HYBRID MODEL FOR ERROR DETECTION AND CORRECTION IN DATABASE

H. M. A. Mohit Chowdhury	13.01.04.136
A. M. Ariful Islam	13.01.04.135
Ayesha Akter	13.01.04.117
A. K. M. Bayezid	12.01.04.032

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH
MAY 2017

HYBRID MODEL FOR ERROR DETECTION AND CORRECTION IN DATABASE

A Project and Thesis Report
Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering

By

H. M. A. Mohit Chowdhury	13.01.04.136
A. M. Ariful Islam	13.01.04.135
Ayesha Akter	13.01.04.117
A. K. M. Bayezid	12.01.04.032

Supervised by
Ms. Shanjida Khatun
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH
MAY 2017

Declaration

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of Ms. Sanjida Khatun, Assistant Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project & Thesis I and CSE4250: Project & Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

....

(H. M. A. Mohit Chowdhury)

....

(A. M. Ariful Islam)

....

(Ayesha Akter)

....

(A. K. M. Bayezid)

Countersign by

....

Ms. Shanjida Khatun
Assistant Professor, Dept. of CSE
Ahsanullah University of Science and Technology

Abstract

Data is the collections of information. It's the most important thing in today's world. World's information moves on online and becomes digitized. Web is an open source and largest database. But there may error in a database and sometimes we have to pay a lot for this error. But it is critical to detect and correct data. Intrinsic and extrinsic techniques can be sufficient to correct data. Fetching data from Web or any external sources and matching with our existing database to detect errors (Inaccuracy, Duplication, Incompleteness and Data currency) and find the correct value for correction. Inconsistent data detection and correction using FDs(Functional Dependencies) and modified apriori algorithm and fetch data from web will give a reliability. In our research work we proposed a three modular approach for detecting and correcting error in a database and thus to pursue an effective way to enhance data quality and reliability.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We intend to provide web based techniques to detect different types of error such as Data currency, inaccuracy, duplication and inconsistency error. Data currency [1] is the value which exists in the database does not match with the current result of the web search or any other data sources. For example, suppose our database records that the prime minister of Bangladesh is Khaleda Zia. If we search in the Web about our prime minister, it results that the Prime minister of Bangladesh is Shekh Hasina. The record of our database is out of date. A database has two values, numeric and nominal. But to determine the inaccuracy [1] of nominal value is sometimes very difficult. So, we are working with the numeric values. In this case we set a real value for a system, then we experiment some other values and measuring the distance between of the experimental value and real value we are checking inaccuracy. For example, suppose we need 2 hours to make a journey from Dhaka to Narayangonj. Then we experiment required time to make that journey another two way. First way we need two and half hour and second way we need two and fifteen minutes. Now we measure the difference between the real value and experimental value. The value which is approximate to the real value, it is accurate. Duplication [1] is the way of checking whether the values are the same in a database. Inconsistency error [1] is the process of finding effective data consistency using the information of database and WWW. In tour ongoing work we are thinking to propose an effective way to detect the above mentioned error successfully.

1.1 Research Statement

Data is a collective fact. So External Sources data (WWW) or any other sources data are a collective term which refers to any type of data we might retrieve from the largest Database WWW or any other sources data. Any types of information may be data. For example, what our competitors are selling published government data, football scores, international news etc. These data can be used to investigate competitors, observe potential customer, keep track of channel partners, generate leads, and build apps, and so on. The information is massive, dynamic, and inconsistent. It is a difficult task on which way to solve data inconsistency, duplication, incompleteness and out-of-datedness problem. One solution varies from the other solution. For example, if we want to correct data incompleteness, we have to save a correct value. On the contrary, if we want to check data duplication, we have to check whether two or many values represent the same thing. In order to detect inconsistency, we are trying to use FD (Functional Dependencies) and CFD (Conditional Functional Dependencies) [2]. But in this case information must be sufficient to detect inconsistency constrains. Otherwise, we won't find efficient result of FDs and CFDs rule [2]. We have decided to use and detect error of the integrated information of the WWW or any other sources data. We have used Apriori Algorithm on the association rule to detect error.

1.2 Motivation

This data mining part discusses the process of achieving sequenced, accreted, deduplicated, completed and updated data which lying World Wide Web (WWW) as a knowledge base .Basically WWW is different from knowledge base having huge information, poor data quality, inconsistency and open system as well. So it is so much difficult to retrieve correct information of data from WWW. But we can get improved data quality of a database using exiting information. We don't know how to use exiting information. To remove data duplication, incompleteness, accuracy and inconsistency from WWW is so much difficult[3]. It is a challenging issue to us to implement, for this reason we are interested. To retrieve a correct value we can get the data which is in the completeness. By checking attributes and tuples we can identify if data information is duplicate or not. It is difficult to get correct information from database and WWW by seeking qualitative data constrains or using any effective repairing methods. We try to implement out-of-dateness and inconsistency of information of a database and WWW.

1.3 Aims

Our aim is to make an efficient process of detecting error from a largest data set. We have made a new approach with the Apriori algorithm [4] applying on the association rule .When we calculate support and confidence there appears some pattern set and rules that are unnecessary. We are trying to reduce the unnecessary things. If we use top-k algorithm, some pattern sets are reduced. But there is a problem. If there is some strong pattern set under the top-k, they are also reduced. So, our objective is how to reduce some pattern set in accordance with our database by calculating support and confidence [5].

Maintaining the research integrity correct data is inevitable. Massive, dynamic and inconsistent information from the largest dataset are needed to be consistent, complete and accurate. Appropriate data and their correct use can eliminate the likelihood of errors occurring. From that sense we are working to detect and correct errors in an effective way.

1.4 Organization of the thesis

In chapter 2, we have discussed about Association rule analysis and other terms and definitions needed to understand our work. In chapter 3, we have discussed about previous works on error detection and correction procedure. In chapter 4, we have discussed and criticized our works and introduced our experimental results and the limitations faced to do the job done. In chapter 5, we have concluded our work and introduced about our probable future approach.

Chapter 2

Preliminaries

2.1 WWW data retrieval constrains

We can search a keyword into www and get many results related this keyword. From these results we can extract the relative information. By evaluating the information we will compare with our existing information and can get the most effective results. But which is the main challenge is that the extraction of relative information as www is a vast recourse[2][6][3].

2.2 Data inconsistency

Data inconsistency occurs when similar data is found in different formats in different sources. For example: Two information of same person is kept in two different sources-John is an employee, salary of Jhon recorded in these two data sources are 40k and 50k subsequently. FD and CFDs are take into account to obtain consistency constrains. The most important aspect of finding effective FD and CFD rules is that the information table must be sufficient. To determine whether a FD (CFD) is right or wrong, it is considered the characteristics and distribution of the data on the WWW[2]. It helps to get relatively consistent values from the WWW. Data inconsistency may occur at three different levels. Data inconsistency may occur at three different levels

- Schema level
- Data representation level
- Data value level

In schema level inconsistency occurs when same data have different schema. For example,

Here the same employee id 519 has different address and skill. In Data representation level inconsistency occurs when data is represented in different measurement System. Value level inconsistency occurs when it is found two objects from different data sources are the version of each other[7].

id	address	skill
429	87 Sycamore Grove	Typing
426	87 Sycamore Grove	Shorthand
519	94 Chestnut Street	Public Speaking
519	96 Walnut Avenue	Carpentry

Table 2.1: Employee Skill

2.3 Duplication

Data duplication occurs when same logical value has different representation. Besides, it may occur either if a field is repeated in two or more tables or if the field is repeated within the table. For example,

name	age	profession
John	48	Teacher
John	48	Driver
John	49	Employee

Table 2.2: Data Duplication

The table shows that the same name is repeated three times and the age is repeated twice. Thus duplication occurs[8].

2.4 Duplication Detection

In [2] when two tuples are same the common model of that tuples

$$SIM(t,t^{'}) = \alpha \times sim(t[A_0],t^{'}[A_0]) + \cdots + \beta \times sim(t[A_i],t^{'}[A_i]) \tag{2.1}$$

In which sim$(t[A_0],t^{'}[A_0])$ is represented the similarity of the attribute – of t and $t^{'}$ weights of different attributes denoted by α and β.There are different forms may be appeared as like “IEEE” and “institute of Electrical and Electronics Engineers”. In this circumstance, distance-based functions can’t calculate the similarity. Having the same sets of attributes and relations of two objects, the possibility of the similarity of that objects is increased, sometimes very high. According to this idea, related objects can be retrieved from the WWW and a relation graph is established to put them. From this graph we can easily evaluate the similarity of two attribute values[9][10].

2.5 Inaccuracy

Data inaccuracy means the value which exists in the database does not match with the true value or there is no adjacency of values in a database to the true values of the sources[5].A database has two values, numeric and nominal. But working with nominal values to determine inaccuracy is challenging. It is convenient to work with numeric value to determine inaccuracy. In this case, there is a real value. Comparing the distance between the experimented values with the real value inaccuracy is detected. To determine the relative currency of rank values, update time, reference and distribution a model is needed to build [6].But it is difficult to access all those in a model for the various kind of information from different database or WWW[2].

fn	ln	age	height	status
John	Clark	14	1.70	Single
J.	Clark	14	1.69	Married
John	Clark	45	1.60	Single

Table 2.3: Data Inaccuracy

The Table represents a person Schema. Each tuple represent the (fn, ln, age, height, marital status) of a person. First row represent true information for John. From these we can terminate that third row [age, height] are inaccurate than second row[age, height] as they are not closer to the true value of John while second row[fn, status] is more inaccurate then third row.

