Thr. :	3
$\beta_1$	0.5
$\beta_2$	0.5
$\beta_3$	0.2

Figure (4.23)

First of all the Data Base – D Table of figure (4.4) is filtered depending on User Thr., (i.e. all transaction of items greater than 3-items are deleted), as shown in figure (4.24).



Data Base - D	
Trans ID	Itemset
T1	[1, 2, 3]
T3	[1, 3, 4]
T8	[1, 3, 4]
T10	[1, 3, 5]
T13	[1, 5, 6]
T14	[2, 3, 4]
T20	[12]
T25	[10]

Trans Count: 8

Figure (4.24)

The construction of  $C_i$  and  $L_i$  tables is the same as the two previously explained methods but  $L_i$  construction depends on  $\beta$ , new minimum support is calculated using steps 4 & 5 of algorithm (3-3).

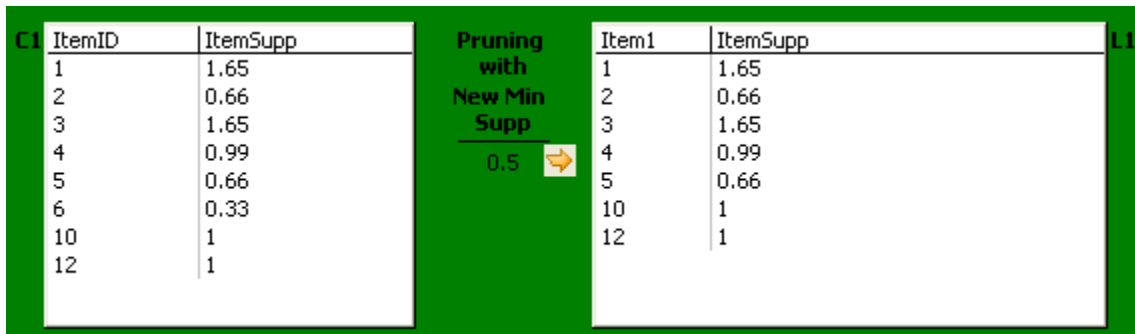


Figure (4.25)

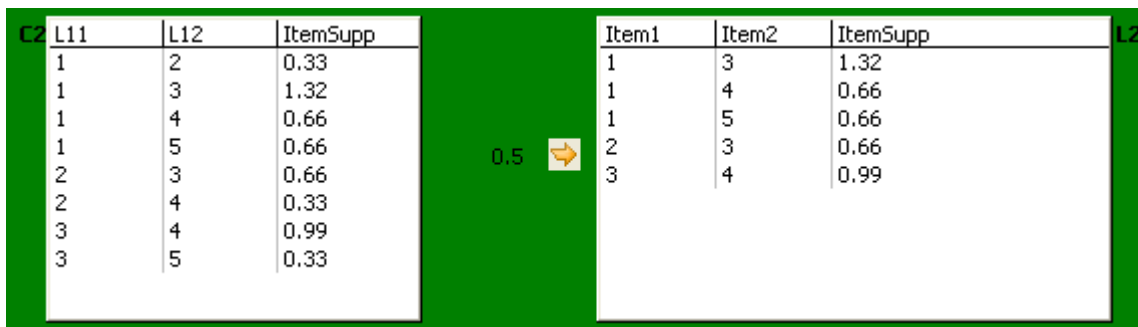


Figure (4.26)

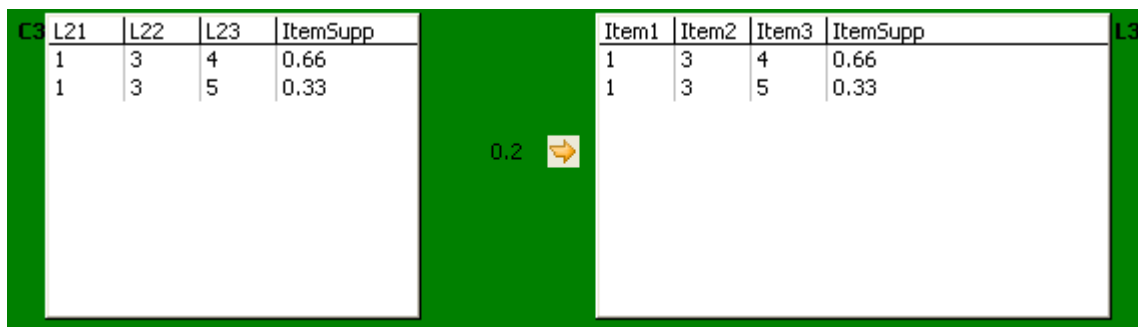


Figure (4.27)

The system stops here, no more tables are generated because  $\delta=3$ , as shown in Figure (4.27).

