# INTRODUCTION

## Abstract

In the data mining field, association rules have been researched for more than fifteen years; however, the degree to which the support threshold effectively discovers interesting association rules has received little attention. A new framework for data mining through which interesting association rules called *coherent rules* can be discovered is studied and implemented. Coherent rules are those associations that can be mapped to logical equivalences according to propositional logic. Hence, coherent rules can be reasoned as logically true statements based solely on the truth table values of logical equivalence. Discovering coherent rules resolves the many difficulties in mining associations that require a preset minimum support threshold. Apart from solving the issues of a support threshold, the coherent rules found can also be reasoned as logical implications due to the mapping to the truth table values of logical equivalence. In contrast, classic association rules cannot be reasoned as logical implications due to their lack of this logic property. An algorithm to discover coherent rules is also presented. The algorithm was designed to find the shortest and strongest rule or most effectual coherent rules by exploiting the properties of coherent rules. Decision or actions can be implemented based on these coherent rules. In a situation whereby users are interested in weaker and/or longer rules, the algorithm enables parameters to be set. Unlike support threshold settings, these parameters do not require users to have prior knowledge of the context in which the data mining takes place. In Association Rule Mining, usually minimum support threshold is used to get the association rules. Minimum support threshold played a vital role in the entire process. To remove this dependency on minimum support threshold, we have proposed a framework through this project to discover interesting positive and negative rules for business by giving the domain knowledge called as coherent rules which are discovered based on the properties of propositional logic and thus do not require the minimum support threshold. We have explained the formation of coherent rules. Later, to reduce the complexity and to make it more efficient we have added the feature of domain – driven to the framework of coherent rules and this feature is demonstrated with the help of an implemented example. Further we have also introduced the concept of Combined Rule Mining which further enhances the results generated.

We have tested our framework on several datasets. The results confirm the strength of coherent rules in finding association rules that can be reasoned logically and in finding association rules that consider both infrequent items and negative associations. The algorithm used to discover coherent rules is also efficient. This was demonstrated by the number of prunings made to the search space during the discovery process.

## Introduction and Motivation

Associations are discovered based on logical implications. The principle of the approach considers that an association rule should only be reported when there is enough logical evidence in the data. To do this, we consider both presence and absence of items during the mining. An association such as beer 🡪 nappies will only be reported if we can also find that there are fewer occurrences of ¬beer🡪nappies and beer 🡪¬nappies but more of ¬beer🡪 ¬nappies. This approach will ensure that when a rule such as beer 🡪 nappies is reported, it indeed has the strongest statistical value in the data as comparison was made on both presence and absence of items during the mining process. In addition, the inverse case of customer not buying beer and customer not buying nappies should have statistics that support the rule being discovered due to the logic properties of equivalence. By considering this new approach in finding data pattern, a solution toward fulfilling domain-driven data mining requirements can be made. The proposed algorithm suggests a solution in two areas[2]:

1. It eliminates the need to use different intelligence models and its combinations as suggested in to determine appropriate threshold for the mining algorithms. The proposed algorithm discovers the natural threshold based on observation of data set. The different intelligence models can be used in conjunction with the proposed algorithm. In determining the target item(s) to be considered during the mining process. Hence, assuming that there are different intelligence models and a way of synthesizing it, the proposed algorithm can incorporate it to determine the target item(s) as an expression of business problem that one wants to solve.
2. It provides a logical underpinning to the discovery process of patterns. Currently, the illustration of the mapping of constraints to the discovery process in this paper is based on support value. However, it may be replaced by another constraint. The challenge is in finding appropriate mapping of constraints expressed in different intelligence models into a proportional logic equivalence that is recognized by the proposed algorithm.

## Problem Statement

To develop a business analysis system, which will help in making decisions and in introducing promotional schemes. This system can be useful to users for analysing the current trends of the domain in which they are interested and also helps business analysts in launching appropriate schemes for appropriate category of people.

## Scope of the Project

To introduce our framework, the distinction between an association rule and the different modes of an implication as defined in propositional logic. The topic of implication from logic is raised because our proposed mining model is based on an association rule’s ability to be mapped to a mode of implication. If an association can be mapped to an implication, then there is reason to report this relation as an association rule. Otherwise, without apriori such as the minimum support threshold, many association rules would be found, and we would need to report all of them. An implication having a rule where the left-hand side is connected to the right-hand side correlates two item sets together. This implication exists because it is true according to logical grounds, follows a specific truth table value, and does not need to be judged to be true by a user. The rule is reported as an interesting association rule if its corresponding implication is true.

## Organization of Project Report

This report has been divided into six chapters.

The first of these contains the *Introduction*. This comprises of the abstract which briefly explains the motivation behind this project. It is followed by a detailed introduction of data mining and technical analysis, and includes the definitions of various terms and concepts used as a part of the project. Next, the problem statement formally defines the aim of the project along with certain details of the design. The scope of the project precisely states the extent of the project and its boundaries. Finally, the organization of the report sheds light on the order in which the report has been compiled.

The second chapter includes the *Review of Literature.* It starts by explaining the domain of the project and explains certain concepts such as data mining, Market Basket Analysis and technical analysis. It then describes the state of the current methodologies and technologies used in this domain. Then, the different methodologies and technologies used in this project are explained and a brief overview of the project is given.

The third chapter is *Analysis and Design.* This illustrates the working and design of the project using various diagrams. Firstly, the functional and non-functional requirements of the system are listed. The use-case diagram describes the functionality provided by the system. The E-R diagram along with the database schema provides a view of the un­­derlying structure of the database system. The data flow diagram depicts the exact flow of data across the system through the various processes. Finally, the software architecture shows the various components and connectors in the system, along with the interactions between them.

This is followed by *Implementation.* This elaborates on the various coding practices and concepts used in creating this application. The entire process of development has been explained including how each part of the system was designed and integrated with the rest of the system along with the problems faced. The outcome of the application is portrayed through various screenshots of the application. Finally, a comparison of the various algorithms in terms of efficiency as well as performance is presented.

The penultimate chapter *Testing* comprises of the various test cases used and their outcomes, which are shown in the form of screenshots. It also contains the performance analysis of the various algorithms. The results of the different algorithms tested in multiple scenarios are presented, followed by a comparison of their space and time complexities.

Finally, the last chapter, *Conclusion and Further Work,* analyses the system implemented with a critical eye and points out improvements and changes that could be carried out in later versions in order to better the system.

# REVIEW OF LITERATURE



## Domain Explanation

### Data Mining

Data mining, theextraction of hidden predictive information from large databases, is a powerful technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. To do this, these tools use computational techniques such as classification, clustering, correlation, pattern matching etc. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to important business questions.

### 

### Business Intelligence

Association rule learning is a popular and well researched method for finding interesting relations between variables in large databases. With increase in the amount of data being collected and stored in databases, there has been increase in the demand of discovering correlation among the data with the help of association rules. It aims to identify strong rules discovered in databases using different measures of interestingness. Association rules find their key applications in the process of making decisions about marketing activities such as planning promotional schemes. A popular example of association rule mining is *Market Basket Analysis* and it is illustrated with the help of a diagram given below:

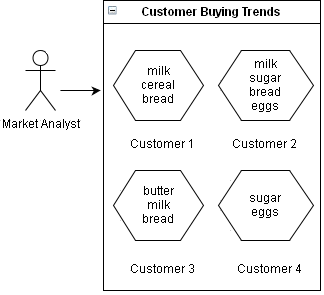


Figure 2.1: Market Basket Analysis

Analysis are made on customer buying habits by discovering associations between the different items that customers place in their shopping baskets as illustrated in Figure 2.1. Retailers find assistance in developing marketing strategies by identifying which items are frequently purchased together by customers with discovery of such associations. For instance, from the analysis done of the buying trends of the customers as shown in Figure 2.1, organizing promotional schemes for selling milk and bread together may further encourage the sale of these items together within single visits to the store. Traditional data mining process have problems such as less repeatability, no particular interest to business and lack of end user understandability. Thus, it lacks soft power in solving real-world complex problems. Domain-driven data mining is a new paradigm shift. It is aimed at making better business solution by providing tools for actionable knowledge which is passed on to the business committee for direct decision making and taking appropriate action. Thus in general terms we can describe it to be a shift from data centered hidden pattern mining to domain driven actionable knowledge discovery. Further, positive and negative association rules are quite important and necessary for decision making process and for analyzing and predicting business trends. It aims at maximizing possible benefits and minimizing the harmful impacts involved in applications like product placement and investment analysis. This technique describes the importance of negative association rules such as X ->Y in decision making because X ->Y can tell us that Y (which may be a harmful factor) rarely occurs when X (which may be an beneﬁcial factor) occurs[3]. There are many heterogeneous data sources like relational tables, files, systems and/or geographic locations which are used in everyday business applications. However the traditional data mining algorithms are not applied directly as patterns extracted from a single normalized table or subject file are less interesting or useful than a full set of multiple patterns extracted from different datasets. Association mining has a drawback of producing large collections of association rules that are difficult to understand and put into action. Thus a novel notion of combined patterns is proposed to extract useful and actionable knowledge from a large amount of learned rules. It gives an account of two kinds of redundancy in combined patterns: (1) the redundancy of combined rules within a rule cluster, and (2) the redundancy of combined rule pairs. Pruning methodology is a way of extracting the interesting rules from all the generated rules. Pruning selects only those rules which are having both the properties of static dataset *d* as well as transactional dataset *t* and lift >=1. In this technique [5] many of the rules get omitted as they are not showing properties of both dataset *d* and *t*. Significant association rules involving items in the database are generated with the help of efficient algorithms like a priori. The algorithm incorporates buffer management and novel estimation and pruning techniques.