2.6 Out of datedness

If the available data does not match with the Current result of the database, the available data is out-of-dated. It can be detected by checking the date and can get the update information. To determine the partial currency order of the values of same column of a table is the out of datedness detection process. Besides, the integrated values across a database and the WWW. In this case, assumption is that inherent information in the database is sufficient. If this assumption is dissatisfied, integrated value should be used. Different types of the primary attributes hidden on the WWW and the currency of an attached primary attribute determine the currency of an attribute value. For example, to determine whether "Khaled Mashud, Test match Captain of Bangladesh Cricket Team" is out of date. If we search "Test Match Captain of Bangladesh Cricket Team", get a list from the Web. Then we can resolve the currency order of nine Test match captains and we can decide that "Khaled mashu", Test match Captain of Bangladesh is out of date.

2.7 Association Rule

Association rule learning [11] is the process of discovering relations between variables in large dataset. It is proposed to identify strong rules discovered in dataset using some part of dataset. For exploring uniformity between products in large-scale transaction data recorded by point-of-sale systems in super shop can use association rule, based on the concept of strong rules. For example the rule {Onions ,Potatoes}→{Burger} found in sales record of a super shop would indicate that if a customer buys onions and potatoes together, they are likely to also buy burger meat. The above example is the application of association rule on Market basket analysis. We used association on our data mining process. For association rule we need to calculate Support and Confidence. To calculate support and confidence some unnecessary pattern set and rules appear. Our objective is to reduce unnecessary things. We can complete our task using top-k selection algorithm. Using top-k we can reduce unnecessary pattern set. In this case if there is some strong pattern set under the top-k, they are also reduced. So, to find approximate result we use Apriori Algorithm on Association rule. Apriori Algorithm is an efficient algorithm to find frequent item sets [11].

2.7.1 Goals of Association Rule

- To find all the itemsets that satisfies the minsup threshold. It means itemsets must have support (number of transactions) greater than the minimum support(large item-sets).These item-sets are defined as frequent item set.
- To derive all the high confidence rules from the frequent itemsets.It means that using the large item set have to generate expected rules that have confidence greater than the minimum confidence.

2.8 Association Rule Analysis

As we are working with data inconsistency, which refer to detect is there any irrelevance among a given data. We will find that where there is an irrelevant information with the other associated data in a row. For example, if we have a phone number we can say that person from a particular country or we can also say that a particular countries phone number should start with a unique code. If we take Bangladesh as our country then our phone numbers must start with +880, so we can say there is a dependency between country and phone number.

Name	Country	Phone Number
Tareq	Bangladesh	+8801733456214
Mark	USA	+1353682467833
Karim	Banglades	+1801678234671
Joi	USA	+1994582348234
Erik	USA	+1093857122334

Table 2.4: Country & Phone Number

We can say this dependency as-

$$\{phone\ number\} \rightarrow \{country\ code\}$$

With this association rule we can find strong relationship among data. If we take another example, if we have a postal code then we can find the area name. suppose, Dhaka – 1208 is a postal code, from this we can say the area should be Tejgaon Industrial area. Here postal code is strongly connected with area. By using association rule we can find many strong rule among the data. From the Table 2.4 we can observe, Karim's phone number start with +1 which is contradictory with his country. We consider to occurrence-

i) We can say his phone number is incorrectly started as his country is Bangladesh and it must start with +880

$$\{country\ code\} \to \{phone\ number\}$$

or

ii) We can say his country is incorrect as his phone number start with +1 which indicates the country should be USA.

$$\{phone\ number\} \to \{country\ code\}$$

Here is a conflict whether we should say his phone number should start with +880 or we should set his country as USA. So, for this dependency we have to generate some efficient rules for our analysis which should be more efficient for dependency.

We started working with Association Rule. Basically Association Rule is most suitable for genetic analysis and transection analysis. Association rule can find strong relationship where there is genetically dependent information. For example, if a person buy Coffee then we can say he/she also buy Biscuits, Milk, Sugar. We want to apply this technique for our research work in our domain based dataset which is the most challenging thing. Association analysis is suitable for discovering hidden relationship in large dataset.

TID	Items
T1	bread, milk
T2	bread, beer, egg
T3	milk, beer, egg
T4	bread, milk, egg
T5	cola, milk, juice
T6	bread, milk, juice, egg

Table 2.5: Super Shop Transaction

If we analysis with Table 2.6 we can find a strong relationship between the transaction. For example, we can find a relation between bread and milk. By analyzing this data we can predict that if a customer buys bread, most of the time he/she will buy milk and vice versa. We can express this relationship with {bread}→{milk} or {milk}→{bread} So, definitely by analyzing association rule we can predict the probability that if a customer by a product then he/she will buy the other particular product. For this hidden relationship, we started with association analysis for our research work to detect inconsistency in our data. This association rule will give us a relationship among our data and with these relational based rules we can find irrelevant information from the database which is contradictory with the other given information.

2.8.1 Itemset and Support Count

Let,

$$I = \{i_1, i_2, i_3, i_4, i_5, ...\}$$

be the set of all item in a transection line and

$$T = \{t_1, t_2, t_3, t_4, t_5, ...\}$$

be the set of all transection. Each transection ti contains a subset of items from itemset I. In association analysis, zero or more item is termed as itemset. If an itemset contains k item then it is referred as k-itemset. For example, from Table 2.5 {bread, milk, juice, egg, cola, beer} is an 6-itemset example. Null or empty itemset which does not contain any item. Transection width is called the number of items it contains. A transection t_j contains itemset X if X is a subset of t_j. For example, transection T_3 from Table 2.5 contains itemset {beer, egg} not {bread, milk}. An important property of association analysis is support count which refers the number of transection for a particular itemset. Mathematically we can represent this-

$$\sigma(X) = \{t_i | X \subseteq t_i,\, t_i \in T\} \tag{2.2}$$

where |.| denote the number of elements in a set. In the dataset shown in Table 2.5 support count for {bread, milk} is equal to 3 because there are only three transection that contains all two items.

But, if we consider some other dataset which is different from Table 2.5

fn	ln	cc	ac	phn	street	city	zip	salary	status
Mike	Col	44	131	4459	New Str.	DH	Z2	60k	single
Jim	Clark	44	131	4467	Old Str.	DH	Z2	60k	married
Mic	Colli	44	131	4478	New Str.	DH	Z2	60k	married
Joi	Tim	01	243	23678	Mtv	FT	A67	60k	single
Bob	Ric	01	243	23964	Mtv	FT	A67	60k	married
Rik	Jov	01	243	23084	Mtv	FT	A67	60k	single
Jovn	Jony	10	120	12567	Oris	HGT	67I	60k	married

Table 2.6: Employee Details

From Table 2.6, if we start analyzing this dataset, we have to calculate support count for a particular itemset. For example, if we consider ln for support count, it is hard to calculate support count for ln as there appear many data which is different from one another.

There are many ways to represent a dataset. The choice of representation of dataset can affect the cost of support count for candidate itemset. There are two type of representation of dataset-

- Vertical Data Layout
- Horizontal Data Layout

Table 2.5 is called Horizontal representation which is adopted by many association rule mining algorithm including Apriori algorithm.

Table 2.7 representation is called Vertical representation. Each transection id is associated with each item[11]

Bread	Milk	Beer	Egg	cola	Juice
1	1	2	2	5	6
2	3	3	3		
4	4	4			
6	5		6		
	6				

Table 2.7: Vertical Representation

Now if we represent Table 2.6 vertically then for ln there will be many columns for individual name which is impractical and if we want to represent the whole table vertically it will be a table that contains infinite number of columns for every individual items. Basically association rule mining is suitable transection based or genetic type analysis. But which is our main idea is to discover hidden relationship among the data. So, for our dataset we can calculate Distance which replaces the concept of Support Count

2.9 Enumerating Subsets

Figure 2.1 shows a systematic way for generating subset of 3-item. In level 1, all item set is followed by {A}, {B}. It is not followed by {C} or {D} because it is not possible to generate subset of 3-item from this combination as there left only one item if it is followed by {C} and zero item if it is followed by {D}.