The total number of generated association rules is (12 association rule), as shown in figure (4.28).

Total Ass. Rules Generated: <u>12</u>		
Left HS	Right HS	Conf.
[1]	---> [3 . 4]	0.4
[3]	---> [1 . 4]	0.4
[4]	---> [1 . 3]	0.67
[1 . 3]	---> [4]	0.5
[1 . 4]	---> [3]	1
[3 . 4]	---> [1]	0.67
[1]	---> [3 . 5]	0.2
[3]	---> [1 . 5]	0.2
[5]	---> [1 . 3]	0.5
[1 . 3]	---> [5]	0.25
[1 . 5]	---> [3]	0.5
[3 . 5]	---> [1]	0.5

Figure (4.28)

All rules having minimum confidence greater than Min Conf (0.8) as entered in the first place by the user is filtered, as shown in figure (4.29).

Ass. Rules with New Min Conf.: <u>0.5</u>			
No.	Left HS	Right HS	Conf.
1	[4]	---> [1 . 3]	0.67
2	[1 . 3]	---> [4]	0.5
3	[1 . 4]	---> [3]	1
4	[3 . 4]	---> [1]	0.67
5	[5]	---> [1 . 3]	0.5
6	[1 . 5]	---> [3]	0.5
7	[3 . 5]	---> [1]	0.5

Figure (4.30)

### 4.3 Implementation Results

The results of three methods are:

1- Apriori results:

Table (4-1) Apriori results

Minimum support	Minimum confidence	Number of association rules
0.1	0.5	59
0.1	0.8	37
0.2	0.5	25
0.2	0.8	7
0.3	0.5	5
0.3	0.8	3
0.4	0.5	1

The implementation results show that whenever the support and confidence increase, the number of association rules decrease, and vice versa.

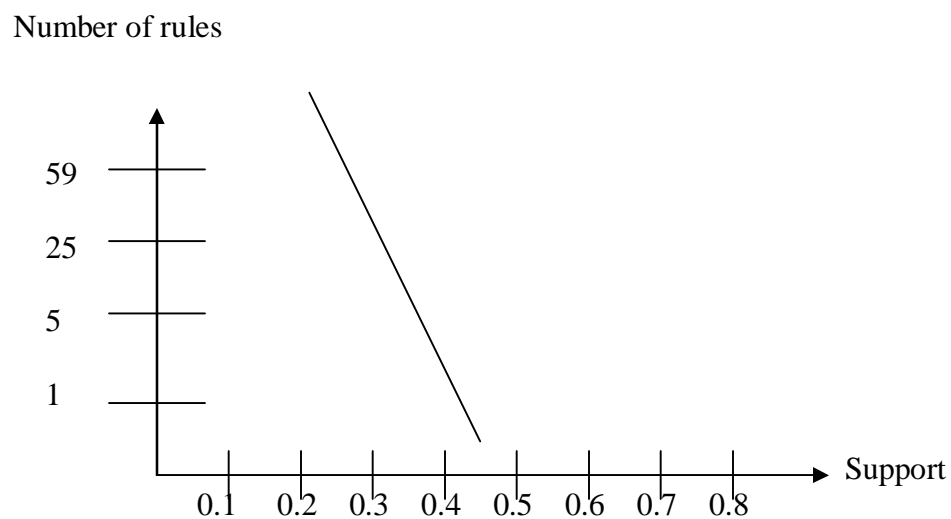


Figure (4.31) Apriori results

## 2- Apriori results with fuzzy function 1:

Table (4-2) Fuzzy Apriori results (Fuzzy Function 1)

Minimum support	User threshold	Minimum confidence	Number of association rules
0.1	0.1	0.5	22
0.1	0.1	0.8	10
0.1	0.01	0.5	17
0.2	0.01	0.5	25
0.2	0.03	0.5	14
0.3	0.01	0.5	5
0.3	0.1	0.5	1
0.4	0.1	0.5	1

The implementation results show that whenever the support, confidence and user threshold increase, the number of association rules decrease, and vice versa.