#### Technical Analysis

Coherent Rules can be found in classification datasets also. Instead of discovering the associations between items, we discover the associations between attribute values, where the attribute value on a rule’s right-hand side is typically the class value. In this case, each attribute within a dataset holds a binary value of being observed or not observed. Consequently, we can then also discover the associations between attribute values and their class values. Suppose ℐ= 𝑖1,2,…,𝑖𝑛 , a set of attribute values. Let 𝐷 be a table of instances such that 𝐷= 𝑑1,2,…,𝑑𝑚 . An instance, 𝑑𝑗, holds a subset of attribute values such that 𝑑𝑗⊆ℐ. Suppose 𝒜 holds attribute values such that 𝒜={𝑎1,…,𝑎𝑢} and 𝒞 holds class values such that 𝒞={𝑐1,…,𝑐𝑣}, where 𝒜⊂ℐ,𝒞⊂ℐ and 𝒜∩𝒞=∅. Let 𝒫 be a power-set function. Let the projection Π𝜃 𝐷 on a table (𝐷) yield records that contain attribute values 𝜃. Let 𝐼𝑋 be the union of all power-sets on attribute values contained in 𝒜 such that 𝐼𝒜= Π𝜃 𝐷 𝜃∈𝒫 . Similarly, let 𝐼𝑌 be the union of all power-sets on 𝐼𝒞= Π𝜃 𝐷 𝜃∈𝒫 . We are interested in coherent rules between two sets of attribute values 𝑋 and 𝑌, where 𝑋∈𝐼𝒜, 𝑌∈𝐼𝒞, 𝑋≠∅, and 𝑌≠∅. We create a very small artificial classification dataset derived from the Zoo dataset to illustrate the type of coherent rules that can be discovered. The dataset is created by writing the common attributes for five types of animal – lion, gorilla, bear, dolphin and bass *fish* – in Table 2.1 given below. We use the attribute value 1 to denote observation of an attribute; a non-observation is denoted using 0. The attribute *leg* contains multiple attribute values, indicating the number of legs.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Animal Type | Hair | Tail | Fins | Aquatic | Milk | Leg | Animal Class |
| Lion | 1 | 1 | 0 | 0 | 1 | 4 | *mammal* |
| Gorilla | 1 | 0 | 0 | 0 | 1 | 2 | *mammal* |
| Bear | 1 | 0 | 0 | 0 | 1 | 4 | *mammal* |
| Dolphin | 0 | 1 | 1 | 1 | 1 | 0 | *mammal* |
| Bass *fish* | 0 | 1 | 1 | 1 | 0 | 0 | *fish* |

Table 2.1: Small Classification Dataset

Based on Table 2.1, we consider six attributes – hair, tail, fins, aquatic, milk, and leg – to describe an animal class. Among these six attributes, there are eight unique attribute values to be considered, as shown in 𝒜:

|  |  |
| --- | --- |
| 𝒜= 𝑕𝑎𝑖𝑟 {1} ,𝑡𝑎𝑖𝑙 {1} ,𝑓𝑖𝑛𝑠 {1} ,𝑎𝑞𝑢𝑎𝑡𝑖𝑐 {1} ,𝑚𝑖𝑙𝑘 {1} ,𝑙𝑒𝑔 {0} ,𝑙𝑒𝑔 {2} ,𝑙𝑒𝑔 {4} |  |

where any attribute value having a numeric value greater than ‘0’ indicates that the attribute value can be observed on an animal. Otherwise, it is not observed. Attribute value that has a numeric value greater than ‘1’ indicates that this attribute value is observed in a certain quantity value. For example, 𝑙𝑒(2) indicates that a pair of legs can be observed on an animal.

Based on Table 2.1, there are two unique class values as shown in 𝒞:

|  |  |
| --- | --- |
| 𝒞= { 𝑚𝑎𝑚𝑚𝑎𝑙 (1) , 𝑓𝑖𝑠h (1) } |  |

The power sets on the attribute values 𝒜 are given as:

|  |  |
| --- | --- |
| 𝐼𝒜= {{𝑛𝑢𝑙𝑙} , {h𝑎𝑖𝑟 (1)} , {𝑡𝑎𝑖𝑙 (1)} , {𝑓𝑖𝑛𝑠 (1)} ,…, {𝑙𝑒𝑔 (4)} , 𝑕{h𝑎𝑖𝑟 (1) ,𝑡𝑎𝑖𝑙 (1)} ,  {h𝑎𝑖𝑟 (1) ,𝑓𝑖𝑛𝑠 (1)} ,…, } |  |

and the power set on class values 𝒞:

|  |  |
| --- | --- |
| 𝐼𝒞= {𝑛𝑢𝑙𝑙 , 𝑚𝑎𝑚𝑚𝑎𝑙(1) , 𝑓𝑖𝑠h(1) } |  |

We are interested in coherent rules between two sets of attribute values 𝑋 and 𝑌, where 𝑋∈𝐼𝒜, 𝑌∈𝐼𝒞, 𝑋≠∅, and 𝑌≠∅. We show a contingency Table 2.2 between the attribute values 𝑚𝑖𝑙(1) and 𝑚𝑖𝑙𝑘(0) , and class values 𝑚𝑎𝑚𝑚𝑎𝑙(1) and 𝑚𝑎𝑚𝑚𝑎𝑙(0). Where 𝑚𝑖𝑙𝑘 (0) and 𝑚𝑎𝑚𝑚𝑎𝑙 ( 0) denotes absence of 𝑚𝑖𝑙𝑘(1) and 𝑚𝑎𝑚𝑚𝑎𝑙(1) respectively, we write each single attribute value without the set (for example, 𝑚𝑖𝑙𝑘(1) instead of {𝑚𝑖𝑙𝑘(1)}).

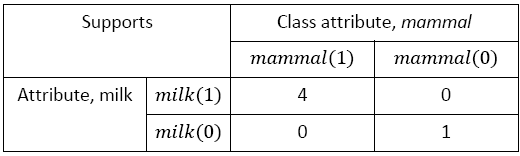


Table 2.2 A Contingency Table for Milk and Mammal

Based on the theory of coherent rules and Table 2.2, equation below shows a coherent rule that consists of two pseudo-implications of equivalences:

|  |  |
| --- | --- |
| 𝑚𝑖𝑙𝑘(1)⇒𝑚𝑎𝑚𝑚𝑎𝑙(1), 𝑚𝑖𝑙𝑘(0)⇒𝑚𝑎𝑚𝑚𝑎𝑙(0) |  |

We can conveniently identify coherent rules from a contingency table such as in Table 2.2, where the frequency of co-occurrences in quadrants 𝑄1 and 𝑄4 has support values greater than quadrants 𝑄2 and 𝑄4. We highlight that the knowledge found using the concept of coherent rules is independent of any user inputs. It was discovered based on mapping to logical equivalences. If decoupled, there are two association rules with the extra property that can be mapped to logical equivalences (that is, the mammalian property of lactation). This is supported by general knowledge. The discovery of coherent rules is useful in application domains where the domain knowledge is not known to a user or is difficult to grasp. In the retail domain, the reasons for associations between items are not obvious. Customers have various reasons to buy different items together. Using mapping to logical equivalences, we can discover coherent rules and its association rules without the need to survey on customers. As a result, we know some items are associated together based on logical grounds.

## Existing Technology

The use of association rule mining technique is to describe the associations among items in a database. These associations represent the domain knowledge encapsulated in databases. Identifying domain knowledge is important because these knowledge rules usually are known only by the domain experts over years of experience. Thus, association rule mining is useful to identify domain knowledge hidden in large volume of data efficiently. The discovery of association rules is typically based on the support and confidence framework where a minimum support (min sup) must be supplied to start the discovery process. A priori is a representational algorithm based on this framework and many other algorithms are a priori-like. Without this threshold specified, typically, no association rules can be discovered because the procedure to discover the rules will quickly exhaust the available resources. Nonetheless, having to constrain the discovery of association rules with a preset threshold, in turn, requires in-depth domain knowledge before the discovery of rules can be automated. The use of min sup generally assumes that: a domain expert can provide the threshold value accurately. The knowledge of interest must have occurred frequently at least equal to the threshold a single threshold is enough to identify the knowledge sought by an analyst. In practice, there are cases where these assumptions are not appropriate and rules reported lead to erroneous actions. So we propose a novel framework to address the above issues by removing the need for a minimum support threshold.