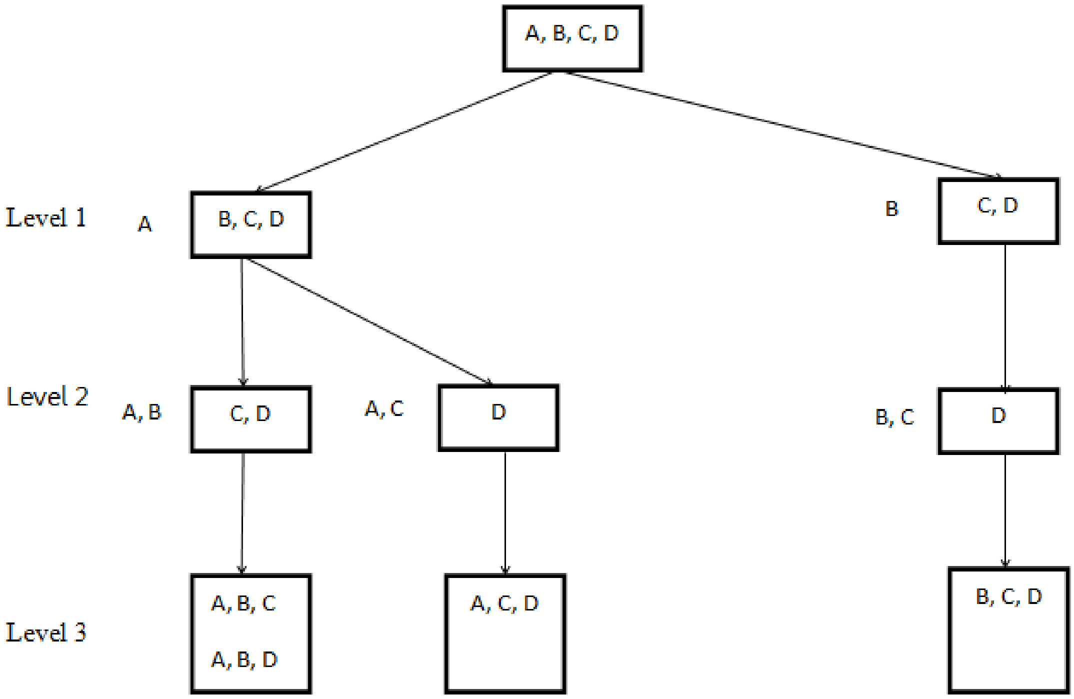


Figure 2.1: Subset Generation

For example, if we take it can represent subset of 3-item whose first element is

{A} and the rest of the two element from {B,C,D}. After determining the first element, level 2 shows the ways to select the second element. For example, {A,B} correspond the subset that begins with {A,B} and the rest of one element can be {C} or {D}. Finally level 3 represent the full subset of 3-item [11].

2.10 Apriori Algorithm

Apriori algorithm, Aprioritid Algorithm, AprioriHybrid algorithm, Tertius Algorithm are the association rule mining algorithm[12]. Apriori is the first association rule mining algorithm. It developed the use of support based pruning to control the exponential growth of candidate itemsets. It is a level wise search algorithm. It prunes many sets which are unlikely to be frequents before reading the database at each level.

2.10.1 Candidate generation

From the database D the algorithm finds the frequent set L. Our objective is to generate super set of the set of all frequent k-item sets from the given set of all frequent (k-1) item sets. If an item-set has minimum support, so do all subset of X. Thus all the (l+1) candidate sequence have been generated. Then they are read one by one and determined the support of these new generated candidates.

2.10.2 Pruning

From being considered for counting support, this step reduces the extension of (k-1) itemsets which are not found to be frequent. The algorithm checks which candidates are included in transaction t. This is done at each transaction and thus the last transactions are completed. Comparing with the support value some candidates are discarded. Those supports are less than the minimum support is discarded.

2.10.3 Drawbacks of Apriori Algorithm

- To generate candidate need much time, space and memory.
- To scan the database multiple times to generate candidate set.

2.11 Support and Confidence

We create subset and generate some rules by applying Apriori Algorithm [11] on association rules. The rule that has a very low support value may take place simply unexpectedly. For that reason support is a significant measure. In the context of business perspective a low support is uninteresting because it may not be productive to improve items that customers hardly buy together. For these reasons Support is usually take into account to eradicate uninteresting rules. For efficient discovery of association rules Support is inevitable [13]. we can calculate support in the following way-

$$Support(XY) = \frac{support\ count\ of\ (XY)}{total\ number\ of\ transaction\ in\ the\ database} \tag{2.3}$$

If the support of an item is 0.2%, it means only 0 .2 percent of the transaction contain purchasing of this item.

Confidence is how often the rule is found to be true. The reliability of the inference is measured by the confidence. For a given rule X→Y, the probability of Y to be present in transactions that contain X is related with the value of confidence. The higher the confidence, the more it is to be present in transactions. An evaluation of the conditional probability of Y given X is also provided by confidence.[13] we can calculate confidence in the following way-

$$Confidence(X|Y) = \frac{Support(XY)}{Support(X)} \tag{2.4}$$

If the confidence of the association rule X→Y is 70%, it means that 70% of the transactions that have X also have Y together.

2.12 Summary

In this chapter we are acquainted with the various error terms that are detected while working with the largest data set or Web. Moreover, the terms are used in our proposed techniques. In later chapter we have shown elaborately how these terms work effectively to detect and correct errors.

Chapter 3

Related Works

3.1 Error detection and correction based on web

How to find out data information omissions significantly and service ably repair that delusions is so much ticklish to attain transcendent data exorcism in diverse infliction's. The proficiency's that are oft subsistence, use exclusively idiosyncratic errand that gives sufficient and favorably-structured information. An overture is analogous to that.

They constitute a platform, be based on web having sundry information models and unerring information where incompleteness errors occur using some techniques [2]. In conformity with this proposal, they have taken an attempt to exploit information from web to find out the information errors and unerring that errors effectually.

But we predominantly focus to construct efficient procedure to find out the information errors from diverse largest data set and ensure the correctness of that data set as if it is used to provide correct information more effectively. In this chapter, we recapitulate the several approaches proposed in isolated research papers and conferences as a result we can gather knowledge about various proposed procedure related to our paper which help us to distinguish various papers and we can become clear that our paper is the distinct one.

3.1.1 Techniques based on web

In [1], they constructing a platform based on web proposed a set of web based technique that assures detecting the errors and correct that errors in a database applying dynamic procedure to retrieve the unerring information of a database which indicates incompleteness errors of information. Diverse algorithms that can find out errors and unerring that errors based on web reliability evaluation are proposed. A demo system provided to estimate the effectiveness of various proficiency is discussed in this chapter of our paper to know diverse procedures to fix the errors after detecting the errors to get unerring information of database.

- **Analytic data information to query construction**

They endeavored to find out traceless attributes of tuples whether subsist or not. If subsist then haphazardly elect affixed amount of tuples owned by tables to form training data set and based on this precise pattern of query used by web query engines is achieved. Using a web query engine such as Bing Search Engine spontaneously pick allied pages from web.

- **Looking forward to ranking in conformity with candidacy value**
  They extract from web pages which was previously produced for getting the values of earmarked candidacy. In conformity with candidacy values rank those values where the relations are taken into account between various types of objects across database or World Wide Web. This rank indicates where should have to correct and the priority of the errors that which error should have to correct first.

- **Data recuperation**
  Having the ranking of the data set it is facile to work in accordance with ranking. They chose the values which were top ranked, it means that the higher ranked values elected to use first. Then evaluated the web based data performance and after that it is provided the efficiency of this techniques.

Applying this technique they detected about sixty percent values which were missing from data set was placed in that database. The accuracy may decrease due to various format of values in distributed database and the largest amount of data information which are in the WWW(World Wide Web).

To detect errors of data information from a database and fixed that errors in that database to make sure that the database is an unerring database with proper information. They pre-eminently detect the errors of data information lying in the database to correct that errors using web information to insure an unerring database. After constructing a training data set arbitrary, extract information from web using web search engines as like as Bing Search API and then compare the information with the database. Depending upon this comparison they construct a ranking system which helps to correct the errors in an effective way. So it is quite easily possible to use the rank to construct an efficient database to the users effectively. In this way, they effectively use the web information to establish an efficiently workable database more accurately than previous database making the database a better one. It proves that find out the data information errors successfully and make the unerring data information of a database utilizing the data information from various pages present in the web.

3.2 The notion using multi-databases of data inconsistency detection

In this paper, they proposed an infrastructure and algorithm indicating the data inconsistency criteria to update data knowledge experimenting extraction of data from various real world examples.

They considered inconsistency of data mainly in two different types such as presenting same data in different databases severally and database maintaining failure which indicate equivalent data information having same values in diverse databases stored with values which is different. To correct the first one it is so much to rename the attributes, transform the framework, converse the values and map the domain. Detect the second errors when update the data information in a different databases. This paper was predominantly focusing on detecting incremental method of upgrading database based on real life case study.

3.2.1 Data Inconsistency Detection Process

This paper has several segments including the criteria given below:

- **Based on knowledge derive the data**
  They proposed an approach to find out analogous entities deducing missed data by utilizing ancillary knowledge. Invented knowledge or common sense that were enciphered by mappings which were a conceptual model.
- **Single database consistency problem**
  This problem arise when more than one attributes name indicates identical meaning given an error in the real world that conflicts the values. They introduced this problem as M-Consistencies. M-Consistencies are related to
  - Incremented Detection of errors
  - Effectiveness of that error detected previously
- **Multi database consistency problem**
  A multi database can be considered while integrated stratagem is completed. They proposed to take a unique key and expanded key for different entities from different databases. This is modeled by an entity uniquely in the multi databases. It is called as EI-Consistency by them.

They wished to conduct both the M-consistency and EI-consistency as an experiment but the main obstacle is space limitation. So their comparison was only the EI-consistency which related to the two incremental strategies.