Number of rules

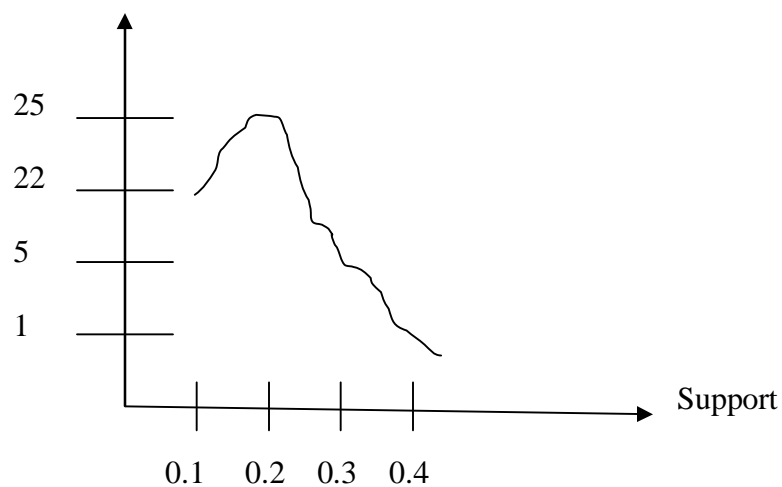


Figure (4.32) Fuzzy Apriori results (Fuzzy Function 1)



## 3- Apriori results with fuzzy function 2:

$\delta$  will be chosen equal to 3 for example, means that execution will reach to level three

Table (4-3) Fuzzy Apriori results (Fuzzy Function 2)

$\beta_1$	$\beta_2$	$\beta_3$	Minimum confidence	Number of association rules
0.1	0.1	0.1	0.5	15
0.1	0.1	0.1	0.8	6
0.1	0.2	0.2	0.5	15
0.3	0.3	0.3	0.5	15
0.4	0.3	0.3	0.5	10
0.4	0.3	0.3	0.5	3
0.4	0.5	0.6	0.5	4
0.5	0.5	0.5	0.5	4
0.7	0.7	0.7	0.5	2

The implementation results shows that whenever user threshold increase, the number of association rules increase, and vice versa.

Number of rules

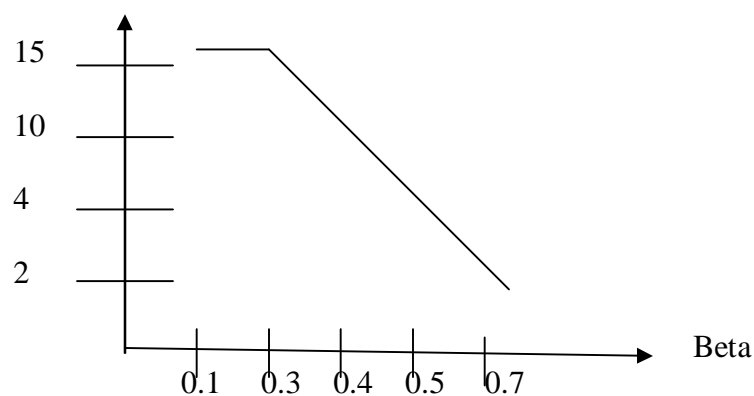


Figure (4.33) Fuzzy Apriori results (Fuzzy Function 2)

# *Chapter Five*

## *Conclusions and Future Work*

# Chapter Five

## Conclusions and Future Work

### 5.1 Conclusions

The conclusions that could be drawn from this work are listed as follows:

- 1- The generated number of association rules with fuzzy is smaller than the number of association rules with the Apriori algorithm only the difference between them is not standard because it depends on the dataset itself.
- 2- When fuzzy function 2 is used, the transaction database is filtered according to a specified threshold (this threshold gives the maximum number of items in the transaction); this will help in reducing space and execution time.
- 3- In Apriori algorithm, if Min Sup is small then number of generated association rules is large and vice versa.
- 4- In Fuzzy Function 1 if Min Sup and User Thr. is small then number of generated association rules is large and vice versa.
- 5- In Fuzzy Function 2 if User Thr. is large then number of generated association rules is large and vice versa.

## 5.2 Suggestions for future works

There are several ideas for developing the fuzzy apriori system such as:

- 1- The fuzzy apriori system could be implemented using hash tree for simplicity, the fuzzy apriori system is not implemented using hash tree here because there is a huge amount of data and using hash tree requiring returning to the RAM for each traversal of the tree, so the fuzzy apriori system is implemented in a way that takes subset from subset so for each iteration the amount of data in a subset decreases, this advantage does not need to return back to the RAM for each iteration and that increases the efficiency of space used i.e. reducing space and this will reduce execution time.
- 2- It is easy to implement the fuzzy apriori system to work on distributed environment and multi clients without big change in the program, i.e., the transaction database can be distributed on multi clients and the system can take information from different clients at the same time.