## Problems with Association Rules

The process of discovering association rules is based on two main factors namely support and confidence i.e. minimum support must be supplied to start the process. Without the threshold being specified, no association rules can be discovered. This is because without the specification of the factors mentioned above, the number of rules generated will be too high and may involve low interestingness measure data which may not solve the purpose for which they are being discovered. Instead, it takes more time and resources in computing uninterested data. Further, setting up the minimum support threshold requires in-depth domain knowledge before the discovery of rules. In this manner, assumption is being made on the usage of minimum support on following factors:

* + - 1. **Accuracy of the specified threshold value**

It basically states that a domain expert always specifies accurate threshold value. This emphasizes the need of domain expert without whom the process will land up in standstill position. Further, it also assumes that the data above the specified threshold value is only of high interestingness measure. It blindly accepts the accuracy of the specified threshold value. Consequently, this leads to loss of association rules which may have contributed in decision making process

* + - 1. **A Single Threshold is Enough for Identifying the Knowledge**

The technique of finding association rules aims to work globally on any kind of data. There can be cases where data is quite uncertain and distributed. So there can occur a situation where a single threshold is not enough for extracting the knowledge from data. This depicts the need of defining multiple thresholds for the particular dataset. In this manner, the assumptions made above are inappropriate and thus the rules reported lead to inaccurate and inconsistent results. In this paper, we propose a framework considering the above issues by removing the need of minimum support threshold .

## PROPOSED SYSTEM

## Architecture of System

The proposed system shown in Figure 2.2 works on both transactional as well as static dataset. After applying the domain knowledge, a combined dataset gets generated which has both the properties of transactional and static or demographic data. Through domain knowledge, only interesting features get selected from generated combined data and so the data gets reduced. Feature selection method helps to select only the interesting features which further reduce the data. On the generated dataset, after applying coherent algorithm, both positive and negative rules get generated. Proposed system suggests the pruning technique which helps to filter the uninterested rules. So the finally generated rules help to take the business decisions.

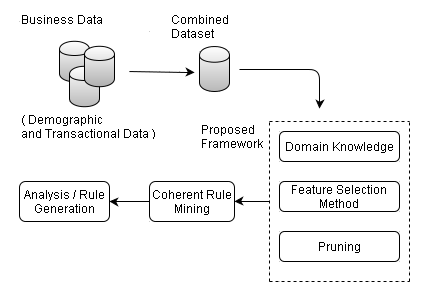


Figure 2.2: Architecture of proposed system

## Measurement of Association Rules Using other Methods of Interestingness

The coherent rules are generated based on logical implications like propositional logic, Implication and Equivalence. Propositional Logic is concerned with propositions and their interrelationships. A proposition is basically a condition of the world about which we want to state something. It is not necessary that the condition must be true always for us to comment on it. We might want to state that it is false or that it is true if some other proposition is true. The syntactic rules for the language of Propositional Logic are given below. In addition, the semantic interpretation for the expressions specified by these rules is elaborated .

Given these semantics, all of the logical conclusions can be drawn from any set of propositional sentences.

* A ***negation*** consists of the negation operator ¬ and a simple or compound sentence, called the *target*. For example, given the sentence *p* as *Customer buys bread*, we can form the negation of *p* as ¬*p i.e. Customer does not buy bread*.
* A ***conjunction*** is a sequence of sentences separated by occurrences of the ∧ operator and enclosed in parentheses, as shown below. The constituent sentences are called *conjuncts*. For example, we can form the conjunction of *p* which is *Customer buys bread* and *q* which is *Customer buys butter* as (*p* ∧ *q*) *i.e.* *Customer buys bread and butter*.
* A ***disjunction*** is a sequence of sentences separated by occurrences of the ∨ operator and enclosed in parentheses. The constituent sentences are called *disjuncts*. For example, we can form the disjunction of *p* which is *Customer buys bread* and *q* which is *Customer buys butter* as (*p* ∨ *q*) *i.e.* *Customer buys bread or butter*.
* An ***implication*** consists of a pair of sentences separated by the ⇒ operator and enclosed in parentheses. The sentence to the left of the operator is called the *antecedent*, and the sentence to the right is called the *consequent*. The implication of *p* which is *Customer buys bread* and *q* which is *Customer buys butter* is (*p* ⇒ *q*) *i.e.* *If Customer buys bread then he buys butter*.
* A ***reduction*** is the reverse of an implication. It consists of a pair of sentences separated by the ⇐ operator and enclosed in parentheses. In this case, the sentence to the left of the operator is called the *consequent*, and the sentence to the right is called the *antecedent*. The reduction of *p* which is *Customer buys bread* to *q* which is *Customer buys butter* is (*p* ⇐ *q*) *i.e.* *If Customer buys butter then he buys bread*.
* An ***Equivalence*** is a combination of an implication and a reduction. For example, we can express the equivalence of *p* which is *Customer buys bread* and *q* which is *Customer buys butter* as (*p* ⇔ *q*) *i.e.* *Customer buys bread if and only if he buys butter and vice-versa*.

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **p ⊃ q** |
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

Table 2.3: Truth Table for Equivalence

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **p ≡ q** |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

Table 2.4: Truth Table for Material Implication

**Logic:**

We highlight here that an implication is formed using two propositions *p* and *q*. These propositions can be either true or false for the implication’s interpretation. From these propositions, we have four implications

1. p → q,

2. p → ¬q,

3. ¬p → q, and

4. ¬p → ¬q.

Each is formed using standard symbols → and ¬ as explained previously. For example, *if butter is observed in a customer market basket*, then *bread is observed in a customer market basket*. For this statement, rule formed will be p→q. On the other hand, for the statement *if butter is not observed in a customer market basket*, then *bread is not observed in a customer market basket*, rule formed will be ¬p→¬ q. This highlights the concept of Negative Association Rules which gives certainty of the result generated.

## Generation of Coherent Rules

For generation of the coherent rules, following steps are involved.

**Algorithm:**

1**.** Generation of Combined data

2. Selection of dataset through Domain knowledge

3. Support Matrix Generation

4. Coherent Rules Generation

5. Pruning for rule reduction

* *Generation of Combined Data*

In combined rule mining Static and dynamic datasets are involved. For example, assume there are static attributes like Gender {Male, Female} *i.e.* for a particular customer these values do not change throughout the dataset. On the other hand, attributes like a particular product which conveys whether a user will buy it or not is dynamic. This value is dynamic since the customer who currently is not buying it might buy it in future. So for marketing purpose we need to analyze the association of the static attributes with dynamic attributes to predict the possibility of a particular product being marketed for a particular set of people. For example if a rule states that {Bluetooth\_Headphone: True, Car\_Video: True → MODERATE}. This rule depicts that if an individual belongs to a MODERATE class then he will buy Bluetooth\_Headphone and Car\_Video. So if in case, a promotional campaign needs to be launched for people buying these products then there will be large set of people to be considered. On the contrary, taking combined rule mining example clarifies this situation as follows:{Gender: Male, Marital\_status: Single, Bluetooth\_Headphone: True, Car\_Video: True → Moderate}. Here, this rule overcomes the drawback of above rule i.e. inspite of considering the entire set of people belonging to moderate class, it focuses on males who are not married and are of moderate class. This reduces computational cost drastically.

* *Selection of dataset through Domain knowledge*

While selecting the dataset for mining, if the domain knowledge is given as input to the system, then selected dataset gets reduced. The steps for selection of the domain knowledge are given below:

1. List the transactional and demographic tables at run time.
2. List the characteristics/features of selected table
3. List the distinct values of each selected characteristics
4. Select only the interested values of listed features
5. Select the class attributes.

Domain knowledge method reduces the selection of data by feature selection method to generate the combined data. So the time complexity and space complexity for the computations get reduced.

* *Algorithm to generate Support Matrix*

To get the support matrix, algorithm is given below:

**Input:**

Class [ ]: Unique class names

N: total number of classes in class [ ] set

Items [ ]: Power set of unique selected items

M: total number of items in Items [ ]

db: dataset with k records

**Algorithm:**

domain\_driven\_coherent (Items [], class [])

Begin

For each record in Class [] // class labels

{

For each record in Items [] // power set

{

For each record in db // records

{

If (db[t].contains (class[j]))

{

If (db[t].contains (Item[i]))

ClassPresent\_ItemPresent ++;

Else

ClassPresent\_ItemAbsent ++;

}

Else

{

If (db[t].contains (Item[i]))

ClassAbsent\_ItemPresent ++;

Else

ClassAbsent\_ItemAbsent ++;

}

}

}

End;

Output: Support Count Matrix

Algorithm : Generate Support Matrix

* *Coherent Rule Generation*

Suppose *p* and *q* are objects in the dataset and S (p) and

S (q) are supports of *p* and *q* respectively. After we get the support matrix, coherent rules are generated if the following conditions are satisfied:

* + - * S(p, q) > S(¬p, q) and S(p, q) > S(p, ¬q)
      * S(¬p, ¬q) > S(¬p, q) and S(¬p, ¬q) > S(p, ¬q)

E.g. If the support matrix formed by applying the algorithm mentioned in the section 3.2.3 is as follows:

|  |  |  |
| --- | --- | --- |
|  | S(p) | S(¬p) |
| S(q) | 10 | 9 |
| S(¬q) | 5 | 15 |

Table 2.5: Support value matrix for p and q

Now, the conditions necessary for forming coherent rules are checked as follows:

(S (p, q) = 10) > (S (¬p, q) = 9)

(S (p, q) = 10) > (S (p, ¬q) = 5)

(S (¬p, ¬q) = 15) > (S (¬p, q) = 9)

(S (¬p, ¬q) = 15) > (S (p, ¬q) = 5)

In this example, the above condition is satisfied and hence coherent rule (p🡪q) is formed and this also implies (¬p🡪¬q). This is called as negative association rule mining [6].