- Incremented Error Detecting
- Non-materialization with free deletions

The proposal of this paper indicates using superfluous data knowledge the incremental detecting while updating the superfluous knowledge based data and using materialized view with very low communication cost at the time of updating user relations. The infrequent data knowledge involves usually less effective but more affordable for the incremental detection. This method efficiently find out the entities which appeared more than one and finally detects consistency more accurately[14].

3.3 Ensure data consistency using data integration

They proposed an estimated object-oriented model based on data defining the exorcism of data criteria from any source and based on this model they derived the solution for the inconsistency of data information from any database efficiently. According to their proposed system, considered a set of observations are performed to evaluate the efficiency of this model. They took an approach for decision making called "fuzzy multi-attribute" based on the criteria of data quality from any given source which was applied to elect "best" data from the source to remove data inconsistency ensuring the data as consistence.

3.3.1 Detecting and repairing data inconsistency

This paper used some algorithms within a model to repair data inconsistency but their main work for ensuring consistence data from any source of data in a database. The proposed procedure to make sure the data is consistence were:

- Proposed a data integration model to detect the inconsistency of data and resolved that data then the data is consistence.
- Fuzzy multi-attribute that makes decision for inconsistency of data used to make the data persistence. This processed the data source quality in conformity with qualitative values.
- Designed to perform an experiment evaluating the efficiency of the algorithms and skills they provided.

They wanted to detect data inconsistency first according to the previous work whether data in a same database conflicts each other. Identified the attributes that were analogous and defined by keys to identify the similar data in the real world. Then performed clustering process in conformity with the predefined keys. They proposed the following steps to construct the clustering query result for the solution of data inconsistency.

- **Getting Fusion Matrix**
  It ensures the quality of data source according to the qualitative criteria values. This values were defined "strong", "weak", "more weak" as qualitative category introducing fuzzy number which indicates the triangle of fuzzy number.
- **Scaling**
  In this part of the paper they retrieved the values of positive and negative criteria of the data source.
- **Fusion Matrix Construction for decision making**
  Find the weight vector of the quality criteria according to the weights of the data source. It can be provided by the users but making this process dynamic it should have to be ignored. Based on the weight vector the fusion matrix were constructed.

- **Compute Distances**
  To find out the positive and negative ideal solutions compared the triangular fuzzy number that defines the qualitative vector for those solutions. Achieved those ideal solutions measuring the "Euclidean Distance" after obtaining the candidate data source.

- **Membership Function for final solution**
  Defined a membership function calculating the membership degree of each data source candidate for positive ideal solution. Then sorted the vector according to membership function to get the final result. Attached to the query result solved the inconsistencies to ensure the data source as consistence to the users.

Track out the inconsistency errors by detecting the conflicted value from the same database to repair the inconsistencies. The conflicted values were denoted by keys which performed the clustering according to acquiring the fusion matrix defined by three category of fuzzy numbers. Scaling the positive and negative data source based on the fuzzy number computed the Euclidean distance that helped to find out the membership function which stands for an ideal solution of inconsistency. Performing some experiments they proved that their proposed system works effectively in the real world and succeeded about that to make sure the data as a consistence data[7].

3.4 Summary

In this chapter, we discussed the previous related works for detecting and correcting the errors of data information existing in any databases where existed three different types of error detection and correction described in three different papers representing diverse techniques, strategies, algorithms and various solved obstacles to detect errors and correct that errors effectively to make more user friendly to the users. The first proposed technique for detecting errors and correction that errors was based on the information of web database detecting the errors of data information and correcting that errors according to the web data information that made the existing data information unerring efficiently to ensure that information more useable and workable for the users effectively[2]. The other proposed technique related to detect the inconsistencies of the data information from any databases efficiently to remove the entities appeared more than one and update the existing database keeping that entity only once in a database more accurately to ensure an effective and useful database to the users[14]. A paper previously discussed represented the correction of the inconsistencies based on various algorithms and strategies by giving an ideal solution to correct the inconsistencies of information in a database ensuring the using database a consistence one[7].

Chapter 4

Proposed Technique

In this section, first we have discussed about "Apriori Algorithm" under association rule and other relative matters pre-requisite for applying this algorithm and then other algorithms to this research procedure. Initially we are detecting data currency with the timestamp and later we will try to use machine learning for data currency, data inaccuracy detection, information from www and find the actual data can be hard as there are many data with the related term but actual information extract from this can be hard and there will some unnecessary information which can affect the result.

4.1 Proposed Model

In our proposed technique, we first start with inconsistent data detection and correction. For this purpose we use modified apriori algorithm which is distance and dependency based. From association rule analysis, we first generate some strong and related rules by using frequent itemsets. Basically apriori algorithm is a rule based technique and used for genetically related data. In our day to day data there is also a relation which we can compare with genetic information. Suppose we can say, from our morning we first complete our breakfast then we start for our office/school and then other works. There is a genetically related sequence in our daily life activity. From this concept, we choose apriori algorithm to find a relationship in our data for finding inconsistent data.

In our proposed model, our first module is inconsistency detection module where first we have a sub module Rule Generation module. In Rule generation module first we take input from the database and then we generate frequent itemset. From this itemset we then generate rules. Rules are the final output of rule generation sub module. In our Inconsistency detection sub-module we take an input from the database and use the generated rules in rule generation sub-module. In this section, we generate frequency of every possible data which indicates how frequent a dataset in a database system. Finally we get an output of new database with frequency.

In our second module which is Inconsistency correction module. Here we first extract the low frequent data from the frequency added database and generate a new database which is a low frequency datasets. From this low frequent database,

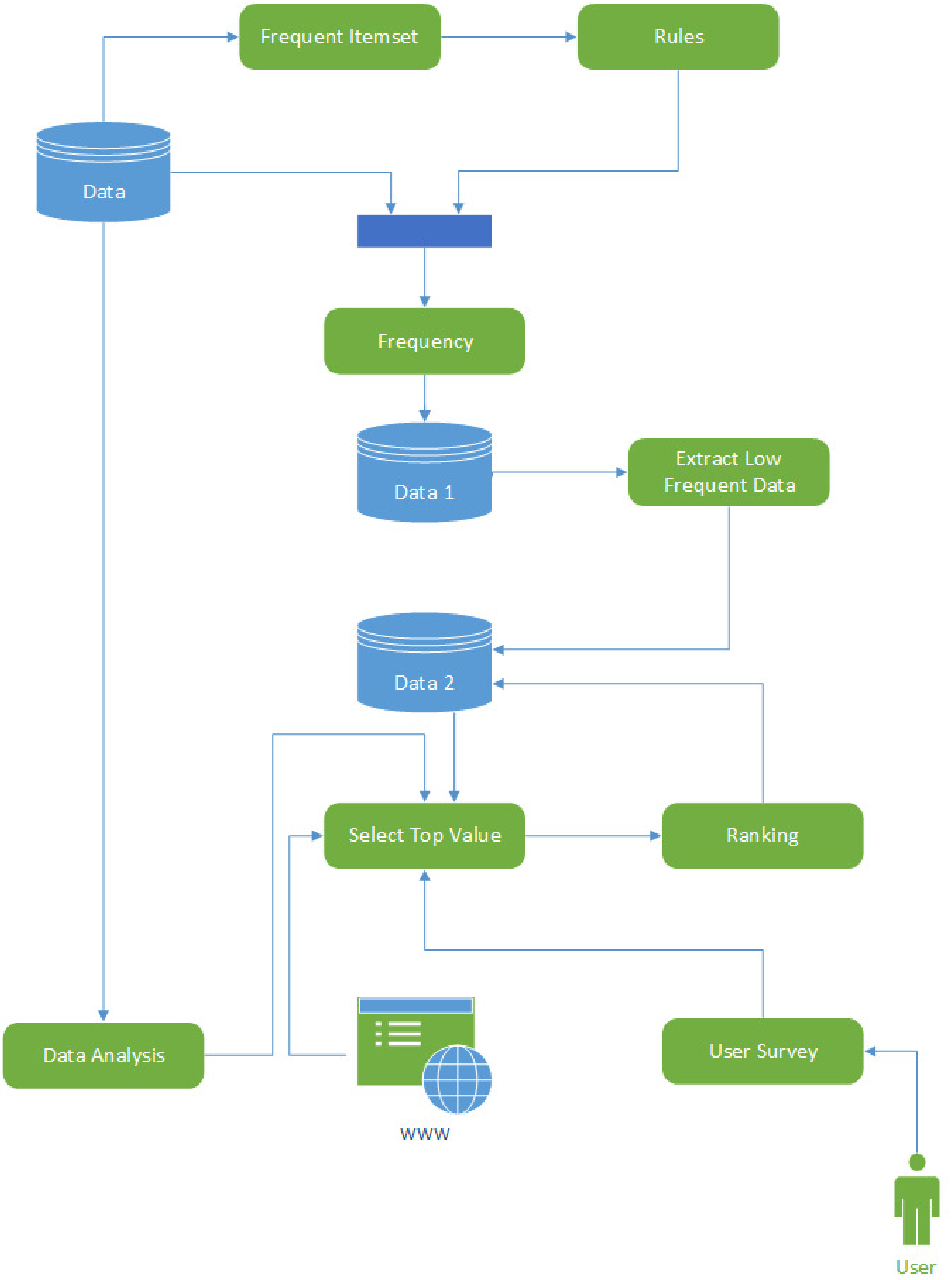


Figure 4.1: Basic Model

we first select the top value and then rank that value. We select the top value by following different three criterions. First one is data analysis phase. In data analysis phase, with our pre-generated rule, it selects a high frequent value according to the rule and returns the result. In www phase, we use the World Wide Web (www) for most correct result. It search in the web for a related term result and get beck the retrieved data. And finally the user survey phase. In user survey phase, we take a user interaction with the system. A user is asked for the most accurate answer for particular information. From these three phases, we will rank the value and the top ranked value will be pushed in our Low frequent database.