# *References*

# References

- § [Ale01] Alex Freitas, “A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery”, Postgraduate Program in Computer Science, Pontificia Universidade Catolica do Parana Rua Imaculada Conceicao, 1155. Curitiba - PR. 80215-901. Brazil, 2001.
- § [And05] Andreas L, “Agent Intellegence Through Data Mining”, Symeonidis Pericles A. Mitkas, Springer, 2005.
- § [Bin07] Bing Liu, “Web Data Mining Exploring Hyperlinks, Contents, and Usage Data”, Department of Computer Science, University of Illinois at Chicago, Springer-Verlag Berlin Heidelberg, 2007.
- § [Cha98] Charu C. Agrawal and Philip S. Ya, “Mining Large Itemsets for Association Rules”, Bulletin of the IEEE computer society Technical Committee on Data Engineering, 21(1) March 1998.
- § [Dav01] David Hand, et al., “Principles of Data Mining”, MIT Press 2001.

- § [Dha03] Dhananjay R. Thiruvady, BComp, “Mining Negative Rules in Large Databases using GRD”, M.sc thesis, School of Computer Science and Software Engineering at Monash University, November, Copyright by Dhananjay R. Thiruvady, 2003.
  
- § [Dan05] Daniel T. Larose, “Discovering Knowledge in Data An Introduction to Data Mining”, by John Wiley & Sons, 2005.
  
- § [Da05] Da Ruan Guoqing Chen Etienne E. Kerre Geert Wets (Eds.), “Intelligent Data Mining Techniques and Applications”, Springer, 2005.
  
- § [Dan06] Daniel T. Larose, “Data Mining Methods and Models”, by John Wiley & Sons, Inc, 2006.
  
- § [Fan06] Fan Jianhua & Li Deyi, “An Overview of Data Mining and Knowledge Discovery”, 2006.
  
- § [Ger02] German Florez, Susan M. Bridges, and Rayford B. Vaughn, “An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection”, National Science Foundation Grant# CCR-9988524 and the Army Research Laboratory Grant# DAAD17-01-C-0011, 2000.

- § [Jia00] Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, 2000.
  
- § [Jef06] Jeffery W. Seifert, ‘Data Mining and Homeland Security: An overview’, CRS Report for congress, January 2006, [www.fas.org/sgp/crs/intel/R131798.pdf](http://www.fas.org/sgp/crs/intel/R131798.pdf).
  
- § [Jia06] Jiawei Han Micheline Kamber, “Data Mining: Concepts and Techniques”, Elsevier, 2006.
  
- § [Lip05] Lipo Wang, Xiuju Fu, “Data Mining with Computational Intelligence”, Springer 2005.
  
- § [Mig03] Miguel Delgado, et al., “Fuzzy Association Rules: General Model and Applications”, IEEE Transactions on Fuzzy Systems, VOL. 11, NO. 2, April 2003.
  
- § [Moh08] Mohammad Amin Rigi, et al. , “An Evolutionary Data Mining Model for Fuzzy Concept Extraction”, Department of Computer Eng.Ferdowsi University, Mashhad, Iran, IEEE, 2008.
  
- § [Ode08] Oded Maimon, Lior Rokach, “Soft Computing for Knowledge Discovery and Data Mining”, Springer, 2008.
  
- § [Pet98] Peter Cabena, et al., “Discovering Data from Concept to implementation”, Prentice-Hall Inc., 1998.



- § [Pie98] Pieter Adriaans, Dolf Zantinge, “Data Mining”, Addison Wesley, 1998.
- § [Rak96] Rakesh Agrawal, Heikki Mannila, R. Srikant, Hannu Toivonen and A. Inkeri Verkamo, “Fast Discovery of Association Rules”, Santiago de Chile, 1996.
- § [Rah98] Rahul Ramachandran, “Application of Fuzzy Logic in Data Mining”, Computer Science Department, University of Alabama in Huntsville, 1998.  
[http://www.cs.uah.edu/~thinke/CS687/Fall97/Tech/rahul\\_dbase\\_paper.html](http://www.cs.uah.edu/~thinke/CS687/Fall97/Tech/rahul_dbase_paper.html)
- § [Rue02] Ruey-Shun Chen, et al., “Mining fuzzy association rules for classification problems”, Institute of Information Management, National Chiao Tung University, Elsevier Science Ltd, 2002.
- § [Ric05] Richard Jensen, “Combining rough and fuzzy sets for feature selection”, Ph.D thesis, University of Edinburgh, 2005.
- § [Rol06] Rolly Intan, “An Algorithm for Generating Single Dimensional Fuzzy Association Rule Mining”, Faculty of Technology Industry, Informatics Engineering Department, Petra Christian University, Journal Informatics VOL. 7, NO. 1, MEI 2006.