* *Pruning*

Pruning selects only those rules which are having both the properties of static dataset *d* as well as transactional dataset *t* and lift >=1. In this step many of the rules get omitted as they are not showing properties of both dataset *d* and *t*. With reference to Table 2.4 given below, rules 1, 2 and 3 are not useful to generate combined patterns. Only the rules given in Table 7 are considered for the generation of combined coherent rules. For each generated rule, interesting measures like Support, Confidence, Lift and Irule are calculated as given as follows:

Assume that the rule is a ^ b → T

1. *Support:*

The support of the rule, that is, the relative frequency of transactions that contain a ^ b and T.

support (a ^ b → T) = support (a+b+T)

1. *Confidence:*

The confidence of the rule.

confidence (a ^ b → T) = support (a+b+T) / support (a+b)

1. *Lift:*

The lift value of the rule is the additional interestingness measures on the rules. These measures can then be used to either rank the rules by importance (present a sorted list to the user) or as an additional pruning criterion.

Lift (a ^ b → T) = confidence (a ^ b → T) / support (T)

Two new lifts for measuring the interestingness of combined association rules are as:

Lifta (a ^ b → T) =Lift (a ^ b → T) /lift (b → T)

Liftb (a ^ b → T) =Lift (a ^ b → T)/Lift (a→ T)

where Lifta (a ^ b → T) is the lift of *a* with *b* as a precondition, which shows how much *a* contributes to the rule and Liftb (a ^ b → T) gives the contribution of *b* in the rule. Based on the above two new lifts, the interestingness of combined association rules is defined as Irule. Irule indicates whether the contribution of *a* (or *b*) to the occurrence of T increases with *a* (or *b*) as a precondition

1. *Irule:*

Irule (a ^ b → T) = Lifta (a ^ b → T)/ lift (a→ T)

Or

Irule (a ^ b → T) = Liftb (a ^ b → T)/ lift (b→ T)

The value of Irule falls in [0, +∞]. When Irule > 1, the higher Irule is, the more interesting the rule is. Once the Irule is calculated for each rule, arrange them in descending order, the rule with Irule > 1 and higher Irule value is more interesting. Before going for the pruning process, ensure that maximum frequent patterns are generated i.e. the ones having support value at least as much as the user-specific percentage of the database.

Pruning technique encompasses of following steps:

1. Selection of rules having both properties of both static and transactional dataset. Calculate the confidence, support, lift with the help of formulae given above. Select those rules which have confidence value greater than the desired value and lift>=1.
2. Suppose there is a pattern of rule A🡪T, then check if A ={X ^ Y} i.e. whether *A* is a combination of more than one item including both static and transactional feature.

## Methods and Technologies used in the Project

* **MICROSOFT SQL SERVER:**

Microsoft SQL Server is a comprehensive database server and information platform offering a complete set of enterprise-ready technologies and tools that help people derive the most value from information at the lowest total-cost-of-ownership. Enjoy high levels of performance, availability, and security; employ more productive management and development tools; and deliver pervasive insight with self-service business intelligence (BI).

Other Reasons:

1. Microsoft DBAs run more mission critical databases, when compared to Oracle DBAs.
2. SQL Server delivers six nines (99.9999%) uptime availability.
3. SQL Server is the most secure of any of the major database platforms.

* **Microsoft Visual Studio**

**Microsoft Visual Studio** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](http://en.wikipedia.org/wiki/Microsoft). It is used to develop [console](http://en.wikipedia.org/wiki/Console_application) and [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) [applications](http://en.wikipedia.org/wiki/Application_software) along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [Windows Mobile](http://en.wikipedia.org/wiki/Windows_Mobile), [Windows CE](http://en.wikipedia.org/wiki/Windows_CE), [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework), [.NET Compact Framework](http://en.wikipedia.org/wiki/.NET_Compact_Framework) and Microsoft Silverlight.

The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, [web designer](http://en.wikipedia.org/wiki/Web_designer), [class](http://en.wikipedia.org/wiki/Class_%28computing%29) designer, and database schema designer.

Visual Studio supports different [programming languages](http://en.wikipedia.org/wiki/Programming_language) by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](http://en.wikipedia.org/wiki/C_%28programming_language%29)/[C++](http://en.wikipedia.org/wiki/C%2B%2B) (via [Visual C++](http://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](http://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](http://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29) (via [Visual C#](http://en.wikipedia.org/wiki/Visual_C_Sharp)), and [F#](http://en.wikipedia.org/wiki/F_Sharp_%28programming_language%29) (as of Visual Studio 2010).

## Project Overview

Our framework for discovering coherent rules offers a technique for data mining that overcomes the limitations associated with existing methods and enables the finding of association rules among the presence and/or absence of a set of items without a preset minimum support threshold. The results justify continuing research in this area in order to increase the body of scientific knowledge of data mining – and specifically, association rules - and to provide practical support to those involved in data mining activities.

# ANALYSIS AND DESIGN

## Requirement Analysis

### Functional Requirements

1. The System should be able to work on any kind of data and generate coherent rules.
2. Rules generated should be more accurate than the traditional methods like apriori.
3. System should follow Domain Driven approach and generate rules considering it.
4. It should work equally efficiently with negative rules supporting negative association rule mining and thus generate more accurate results.
5. It should eliminate redundancy.
   * 1. **Non-functional Requirements**

Functional requirements are supported by [non-functional requirements](http://en.wikipedia.org/wiki/Non-Functional_Requirements) (Quality requirements), which impose constraints on the design or implementation.

Execution qualities, such as accessibility, dependability, accuracy and throughput are observable at run time. Evolution qualities, such as performance, maintainability, and scalability, legality are embodied in the static structure of the software system.

1. **Accessible**

The system should be easily accessible to the end user and should not produce any type of complexity during the use.

1. **Dependable**

Since the protection of entire user data completely depends on our system, a high dependability factor should be present that could develop a factor of trust between the user and the system.

1. **Accuracy**

The system should be as accurate as possible to always provide correct results to all the challenges put forward.

1. **Performance and Reliability**

The performance of the system highly depends upon both the accuracy and the average time. It should be reliable enough to provide high performance.

1. **Maintainable**

It should also not possess high maintainability issues wherein the whole system has to be rebuilt for a minimal amount of change.

1. **Scalable**

It should be built in such a way that scalability should not be a problem when new types of data are to be dealt with.

## Project Design

## Design Considerations

The design consists of the Use Case Diagram, Class Diagram, Database Schema, Data Flow Diagram and the Software Architecture of our System.

* + - 1. **Use Case Diagram**

A Use case Diagram focuses on the behaviour of the system from an external point of view. A use case describes a function provided by the system that yields a visible result for an actor.

* + - 1. **Entity Relationship Diagram**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

* + - 1. **Database Schema**

A database schema  of a database system is its structure described in a formal language supported by the database management system (DBMS) and refers to the organization of data to create a blueprint of how a database will be constructed (divided into database tables).

#### Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

#### Software Architecture

 In software architecture, a system is represented as a set of software components, their connections, and their important behavioural interactions. Creating software architecture promotes better understanding of the system, thus aiding the design process. Architecture Description Languages (ADLs) are used to describe a Software Architectures. Common elements of an ADL are component, connector and configuration.

### Design Details

#### Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a [use case](http://en.wikipedia.org/wiki/Use_Case). A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. For our application, the use case diagrams from system point of view and user point of view are as given below:

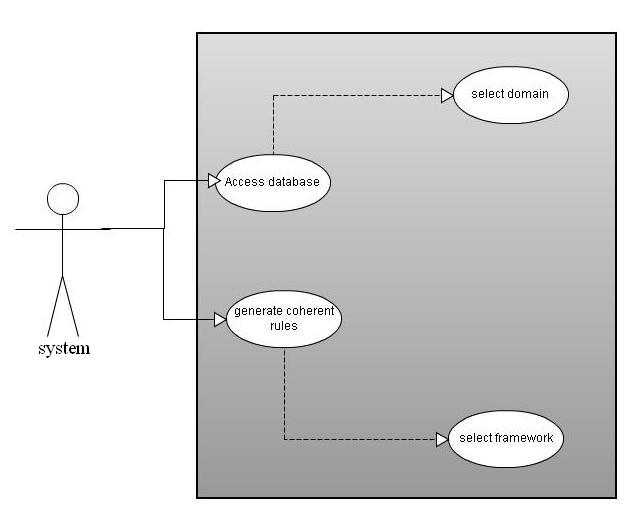


Figure 3.1: Use Case Diagram

Figure shows some of the system use cases according to the system. The System can access database then according to user selection it will generate domain driven then on that domain driven framework will be applied this leads to the generation of the coherent rules.