4.2 Distance

Suppose if we want to calculate support count for Table 2.6, then we have a big dataset like for ln column there will be individual column for every ln. If we want to calculate support count for column cc and ac then we can represent the table like this

ID	Itemset {cc,ac}
1	{44,131}
2	{44,131}
3	{44,131}
4	{01,243}
5	{01,243}
6	{01,243}
7	{10,120}

Table 4.1: Itemset

Here we can see {44,131}, {01,243} and {10,120} is an individual itemset where every itemset represent {cc, ac} and in every itemset individual is an item. Now if we want to calculate support count for {44,131} then we can find that three id contains this items. So support count for {44,131} is 3. But This type of representation like our database is costly and impractical for our domain. But according to our approach, we have to calculate support count for applying apriori algorithm. For applying apriori algorithm we have proposed a new idea which will substitute Support Count. With this new idea, we can easily apply apriori algorithm for our domain. We calculate Distance for every column where support count calculates for every individual itemset or items. Distance will return a similarity based distance that is connected with another column. If the distance is less than a minimum value then they are connected with another less distance column which returns a dependency. Let I={fn,ln,cc,ac,phn,street,city,salary,status} be the itemset for our dataset Table 2.6 and every row represent the individual record of individual person. From this itemset we will find a strong relationship among the itemset by calculating the Distance among them. Distance is referred to the number of distinct value for an individual column. If there are more frequent items then the Distance for this column will be less. For example, if we take ln, its distance is D(ln) is equal to 7 as there are seven different value and if we take cc its distance D(cc) is equal to 3 as there are three different cc value in the dataset. For theses table ln is more distance from cc. If we take ac, then D(ac) will be 3 which is in equivalent to cc. So we can say cc and ac is strongly connected with each other. For example, if we have a {ac} = 243 then we can say the cc should be 44. Or if we have {cc}=44, then we can say the ac should be 243, as there can be more value of ac for {cc} = 44, but this is not our concern right this time. Our main concern is we want to generate some rules.

4.3 Dependency

Dependency means the probability an itemset depends on another itemset. It's a probability calculation which gives us a concept whether a rule should be taken or not. As there will be generated an exponential wise itemset, this will be very much expensive if we use all the itemset. For this reason we take only that itemsets that is more frequent by calculating the distance. With this we will get some frequent itemsets. Using this itemsets we will generate some rules. There will be exponential wise rules from our frequent itemset. All of this rules are not same important. There are some rules that are less important or there are some rules which are more important. If we consider less important rules, that will increase our computational cost. So, we have to prune some rules which are less important. For this purpose we have introduced a mathematically probabilistic function which gives us probability for a rule which tells us a rule is more important or less. We compute Dependency for every generated rule. According to this Dependency we can prune some less important rules. If we want to define Dependency mathematically-

$$dependency, D = \frac{distance(subset\, of\, left\, side\, of\, the\, generated\, rule)}{distance(itemset\, of\, the\, generated\, rule)} \tag{4.1}$$

now, the distance of itemset means, if we take an itemset for rule generation, then the entire distance will be calculated for that itemset. We will take the pre generated distance for that itemset. Suppose, if our itemset is {A,B,C}, then first we will calculate the distance for that itemset, not individually for {A} or {B} or {C} but we will calculate the distance for {A,B,C}. The distance for subset of left side is, from the itemset {A,B,C} we can generate a rule like $\{A, B\} \rightarrow \{C\}$, which is if we know A and B then we can find the correct value of C. Here Left side itemset is {A,B}. Now we calculate the distance for {A,B} itemset. Form this two itemset, we can calculate the dependency of this rule which gives probabilistic value from [0-1]. If the probability is enough good then we can say than {C} is more dependent to {A,B}. From this dependency, we can find some rules are more dependent and some are less. According to dependency value, we will take that rules which is more dependent and exclude that rules that are less dependent. This will reduce our computational cost and gives us more accurate result.

4.4 Association Rule Discovery

Given a set of transection T, find all rules having $distance \leq mindis$ and $dependency \geq mindep$, where mindis and mindep is the corresponding distance and dependency threshold.

A brute-force approach for association mining is to compute distance and dependency for every possible rule. This approach is expensive because there are exponentially many rules that can be extracted from a given dataset. More specifically, if there is a dataset that contains d itemset, then we can express this [11]-

$$R = 3^d - 2^{d+1} + 1 \tag{4.2}$$

For example, if there is an itemset {A,B,C} then we can find some subset of this itemset like- {A}, {B},{C}, {A,B},{A,C},{B,C}, {A,B,C}, then we can find that there are 7 itemset. If we compute how many rules there will generated

$$R = 3^7 - 2^{7+1} + 1$$

$$= 1932$$

There is huge amount of rule generated from this 7 itemset. This huge amount rule will increase our computational complexity and cost. Among this rules, all of them are not equally important. So we have to deduce rules that are not important. From this huge amount of rule we will keep only strong rule by calculating distance and dependency. More than 80% of rules will be discarded after applying mindis and mindep threshold value. We can set the mindis threshold value. But sometimes it is not so effective. So, we set mindis by calculating a mean value of distance and taking the floor value of mean. Mathematically we can represent this-

$$mindis = floor[average\ distance] \tag{4.3}$$

4.5 Frequent Itemset Generation

Whose objective is to find the itemset that satisfy the mindis threshold value and this itemset are called frequent itemset.

A lattice structure can used to enumerate the list of all possible itemset. Figure 4.2 shows a lattice itemset I={A,B,C,D}. a dataset that contains k items can generate $2^k - 1$ itemsets excluding the null set. Practically k can be very large and it will generate exponentially very large itemsets and the search space will become large which is also exponentially large.

A brute-force approach for finding frequent itemsets is to calculate the distance for every individual candidate itemsets. Calculating the distance for every individual candidate itemsets, we can reduce some itemsets from our dataset.

For example, we can consider distance using *eq.* 4.3

$$D(A) = 7,\ D(B) = 3,\ D(C) = 2,\ D(D) = 4$$

$$mindis = floor[(7 + 3 + 2 + 4)/4] = 4$$

Now according to our approach, if distance$\leq$4 that itemset will be accepted otherwise it will be discarded. If we apply this then {A} item will be discarded and the remaining itemset is {B,C,D} as it satisfies the threshold value and {B,C,D} is our frequent itemset. There are more frequent itemsets which is shown in figure 4.3

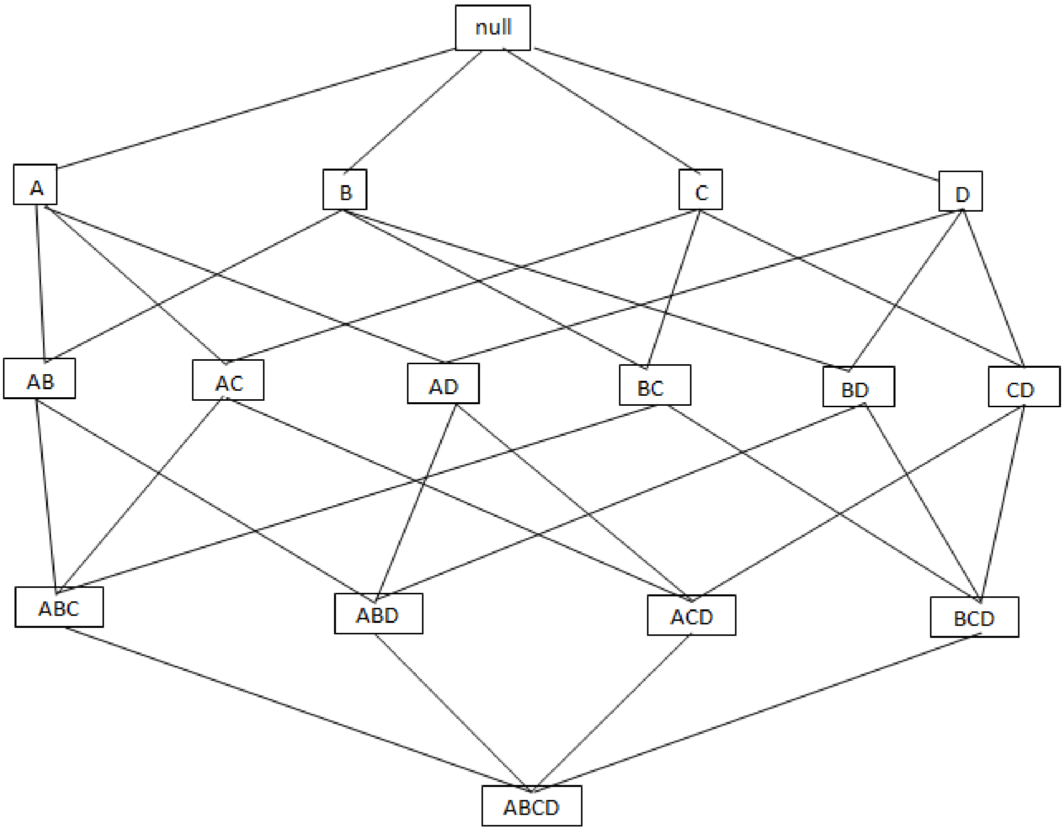


Figure 4.2: Itemset

Figure 4.4 shows the infrequent itemset which should not appear in our itemset generation because this will make our search space exponentially large which is very expensive. We eliminate this infrequent itemset by level wise computation by calculating distance value for each individual itemset. The more enhanced form is that mindis threshold value is not fixed rather it is changing with the corresponding itemset distance and discard some item every level in its frequent itemset generation.