- § [Raw07] Rawia Tahrir Salih Kadoori, “Extracting Association Rules From Distributed Association Rules”, M.sc thesis, University of Technology, Department of Computer Science, October 2007.
  
- § [Ser97] Sergy Brin, Rajeev Motwani, Jeffery D. Ullman and Sergy Tsur. “Dynamic Itemset Counting and Implication Rules for Market Basket Data”. In processing of data (SGMOD97) Tucson, Arizona USA May 1997.
  
- § [Sar05] Sara Morgan Rea, “Building Intelligent .NET Applications: Agents, Data Mining, Rule-Based Systems, and Speech Processing” Addison Wesley Professional, 2005.
  
- § [Sul08] M. Sulaiman Khan, “A Framework for Mining Fuzzy Association Rules from Composite items”, University of Liverpool, UK, 2008.
  
- § [Tzu99] Tzung-Pei Heng, et al., “A Fuzzy Data Mining Algorithm for Quantitative Values”, I-Shou University, Department of Information Management, Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, IEEE, 1999.
  
- § [Two99] Two Crows Corporation, “Introduction to Data Mining and Knowledge Discovery”, Third Edition, Two Crows Corporation, 1999.

- § [Tay04] Taylor & Francis Group, “Pattern Recognition Algorithms for Data Mining: Scalability Knowledge Discovery and Soft Granular Computing”, Taylor & Francis Group, LLC, 2004.
  
- § [Wei04] Weiqing Jin, “Fuzzy Classification Based on Fuzzy Association Rule Mining”, Ph.D. Thesis, North Carolina State University, 2004.
  
- § [Yik02] Yike Guo, Robert Grossman, “High performance data mining Scaling Algorithms, Applications and Systems”, Kluwer Academic Publishers, 2002.
  
- § [Zhe01] Zhengxn Chen, “Data Mining and Uncertain Reasoning”, John Wiley & Sons Inc. 2001.
  
- § [Zim02] Zimpi Helen Komo, “Thesis Proposal”, Department of Computer Science, University of Manitoba, July 29, 2002.
  
- § [Zen05] Zengchang Qin, “Learning with Fuzzy Labels A Random Set Approach Towards Intelligent Data Mining Systems”, Ph.D thesis, Department of Engineering Mathematics, University of Bristol, October, 2005.
  
- § [Zho07] Zhongfu Zhang, “Utilize Fuzzy Data Mining to Find the Living Pattern of Customers in Hotels”, Lanzhou Jiaotong University, IEEE, 2007.

# الخلاصة

نمو مخازن بيانات ضخمة قد ادى إلى وضع عدد من المعالجات التي تعمل آليا لاكتشاف العلاقات بين تلك البيانات في المخازن. هذه المعالجات وغالبا ما يشار الى جانب عدد من الأسماء بما في ذلك التنقيب عن البيانات ،اكتشاف المعلومات ، تمييز الاشكال ، وآلية التعلم . التنقيب عن البيانات هي عملية استخراج معلومات ضمنية مفيدة، لم تكن معروفة سابقا. وهو يستخدم تلقائيا لاستخراج المعلومات من مجموعات كبيرة من البيانات .

تطبيق المنطق المشوش مع التنقيب عن البيانات يجعل المعلومات مفهومه من قبل البشر. التنقيب عن البيانات يمكن أن يكون لها العديد من الأساليب مثل القواعد الترابطية, التصنيف, التجميع. واحدة من وسائل تطبيق القواعد الترابطية هو الخوارزمية التكهنية. في هذه الاطروحة، سوف يبنى نظام تكهني مشوش؛ اولا سوف يستخدم خوارزمية تكهنية فقط ، ثم خوارزمية تكهنية مع تطبيق المنطق المشوش لإيجاد القواعد الترابطية. وسوف يتم ايجاد العلاقات بين البضائع المخزونة في سوبر ماركت لتقديم معلومات عن اكثر البضائع المباعه. من نتائج التجارب ، تبين أن وظائف المنطق المشوش تؤدي الى استخراج قواعد أقل من عدد من القواعد المستخرجة من خلال تطبيق خوارزمية تكهنية فقط.



جمهورية العراق  
وزارة التعليم العالي و البحث العلمي  
جامعة النهرين  
كلية العلوم

# التنقيب عن البيانات باستخدام العلاقات الترابطية مع المنطق المشوش

رسالة مقدمة الى كلية العلوم, جامعة النهرين و هي جزء من متطلبات نيل  
شهادة الماجستير في علوم الحاسوب

من قبل  
**اماني محمد عبود**

بكالوريوس  
2005

اشراف  
**د. سوسن كمال ثامر**

1429

2008