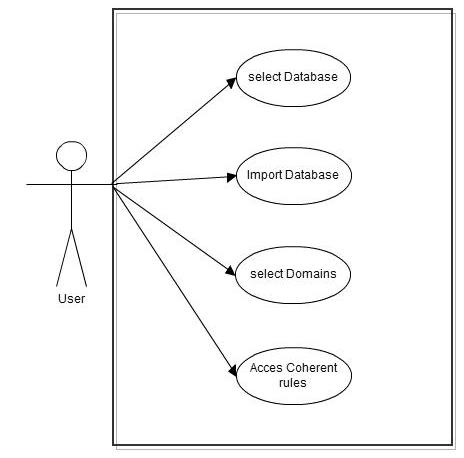


Figure 3.2 : Use case for the user.

The above diagram shows the use case diagram from user point of view wherein the user can select database, it will then import that database and select the domains in which he is interested and then finally can access the coherent rules.

* + - 1. **Class Diagram**

A class diagram in the [Unified Modeling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of static structure diagram that describes the structure of a system by showing the system's [classes](http://en.wikipedia.org/wiki/Class_%28computer_science%29), their attributes, operations (or methods), and the relationships among the classes.

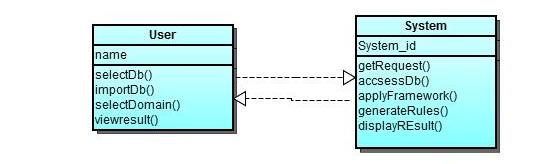


Figure 3.3:Class Diagram.

The above diagram shows the interaction between the two important classes user and system.

1. ACTIVITY DIAGRAM

Activity diagrams are graphical representations of [workflows](http://en.wikipedia.org/wiki/Workflow) of stepwise activities and actions with support for choice, iteration and concurrency. The following diagram describes the flow of the process of generation of coherent rules.

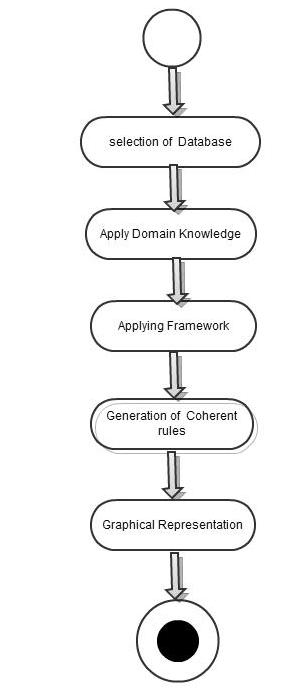


Figure 3.4:Activity Diagram

1. FLOWCHART DIAGRAM

A flowchart is a type of [diagram](http://en.wikipedia.org/wiki/Diagram) that represents an [algorithm](http://en.wikipedia.org/wiki/Algorithm) or [process](http://en.wikipedia.org/wiki/Process_%28science%29), showing the steps as boxes of various kinds, and their order by connecting them with arrows. This is depicted with the help of diagram as given below:

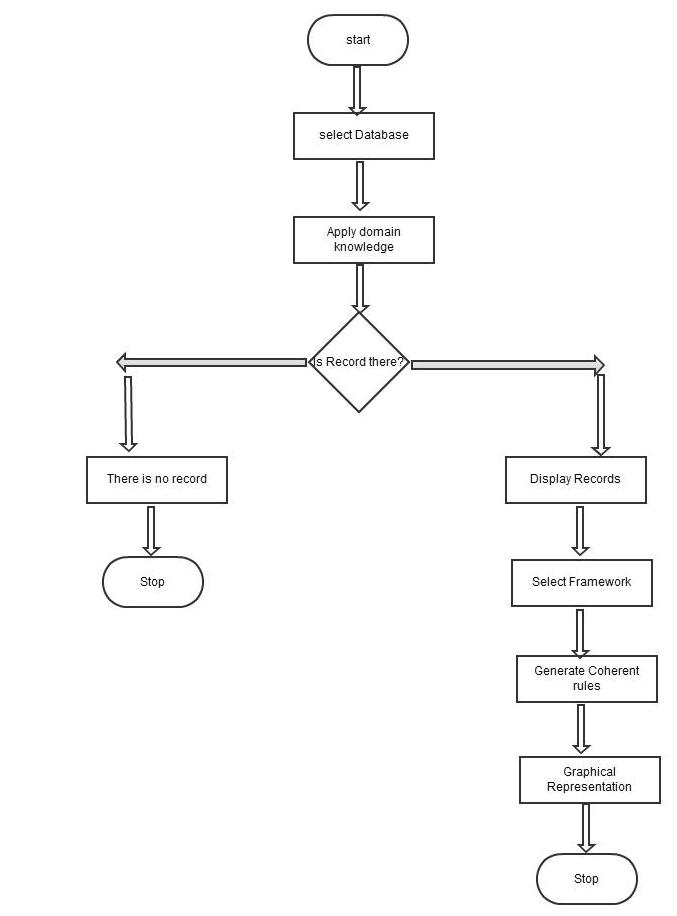


Figure 3.5: Flowchart Diagram.

1. DATABASE DIAGRAMS

A database schema of a [database system](http://en.wikipedia.org/wiki/Database_system) is its structure described in a [formal language](http://en.wikipedia.org/wiki/Formal_language) supported by the [database management system](http://en.wikipedia.org/wiki/Database_management_system) (DBMS) and refers to the organization of data to create a blueprint of how a database will be constructed (divided into database tables). The diagram is as shown below:

|  |
| --- |
| **Zoo Table** |
| Animal name |
| hair |
| feathers |
| eggs |
| milk |
| airborne |
| aquatic |
| predator |
| toothed |
| backbone |
| breathes |
| venomous |
| fins |
| legs |
| tail |
| domestic |
| catsize |
| type |

Figure 3.6: Zoo Table

|  |
| --- |
| **Retail Dataset** |
| Id |
| Marital\_Status |
| Gender |
| Bluetooth\_Headphone |
| Car\_Video |
| Home\_th\_System |
| Mp4\_Mp3 |
| Recording\_ph |
| Television |
| VCD\_DVD |
| Class\_Customer |

Figure 3.7 Retail Dataset Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Animal | hair | Feather | eggs | milk | Airborne | aquatic | predator | toothed | Back-bone | Breat-hes | Venomo-us | fins | legs | tail | domestic | catsize | type |
| Aardvark | T | F | F | T | F | F | T | T | T | T | F | F | 4 | F | F | T | mammal |
| Antelope | T | F | F | T | F | F | F | T | T | T | F | F | 4 | T | F | T | mammal |
| Bass | F | F | T | F | F | T | T | T | T | F | F | T | 0 | T | F | F | fish |
| Bear | T | F | F | T | F | F | T | T | T | T | F | F | 4 | F | F | T | mammal |
| Boar | T | F | F | T | F | F | T | T | T | T | F | F | 4 | T | F | T | mammal |
| buffalo | T | F | F | T | F | F | F | T | T | T | F | F | 4 | T | F | T | mammal |
| calf | T | F | F | T | F | F | F | T | T | T | F | F | 4 | T | T | T | mammal |
| carp | F | F | T | F | F | T | F | T | T | F | F | T | 0 | T | T | F | fish |
| catfish | F | F | T | F | F | T | T | T | T | F | F | T | 0 | T | F | F | fish |
| cavy | T | F | F | T | F | F | F | T | T | T | F | F | 4 | F | T | F | mammal |
| cheetah | T | F | F | T | F | F | T | T | T | T | F | F | 4 | T | F | T | mammal |
| chicken | F | T | T | F | T | F | F | F | T | T | F | F | 2 | T | T | F | bird |

Figure 3.8: Zoo Database

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | Marital\_status\_ | Gender | Bluetooth\_Headphone | Car\_Video | Home\_th\_System | Mp4\_Mp3 | Recording\_ph | Television | VCD\_DVD | Class\_Customer |
| 13 | M | F | T | T | F | F | F | F | F | MODERATE |
| 14 | M | M | F | T | F | F | F | F | F | MODERATE |
| 15 | S | F | F | T | F | F | F | F | F | MODERATE |
| 16 | S | F | F | F | T | F | F | F | F | LOW |
| 17 | M | M | F | F | T | F | F | F | F | LOW |
| 18 | S | F | F | F | T | F | F | F | F | LOW |
| 19 | S | M | F | F | T | F | F | F | F | LOW |
| 20 | S | M | F | T | F | F | F | F | F | MODERATE |
| 21 | S | M | F | T | F | F | F | F | F | MODERATE |
| 22 | S | F | F | T | F | F | F | F | F | MODERATE |
| 23 | M | M | F | T | F | F | F | F | F | MODERATE |
| 24 | M | M | F | T | F | F | F | F | F | MODERATE |
| 25 | M | M | F | T | F | F | F | F | F | MODERATE |
| 26 | M | M | F | F | T | F | F | F | F | LOW |
| 27 | S | M | F | F | T | F | F | F | F | LOW |
| 28 | M | M | F | F | T | F | F | F | F | LOW |
| 29 | M | F | F | F | T | F | F | F | F | LOW |
| 30 | M | M | F | F | T | F | F | F | F | LOW |
| 31 | M | F | F | F | T | F | F | F | F | LOW |
| 32 | M | F | F | F | T | F | F | F | F | LOW |
| 33 | M | F | F | F | T | F | F | F | F | LOW |
| 34 | M | M | F | F | T | F | F | F | F | LOW |