From Figure 4.5, we can see that in step 1 mindis = 4 and we discard some item. In step 2 our mindis = 3 and now we also discard some itemse. Within this approach we can reduce the number of itemset by keeping the frequent itemset every step of our frequent itemset generation.

4.6 Rule Generation

Whose objective is to find the strong rule from the previous frequent itemset which satisfy the mindep threshold value. This section describes how to extract strong rules from frequent itemsets generated in previous section. As there are many generated rules which are not necessary though they are generated from frequent itemset. It will generate rule exponentially which will cause a big search space which is very expensive. So we should prune the unnecessary rules.

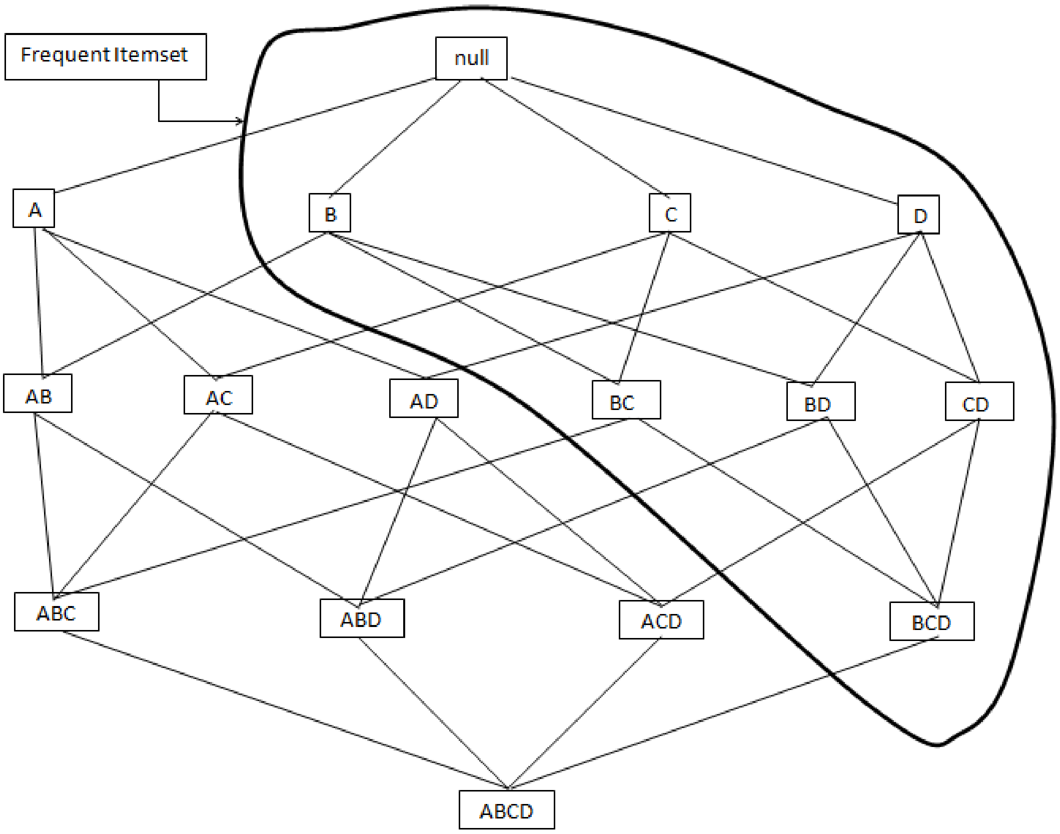


Figure 4.3: Frequent Itemset

For example, if we take I={B,C,D} frequent itemset. There exists six candidate association rule that can be generated from I, $\{B, C\} \rightarrow \{D\}, \{B, D\} \rightarrow \{C\}, \{C, D\} \rightarrow \{B\}, \{B\} \rightarrow \{C, D\}, \{C\} \rightarrow \{B, D\}, \{D\} \rightarrow \{B, C\}$. We are not just taking all the rules generated from frequent itemset I, rather than we will compute dependency for each rule and compare with the mindep. If it satisfies the threshold value mindep, then we will accept the rule otherwise it will be discarded. We will set the mindep threshold value manually. For example, we can set mindep as 80% which means that if the dependency is greater or equal to 80% then we will accept the rule otherwise it will be considered as less dependent rule and it will be discarded.

For example, if we take $\{B, C\} \rightarrow \{D\}$ then we first calculate its dependency using *eq.* 4.1

$$dependency = \frac{distance(B, C)}{distance(B, C, D)}$$
$$= 3/4$$
$$= 0.75$$
$$= 75\%$$

So, its dependency is 75%. With this formula we can calculate dependency and extract the strong rules. If we take $\{C, D\} \rightarrow \{B\}$ then we first calculate depen-

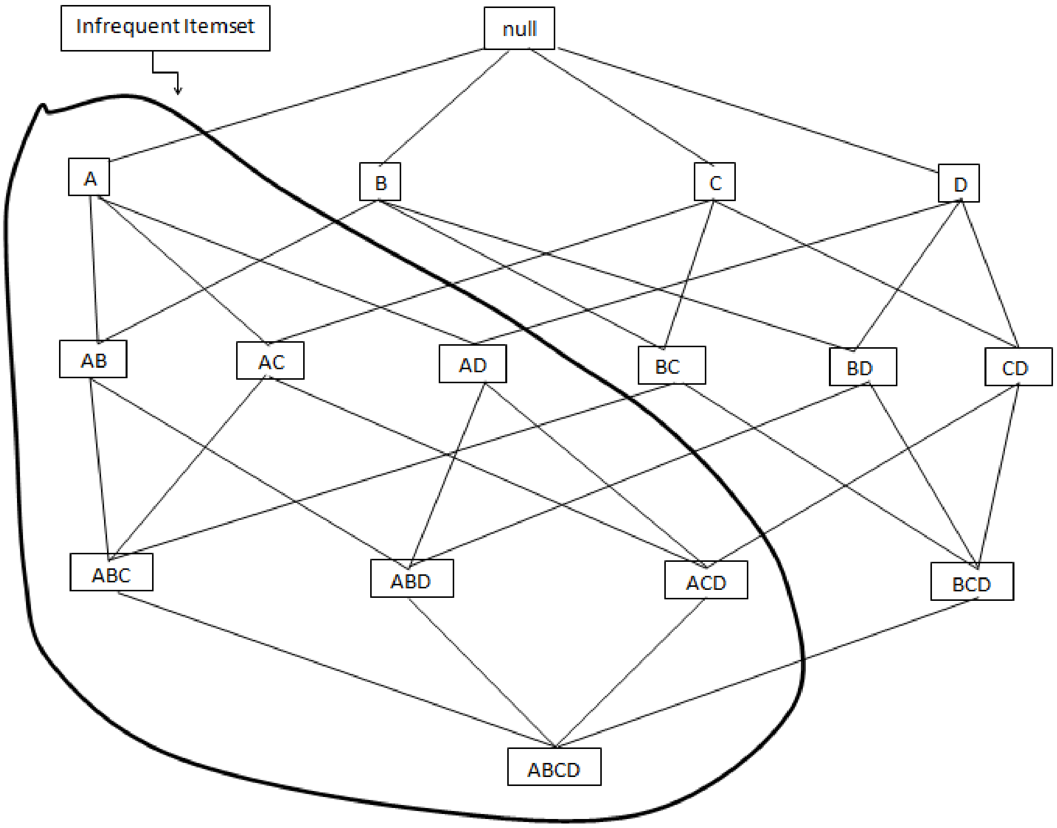
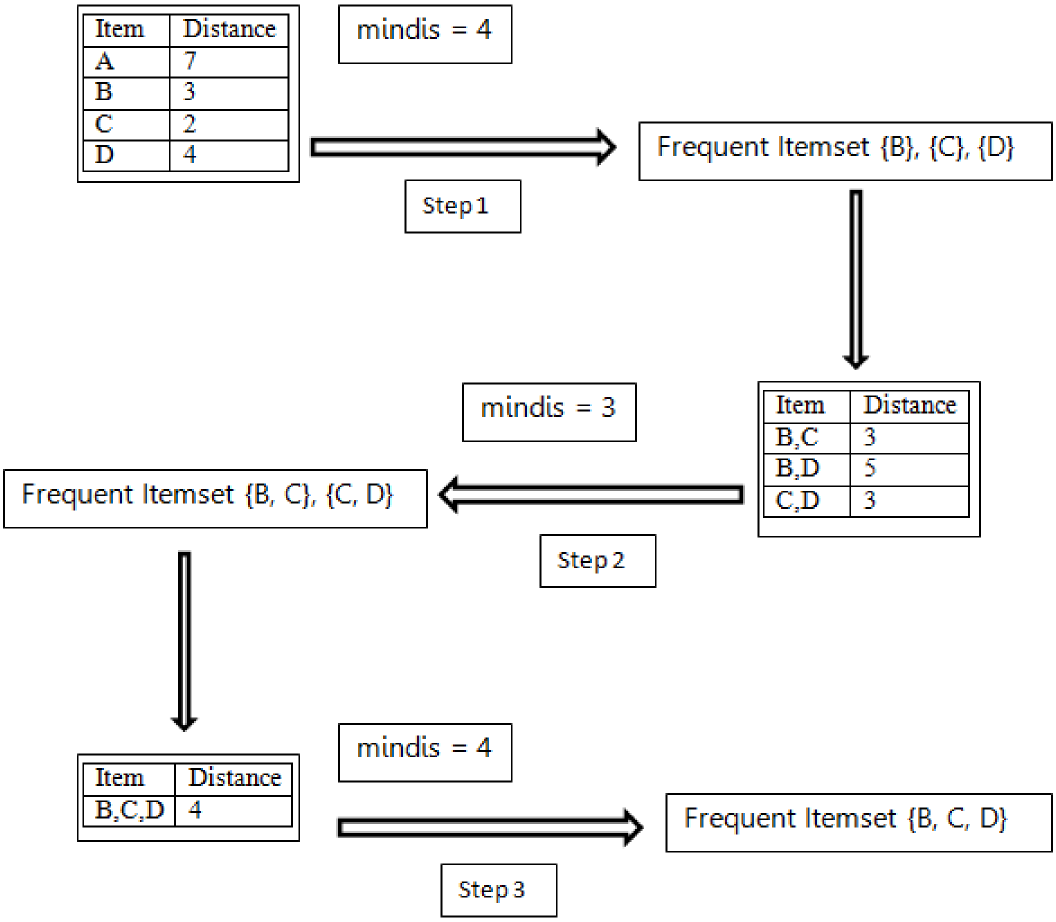


Figure 4.4: Infrequent Itemset

dency using *eq.* 4.1

$$
\begin{aligned}
Dependency &= \frac{distance(C, D)}{distance(B, C, D)} \\
&= 3/4 \\
&= 0.75 \\
&= 75\%
\end{aligned}
$$

So, we can find dependency for $\{C, D\} \rightarrow \{B\}$ is 75%, here 75% means this rule is more dependable. In practical case we can also find dependency for rule pruning. If we consider a real database then we can generate huge amount of rules. But all of them are not equally dependent. This huge amount of rule will increase our cost. Sometimes, these rules will not give us the exact information. So, for these reasons we should consider such rules which are more accurate by calculating the dependency. From figure 4.6 we can see there are some marked rules which are more dependable. The unmarked rules are unnecessary for search space and for analysis efficiency this rules should be extracted otherwise it will grow more expensive.



Item	Distance
A	7
B	3
C	2
D	4

Item	Distance
B,C	3
B,D	5
C,D	3

Item	Distance
B,C,D	4

mindis = 4
Step 1
Frequent Itemset {B}, {C}, {D}
mindis = 3
Frequent Itemset {B, C}, {C, D}
Step 2
mindis = 4
Frequent Itemset {B, C, D}
Step 3

Figure 4.5: Frequent Itemset with Distance

4.7 Inconsistency Detection and Correction

This section will describe how we can detect data inconsistency from a given dataset. If we apply modified association analysis, then we can discover a hidden relationship from our dataset which is associated with consistency.

First we will apply a pre generated rule from the previous section. According to the rule we can fetch data and compute the frequency of that particular data. Here frequency referred as the number of times a data appears in a dataset. By calculating the frequency, we can assume that a record is consistent or not. Less the frequency, more the probability of being the data inconsistent.

For example, if we consider Table 4.2, we can see an extra column frequency. Consider row 4, 5, 6.if we take a pre generated rule $\{city\} \rightarrow \{cc\}$, then we can observe frequency is 1, 2, 2 accordingly. Here we took city value as 'FT' which refer to cc=01, but in row 4 cc is 20 which is inconsistent with other data in dataset. More examples is shown from dataset-

If we consider our data set, then we can find that almost all the frequency is 7 except one of the data is set frequency 1. We compute this frequency by applying our pre generated rule which is {ac}→{city} which means that if we know the

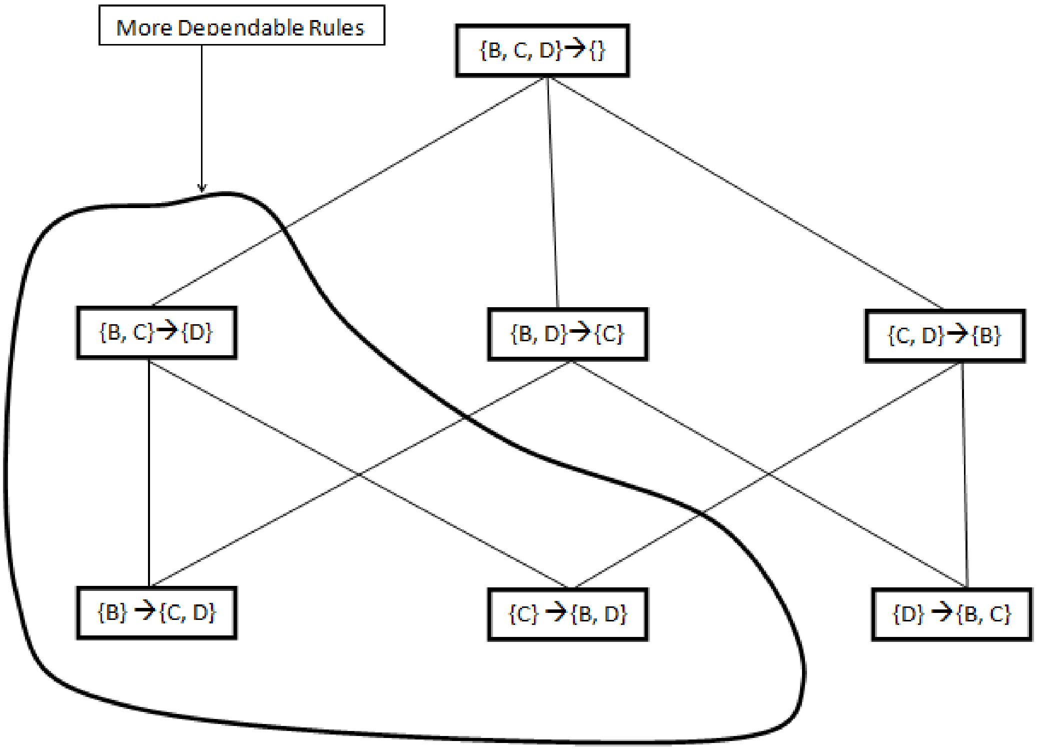
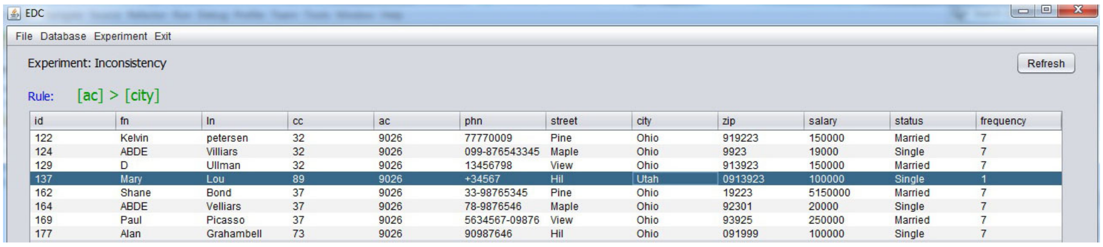


Figure 4.6: Dependable Rules

fn	ln	cc	ac	phn	street	city	zip	salary	status	frequency
Mike	Col	44	131	4459	New Str.	DH	Z2	60k	single	3
Jim	Clark	44	131	4467	Old Str.	DH	Z2	60k	married	3
Mic	Colli	44	131	4478	New Str.	DH	Z2	60k	married	3
Joi	Tim	10	243	23678	Mtv	FT	A67	60k	single	1
Bob	Ric	01	243	23964	Mtv	FT	A67	60k	married	2
Rik	Jov	01	243	23084	Mtv	FT	A67	60k	single	2
Jovn	Jony	10	120	12567	Oris	HGT	67I	60k	married	1

Table 4.2: Employee Details with Frequency



Experiment: Inconsistency

Rule: [ac] > [city]

id	fn	ln	cc	ac	phn	street	city	zip	salary	status	frequency
122	Kelvin	petersen	32	9026	77770009	Pine	Ohio	919223	150000	Married	7
124	ABDE	Villiars	32	9026	099-876543345	Maple	Ohio	9923	19000	Single	7
129	D	Ullman	32	9026	13456798	View	Ohio	913923	150000	Married	7
137	Mary	Lou	89	9026	+34567	Hil	Utah	0913923	100000	Single	1
162	Shane	Bond	37	9026	33-98765345	Pine	Ohio	19223	5150000	Married	7
164	ABDE	Velliars	37	9026	78-9876546	Maple	Ohio	92301	20000	Single	7
169	Paul	Picasso	37	9026	5634567-09876	View	Ohio	93925	250000	Married	7
177	Alan	Grahambell	73	9026	90987646	Hil	Ohio	091999	100000	Single	7

Figure 4.7: Employee Details(Implementation)

ac then we can tell a city is in correct value or not. Obviously this is a strong rule by computing distance and dependency and also there is a strong connection between them. Now, if the {ac}=9026 then we can find {city} is {Ohio, Utah}. But from our database, {Ohio} is the top frequent value where the frequency is 7 but {Utah} holds only one value. From this analysis, we can find an inconsistent with the other data in our data set. So, by calculating this frequency we can find some inconsistent data in accordance with our other data in the data set.