Figure 3.9: Retail Dataset

#### Software Architecture

The term software architecture intuitively denotes the high level [structures](http://en.wikipedia.org/wiki/Structure) of a [software system](http://en.wikipedia.org/wiki/Software_system). It can be defined as the set of structures needed to reason about the software system, which comprise the software elements, the relations between them, and the properties of both elements and relations. The software architecture of our system is as given below:

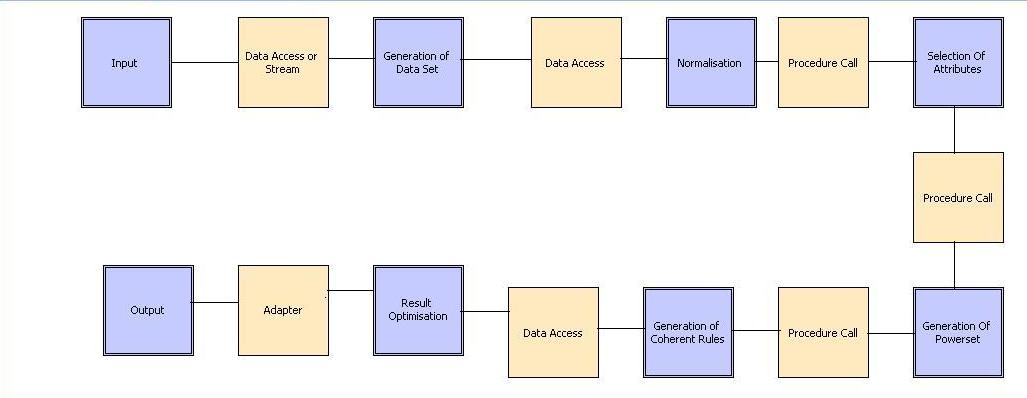


Figure 3.9: Component Connector Architecture Diagram

# IMPLEMENTATION

## Implementation Details

The main aim of the project is to generate accurate coherent rules. We are planning to accomplish this task of generating coherent rules according to the implementation plan given below:

* 1. Firstly we eliminate redundancies by keeping unique transaction ids associating with a specific group of items. In this manner we cluster them and each cluster is identified with unique id.
  2. Then after that we can have access to unique transactions present in the database table. Further, we also have access to the number of items present in a particular group.
  3. Then we apply domain driven approach by giving access to the data belonging to a particular domain only. Then processing can be done and rules can be generated for the particular domain only.
  4. The main advantage of this approach is that there is no need for the support. Confidence can be provided and thus the rules are generated without the need of support. This is done basically by the use of propositional logic
  5. Further we also move towards accuracy by considering negative association rule mining which is done by considering both possibilities namely of all present and all absent.
  6. In this manner the frequent items are generated and finally the rules are constructed accordingly.

|  |  |
| --- | --- |
| MONTH | WORK |
| July | 1. Search for the project topic. 2. Selection and study of the approved topic. |
| August | 1. Hardware and software planning. 2. Selection and study of the database. |
| September | IMPLEMENTATION-1   1. General classification of data. 2. Graphical user interface design. |
| October | IMPLEMENTATION-2  Domain driven knowledge based classification. |
| January | Generation of Coherent Rules.  Applying Coherent rules. |
| February |
| March | Final Implementation. |
| April | Testing. |

Figure 4.1 Project Plan

## Results

The proposed technique is tested on the subset of retail demo dataset available on msdn site. The sample subset of data of 139 customers for 7 different products is selected for experiment. Customer data is classified as High, Moderate and Low customers based on their features. The aim of the experiment was to find the association of demographic features of customer, product buying pattern and the class of customers which could help to give a promotional campaign on different products based on customer class and their demographic feature. The experimental setup used SQL server 2008 and DOTnet technology for implementation purpose. As per the proposed framework, domain knowledge method and feature selection method is used to generate coherent rules. Table 5 displays the subset of coherent rules generated after applying the algorithm given in previous section. Using domain knowledge method, Marital\_status and Gender from the demographic data is selected. From the transactional data only the interesting products are selected. For example, if we select certain attributes like Car\_Video and Bluetooth\_Headphone from the entire set of attributes then rules will be generated only for those selected products and thus we can reduce the complexity by reducing the number of computations. Further, provision has also been given to select the feature of the selected attributes which further help in achieving the desired results and also reduce the number of computations to a large extent. For example, after selecting attributes like *Car\_Video* and *Bluetooth\_Headphone*, we can further reduce computations by providing user an option of whether one is interested in generating rules for presence of the product or absence of product. Accordingly, then the rules will be generated which help the user in making promotional events to increase the business of their products. Table 6 given below displays the results after selecting 7 attributes namely Marital\_status, Gender, Bluetooth\_Headphone, Car\_Video, Home\_theater\_system, Mp4\_Mp3 and Television and the values of the above mentioned attributes that are selected. The domain driven concept is further expanded to the selection of the class labels for which rules needs to be generated. Finally only the subset of the rules with static as well as transactional features having lift>=1 are selected as shown in Table 7. Those rules can be reduced by applying pruning technique as given in previous section. After pruning and combined mining, the rules generated with the selected features are shown in Table 7. Total 8 combined rules have been generated. This result shows that how the different class customer with different demographic characteristics changes the product pattern. In Table 6, rule 3 states {Gender: Female, Bluetooth\_Headphone: False, Car\_Video: False🡪 HIGH}. This rule states that if Class is HIGH and the individual is female then there is minimal possibility of that individual buying the products like *Car\_Video* and *Bluetooth\_Headphone*. So schemes related to these products need not be provided to that individual. On the contrary, that individual must be provided with the schemes related to products in which he/she is interested which in turn increases the possibility of purchasing the product, consequently increasing the sales of the product intelligently. This saves a lot of work and effort required. Such type of knowledge helps to give the promotional campaign on products with respect to the class of customer and their demographic features. The rules are readable and understandable to human to take business decision and can reduce the cost of promotion.

The results of the same are given as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Rules** | **support** | **Confi-dence** | **lift** | **Lift**  **A** | **Lift**  **B** | **Irule**  **A** | **Irule**  **B** |
| **1** | **Car\_Video\_False-->HIGH** | 59/139 | 46 | 0.78 | - | - | - | - |
| **2** | **Bluetooth\_Headphone\_False-->HIGH** | 37/139 | 56 | 0.95 | - | - | - | - |
| **3** | **Gender\_Female-->HIGH** | 58/139 | 73 | 1.24 | - | - | - | - |
| **4** | **Gender\_Female, Car\_Video\_False -->HIGH** | 58/139 | 79 | 1.34 | 1.339 | 1.34 | 1.339 | 1.339 |
| **5** | **Gender\_Female,Bluetooth\_Headphone\_False-->HIGH** | 37/139 | 84 | 1.42 | 1.424 | 1.42 | 1.4237 | 1.4237 |
| **6** | **Gender\_Female,Bluetooth\_Headphone\_False,**  **Car\_Video\_False-->HIGH** | 37/139 | 95 | 1.61 | 1.61 | 1.61 | 1.6102 | 1.6102 |
| **7** | **Marital\_status\_Married-->HIGH** | 47/139 | 53 | 0.9 | - | - | - | - |
| **8** | **Marital\_status\_Married, Car\_Video\_False-->HIGH** | 47/139 | 57 | 0.97 | 0.966 | 0.97 | 0.9661 | 0.9661 |
| **9** | **Marital\_status\_Married,Bluetooth\_Headphone\_False**  **-->HIGH** | 30/139 | 60 | 1.02 | 1.017 | 1.02 | 1.0169 | 1.0169 |
| **10** | **Marital\_status\_Married,Bluetooth\_Headphone\_False,**  **Car\_Video\_False-->HIGH** | 30/139 | 67 | 1.14 | 1.136 | 1.14 | 1.1356 | 1.1356 |
| **11** | **Marital\_status\_Married, Gender\_Female-->HIGH** | 46/139 | 82 | 1.39 | 1.39 | 1.39 | 1.3898 | 1.3898 |
| **12** | **Marital\_status\_Married,Gender\_Female,**  **Car\_Video\_False-->HIGH** | 46/139 | 87 | 1.47 | 1.475 | 1.47 | 1.4746 | 1.4746 |
| **13** | **Marital\_status\_Married,Gender\_Female,**  **Bluetooth\_Headphone\_False-->HIGH** | 30/139 | 91 | 1.54 | 1.542 | 1.54 | 1.5424 | 1.5424 |

Table 4.1: Coherent Rules

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Rules** | **support** | **Confid-**  **ence** | **lift** | **Lift**  **A** | **Lift**  **B** | **Irule**  **A** | **Irule**  **B** |
| **1** | **Car\_Video\_False-->HIGH** | 59/139 | 46 | 0.78 | - | - | - | - |
| **2** | **Bluetooth\_Headphone\_False-->HIGH** | 37/139 | 56 | 0.949 | - | - | - | - |
| **3** | **Gender\_Female-->HIGH** | 58/139 | 73 | 1.237 | - | - | - | - |
| **4** | **Gender\_Female, Car\_Video\_False-->HIGH** | 58/139 | 79 | 1.339 | 1.339 | 1.339 | 1.33898 | 1.33898 |
| **5** | **Gender\_Female, Bluetooth\_Headphone\_False**  **-->HIGH** | 37/139 | 84 | 1.424 | 1.4237 | 1.4237 | 1.42373 | 1.42373 |
| **6** | **Gender\_Female,Bluetooth\_Headphone\_False,**  **Car\_Video\_False--->HIGH** | 37/139 | 95 | 1.61 | 1.6102 | 1.6102 | 1.61017 | 1.61017 |
| **7** | **Marital\_status\_Married-->HIGH** | 47/139 | 53 | 0.898 | - | - | - | - |
| **8** | **Marital\_status\_Married, Car\_Video\_False**  **-->HIGH** | 47/139 | 57 | 0.966 | 0.9661 | 0.9661 | 0.9661 | 0.9661 |
| **9** | **Marital\_status\_Married, Bluetooth\_Headphone\_False**  **-->HIGH** | 30/139 | 60 | 1.017 | 1.0169 | 1.0169 | 1.01695 | 1.01695 |
| **10** | **Marital\_status\_Married,Bluetooth\_Headphone\_False,**  **Car\_Video\_False-->HIGH** | 30/139 | 67 | 1.136 | 1.1356 | 1.1356 | 1.13559 | 1.13559 |
| **11** | **Marital\_status\_Married,Gender\_Female,**  **Car\_Video\_False-->HIGH** | 46/139 | 87 | 1.475 | 1.4746 | 1.4746 | 1.47458 | 1.47458 |
| **12** | **Marital\_status\_Married,Gender\_Female,**  **Bluetooth\_Headphone\_False-->HIGH** | 30/139 | 91 | 1.542 | 1.5424 | 1.5424 | 1.54237 | 1.54237 |
| **13** | **Gender\_Female,Bluetooth\_Headphone\_True**  **-->MODERATE** | 12/139 | 34 | 1.478 | 1.4783 | 1.4783 | 1.47826 | 1.47826 |

Table 4.2: Coherent Rules with Domain Driven Approach

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Rules** | **support** | **confid-ence** | **lift** | **Lift**  **A** | **Lift**  **B** | **Irule**  **A** | **Irule**  **B** |
| **1** | **Gender\_Female,Car\_Video\_False-->HIGH** | 58/139 | 79 | 1.339 | 1.339 | 1.339 | 1.33898 | 1.33898 |
| **2** | **Gender\_Female, Bluetooth\_Headphone\_False**  **-->HIGH** | 37/139 | 84 | 1.424 | 1.4237 | 1.4237 | 1.42373 | 1.42373 |
| **3** | **Gender\_Female,Bluetooth\_Headphone\_False,**  **Car\_Video\_False-->HIGH** | 37/139 | 95 | 1.61 | 1.6102 | 1.6102 | 1.61017 | 1.61017 |
| **4** | **Marital\_status\_Married,**  **Bluetooth\_Headphone\_False-->HIGH** | 30/139 | 60 | 1.017 | 1.0169 | 1.0169 | 1.01695 | 1.01695 |
| **5** | **Marital\_status\_Married,Bluetooth\_Headphone\_False,Car\_Video\_False-->HIGH** | 30/139 | 67 | 1.136 | 1.1356 | 1.1356 | 1.13559 | 1.13559 |
| **6** | **Marital\_status\_Married,Gender\_Female,**  **Car\_Video\_False-->HIGH** | 46/139 | 87 | 1.475 | 1.4746 | 1.4746 | 1.47458 | 1.47458 |
| **7** | **Marital\_status\_Married,Gender\_Female,**  **Bluetooth\_Headphone\_False-->HIGH** | 30/139 | 91 | 1.542 | 1.5424 | 1.5424 | 1.54237 | 1.54237 |
| **8** | **Gender\_Female,Bluetooth\_Headphone\_True**  **-->MODERATE** | 12/139 | 34 | 1.478 | 1.4783 | 1.4783 | 1.47826 | 1.47826 |

Table 4.3: Combined Rules after pruning process

The following are a few screenshots from the system implemented. This is performed on the Zoo Table whose details have been provided in the previous section.

### User Interface – Screen 1

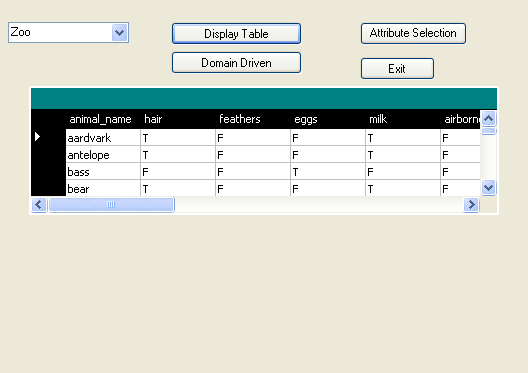


Figure .: Initial User Interface –Screen 1 Display

**Details:**

* The screen has been designed keeping in mind the dynamism property. Accordingly provisions are given to accordingly generate rules for any table. This is done by selecting the table from the drop down list provided above. Further, screen also contains a grid in which all the records of the database are displayed. This is done by clicking the button provided namely “Display Table”.
* The process of rule generation proceeds by generating comma seperated value (CSV) of the records in the table selected. So accordingly provision has been given to generate CSV form by clicking the CSV button provided. The CSV data will be displayed in the grid provided.

### Introduction of Domain Driven Concept

* The screen given below highlights the various properties / characteristics of the table selected. So the user according to his interest can view the associations in the form of rules after giving the selected attributes for further processing.

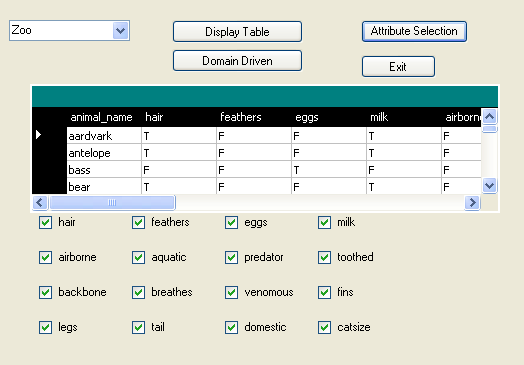


Figure .: User Interface Screen providing domain driven concept

As given in the screen below user selects the attributes for which he is interested.

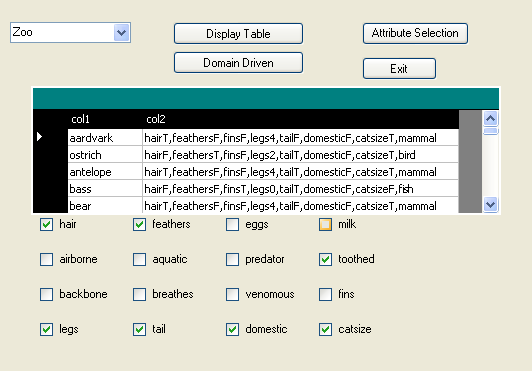


Figure .: Usage of Domain Driven Concept

After the selection of attributes in the previous screen, further deep level mining is provided to gain more precise and accurate results . This is done in the form of whether the user is interested in finding associations for presence or absence of the characteristic selected in the previous screen. This can be done by selecting the button “Domain Driven” after the selection of specific characteristics is done. Further mining has also been provided by giving a provision to the user to select the classes for which on is interested, thus giving more control to the user.

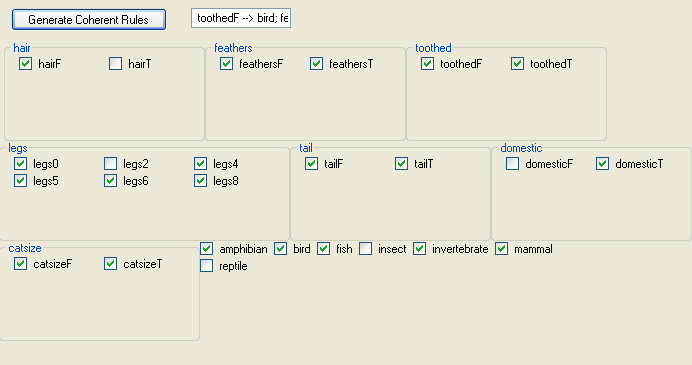


Figure .: Screen displaying domain mining at a deeper level

## 4.2.3 Coherent Rules Generation

Finally this screen appears when the user clicks “Generate Coherent Rules” after the domain selection process is complete. It first displays the number of rules generated and then the rules are displayed in the grid in a new form along with the lift , IRule which help user in assessing the certainty of the rule.



Figure 4.6: Total number of rules

Rules generated are displayed in the grid as follows:

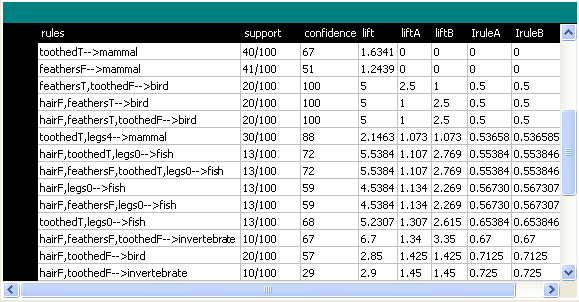


Figure .: Final generated Coherent Rules

# TESTING

## Test Cases

A specific executable [test](http://www.businessdictionary.com/definition/test.html) that [examines](http://www.businessdictionary.com/definition/examine.html) all aspects including inputs and [outputs](http://www.businessdictionary.com/definition/output.html) of a system and then [provides](http://www.businessdictionary.com/definition/provide.html) a [detailed](http://www.businessdictionary.com/definition/detailed.html) [description](http://www.businessdictionary.com/definition/description.html) of the steps that should be taken, the [results](http://www.businessdictionary.com/definition/result.html) that should be achieved, and other [elements](http://www.businessdictionary.com/definition/element.html) that should be identified.

There are various test cases built in this project to check generation of records.

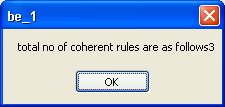


Fig 5.1 rule generation test case

## Performance Testing

Performance testing is done to provide stakeholders with information about their application regarding speed, stability and scalability. More importantly, performance testing uncovers what needs to be improved before the product goes to market. Without performance testing, software is likely to suffer from issues such as: running slow while several users use it simultaneously, inconsistencies across different operating systems and poor usability. Performance testing will determine whether or not their software meets speed, scalability and stability requirements under expected workload. Applications sent to market with poor performance metrics due to nonexistent or poor performance testing are likely to gain a bad reputation and fail to meet expected sales goals. Also, mission critical applications like space launch programs or life saving medical equipments should be performance tested to ensure that they run for a long period of time without deviations.

* **Load testing –** Checks the application’s ability to perform under anticipated user loads.

In our project there is only single user dealing with the system at a time. So, Load testing was not an issue.

* **Stress testing –** Involves testing an application under extreme workloads to see how it handles high traffic or data processing .The objective is to identify breaking point of an application.

In this project, according to user’s selection of attributes the flow of processing of data varies. e.g. Selection of all the attributes and classes as domain more than 15,000 records gets generated.

* **Scalability testing**– The objective of scalability testing is to determine the software application’s effectiveness in “scaling up” to support an increase in user load. It helps plan capacity in addition to software system.

## Unit Testing:

In [computer programming](http://en.wikipedia.org/wiki/Computer_programming), **unit testing** is a method by which individual units of [source code](http://en.wikipedia.org/wiki/Source_code), sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In [procedural programming](http://en.wikipedia.org/wiki/Procedural_programming) a unit could be an entire module but is more commonly an individual function or procedure. In [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming) a unit is often an entire interface, such as a class, but could be an individual method.

There are different modules built in our project which are separately tested as unit testing and then combined together for system testing.

* + 1. The first module deals with the selection of table, creating stream that connects database, displaying comma separated form, and selection of attributes of interest.
    2. The next module does the important function of calculating powerset of attributes selected in module one.



Fig 5.2 Powerset generation

* + 1. The last module is the heart of the system, it actually generates Coherent rules and deals with representation of rules.

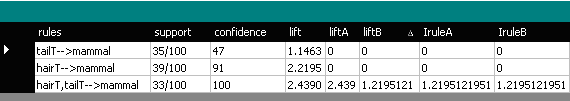


Fig 5.3: Coherent rules

After testing all the module separately, we proceed to system testing.

## System Integration Testing

System Integration Testing (S.I.T.) is the testing of the sub-systems, as a whole, to ensure that they work as a [system](http://www.sharpy.dircon.co.uk/index_files/DefinitionOfSystem.htm).

The testing should be driven by the [requirement specification](http://www.sharpy.dircon.co.uk/index_files/RequirementSpecificationExplanation.htm) and the System Integration (S.I.) engineer’s knowledge of what the system should do.

No requirement specification can fully define a system, their will always be gaps in what the requirement specification states and what is expected of a system.

As an extreme, the requirement specification will not state that the system should not burst into flames after an hour of operation but this is certainly a requirement, i.e. no one would be happy if the system burst into flames after an hour of operation.

This is where the experience of the S.I. engineer comes into play, as he/she carries out their work they need to keep their eyes open for any faults or problems.  The S.I. engineer needs to be aware that faults will be present that will only be found through informal testing, i.e. running the system and seeing what happens.

The S.I.T. should ensure that the system meets the requirements of the formal requirements specification document and also that any implied, or ‘common sense’ requirements are met.

# CONCLUSION AND FURTHER WORK

## Conclusion

This paper basically introduces the technique of finding association rules and the drawbacks of the same. Through this introduction, it highlights the need of introducing the concept of coherent rules and the benefits of the same. Coherent Rules gives both positive and negative rules without giving support threshold. Further, it enhances the concept by adding certain features like domain-driven knowledge which further embellish the quality of the result generated in terms of optimization of time and space complexity. The domain driven concept bestows the idea of selection of data features before generating the frequent patterns. Hence, the users have the freedom to select the product on which the company wants to give promotional campaign before generating the rules. Through domain driven user can select the customer characteristics from static customer data. This technique is giving the rules for customers with their change of class as the transactional characteristic changes. This proposed technique gives more actionable rules than traditional association technique which help to improve business process.

* 1. **Further Work**

The coherent rules mining framework and the support and confidence framework share a common goal of discovering interesting association rules. The former has a larger search space among the positive and negative association rules to report association rules that can be reasoned as logical implications. Many of the extensions of a support and confidence framework are applicable also to a coherent rules mining framework. It would also be interesting to exploit pruning properties such as monotone properties to discover coherent rules. Another possible extension to the coherent rules mining framework is to consider the statistical significance test in generating coherent rules. A detailed study on the topic of statistical tests is required to identify a suitable test. Coherent rules that pass some statistical significance test would be interesting and not likely to exist by chance. It is possible to utilise and apply a coherent rules mining framework to discover coherent rules that deviate from user expectations. These deviations are important to show significant differences between coherent rules that are logically implied from a dataset and the rules according to user expectation. In propositional logic, there are several modes of implication. We have restricted our research to consider only the logical equivalences. It is also possible to map association rules to the implication mode of material implication. A logical equivalence is a subset to material implication. Finding association rules without a minimum support threshold by mapping to material implications could find more association rules because material implications are less constrained. The possibility of this future work based on material implication is depicted in Figure 6.1.

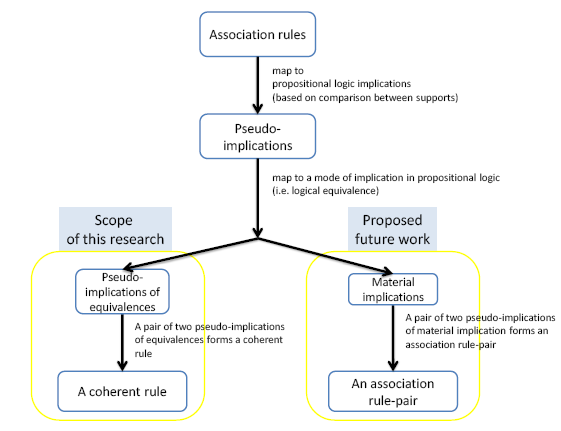


Figure 6.1: Future Work based on Pseudo-Implications

Association rules found from decoupling pairs of pseudo-implications mapped to material implications are also pseudo-implications that have material implication properties. This study suggests that our framework for discovering coherent rules offers a technique for data mining that overcomes the limitations associated with existing methods and enables finding association rules among the presence and/or absence of a set of items without a preset minimum support threshold. The results justify continuing research in this area in order to increase the body of scientific knowledge of data mining specifically association rules, and to provide practical support to those involved in data mining activities.

**REFERENCES**

[1] C. Longbing, “Introduction to Domain Driven Data Mining,” Data Mining for Business Applications, L. Cao, P.S. Yu, C. Zhang, and H. Zhang, eds., pp. 3-10, Springer, 2008.

[2] Alex Tze Hiang Sim et. Al., “Logic based pattern discovery,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 6, JUNE 2010

[3] X. Wu, C. Zhang, and S. Zhang, “Efficient Mining of Both Positive and Negative Association Rules,” ACM Trans. Information Systems, vol. 22, pp. 381-405, 2004.

[4] Y. Zhao, H. Zhang, L. Cao, C. Zhang and H. Bohlscheid “Combined pattern mining: From learned rules to actionable knowledge,”  Proc.AI,  pp.393-403 2008

[5] Arti Deshpande and Anjali Mahajan “Domain Driven Multi-Feature Combined Mining for retail dataset,” International Journal of Engineering and Advanced Technology (IJEAT) Volume-2, Issue-3, February 28, 2013 Available: <http://www.ijeat.org/article.php>

[6] J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” Morgan Kaufmann Publishers, 2006.

[7] [Michael Genesereth](mailto:genesereth@stanford.edu), “Propositional Logic,” Internet: http://logic.stanford.edu/classes/cs157/ 2009/notes/chap02.html, 2009