Previously we have generated rules from frequency .This table is generated from the rule $\{age \rightarrow status\}$ Here sum of frequencies are

$$6 + 6 + 3 + 3 + 3 + 6 + 6 + 6 + 6 = 45$$

$$Average frequency, avgFrq = \frac{45}{9}$$

$$= 5$$

We considered average frequency as lower limit of frequency. If frequency of any tuple is less than this lower limit, we can suggest them as error data. To check as they are exactly error or not, we extract that or those tuples from the data table for further operations.

name	age	status	frequency
A	45	Married	6
B	30	Married	6
C	28	Single	3
D	29	Single	3
E	35	Single	3
F	37	Married	6
G	30	Married	6
H	32	Married	6
I	55	Married	6

Table 4.3: Example of Frequency

From the table 4.3 we can see, tuple 3, 4, 5(C, D, E) have frequency value 3, which is less than lower limit of frequency. So, we can suggest them as error data, not surely as an error. So, we will extract these tuples for detecting and correcting the errors.

From some observations of data tables from the dataset, we came to a decision that, those who are below 30 years old, are usually single. Besides who are 30 years old or above, are considered as married. In some cases we see different results. But they are exceptional cases.

name	age	status	frequency
C	28	Single	3
D	29	Single	3
E	35	Single	3

Table 4.4: Low Frequent Data

Here, This table 4.4 is extracted from the actual data table 4.3 as all they have frequency 3. But as per our decision, C and D are correctly classified in same status. On the other hand we suggest that, E is not correctly classified in appropriate status. As C and D are correctly classified, we will delete them from error table 4.4. But E will remain in error table. It will stay in error table and further we will notify it as a suggestion.

Case 2: In this example, this table is generated from the rule $\{cc, ac\} \rightarrow \{zip\}$ as the rule is generated from frequency.

name	cc	ac	zip	frequency
A	1	101	1111	4
B	2	201	2112	1
C	1	101	1111	4
D	2	201	2111	2
E	3	331	3333	2
F	1	101	1111	4
G	2	201	2111	2
H	1	101	1111	4
I	3	331	3333	2
J	1	101	1112	1

Table 4.5: Another Example of Frequency

Here, Sum of frequencies

$$4 + 1 + 4 + 2 + 2 + 4 + 2 + 4 + 2 + 1 = 26$$

$$AverageFrequency, avgFrq = \frac{26}{10}$$

$$= 2.60$$

So, the lower limit of frequency is 2.60. The tuples which are lying down the lower limit of frequency, we will initially treat them as assumed error data. Then we will extract from the data table and will generate an error table. In our data table 4.5, tuple 2,4,5,7,9,10(B,D,E,G,I,J) has frequency lower than the lower limit of frequency. So, we can assume them as error data and then we will extract them from data table and generate error table for further comparison and operation.

name	cc	ac	zip	frequency
B	2	201	2112	1
D	2	201	2111	2
E	3	331	3333	2
G	2	201	2111	2
I	3	331	3333	2
J	1	101	1112	1

Table 4.6: Another Low Frequent Data

Now, we will check as they are error data or not. So, at first we will compare this error tuples with the top ranked values of the actual table. For B, comparing with cc and ac of actual table, the top ranked value of zip is 2111. But here ZIP is 2112. So, we can decide that this zip code is wrong. It will remain in the table as a suggestion for it's being wrong. Top rank value: according to the rule every specific tuple having the same attribute, have the same frequency. This frequency value of that tuple for the rule is called top ranked value. For example, From the actual data table 4.5, for the same cc(2) and ac(201) we got zip(2111) for 2 times. This is the highest value for this same cc and ac. So, it is the top ranked value.

For D and G, Comparing with cc and ac of actual table, top ranked value of zip is 2111. Here from the extracted table zip is also 2111.

name	cc	ac	zip	frequency
B	2	201	2112	1
E	3	331	3333	2
I	3	331	3333	2
J	1	101	1112	1

Table 4.7: Actual Low Frequent Data

So, for this low frequency below the lower limit of frequency, this is not error data. After checking this, we will delete these values from the extracted table. Then the table will be like this following

For E and I, Comparing with cc and ac of actual table, top ranked value of zip is 3333. Here from the extracted table zip is also 3333. So, for this low frequency below the lower limit of frequency, this is not error data. After checking this, we will delete these values from the extracted table. Then the table will be like this following

name	cc	ac	zip	frequency
B	2	201	2112	1
J	1	101	1112	1

Table 4.8: Inconsistant Data with Error

For J, comparing with cc and ac of actual table, the top ranked value of zip is 1111. But here zip is 1112. So, we can decide that this zip code is wrong. It will remain in the table as a suggestion for it's being wrong.

4.7.1 WWW Data Retrieval

WWW(World Wide Web) is a vast area of knowledge, information. We can get any kind of information from WWW by searching in the web[15]. For our inconsistency detection, we can search in the web with the related term. For example, we have a database which updates all the president list of the world. If we consider US President 2017, then our database return the name of Barak Husain Obama. But it is irrelevant information, because one person can be at most back to back two term president in us president law. With this constrain this is an inconsistent information with the US president database system and at the same time it is a wrong information. In another case, if the Obama is for only one term president then it will not show any error because one person can be the president at most two times. But it can also be wrong because he is not the current selected president, but in database analysis this will not detected. So, in these two cases we can search in the web for the actual information.

WWW search will return us huge informational page. But we don't need all this pages. Which pages should we consider? For this selection we used tf-idf which stands for term frequency and inverse document frequency.

TF: Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization [16]:

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \tag{4.4}$$

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

$$IDF(t) = log_e \frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it} \tag{4.5}$$

Example: Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12[16][17]

Tf-idf return us the top documents according to our search term. This tf-idf will reduce our computational cost and narrow our search space. We can select top 5/6 or other numbered document using tf-idf. From these top ranked documents, we will search for our related term. First we use string matching criterion. By matching with our related search term we extract the related sentence from the top ranked documents. By this we will get the related termed sentences which contains the information of our search term[18][19].

4.8 Proposed Algorithm

```
Result: result,R
for i=1 to itemset.size() do
    for every i items in itemset do
        generate i items subset, S;
        calculate the distance, D of subset, S;
        if $D \leq mindis$ then
            add subset, S to result, R;
        end
    end
end
```

Algorithm 1: Frequent Itemset Generation

Suppose there is a itemset {A,B,C} and its size is 3. For this we will generate different subset itemset. If we start with every i itemset, where i will be from 1 to itemset size which is here 3. If i=1, then for every 1 items in itemset, we will generat 1 item subset, which will be {A}, {B}, {C}. in every step we will calculate distance for every subset, suppose for {A}, if the distance for {A} is less than minimum distance then we will add A to the result otherwise we will discard that subset. If we consider i=2, then it will generate 2 item subset for every 2 items from itemset. This will generate {A,B}, {A,C}, {B,C} subsets.

```
Result: rule,{Rl}
for i=1 to itemset.size()-1 do
    for every i items in itemset do
        generate i items subset, S;
        add rest of the items in findset, F;
        calculate the dependency, Dp;
        if Dp≥mindep then
            add {(subset, S), (findset,F)} to rule, {Rl};
        end
    end
end
```

Algorithm 2: Rule Generation

If we have an resultant itemset like {B,C,D,E} then from this itemset we have to generate rules. We first start from i=1 and continue the process until size-1, for example here 1 to 3. For every i items in itemset we will generate I items subset. If i=1, then for every 1 item in itemset, we will generate 1 item subset and add the rest of the items in findset. For example, if we take {C} as subset and add the rest of the items in findset that is {B,D,E}. After that we will calculate the dependency for this temporary rule. If the dependency can fulfill the minimum dependency then we will add this as a strong rule in our rule, {Rl}, otherwise we will discard.

```
Result: frequency, F
select a rule, {Rl};
according to rule, {Rl};
      calculate the similar value count;
      assign this count as frequency, F for those data;
```

Algorithm 3: Frequency Calculation

Suppose we first select a rule $\{C\} \rightarrow \{D\}$. there are some values in C like {11, 11, 11, 11, 11, 22, 22, 22,$\cdots$} and there are some value in D like {33, 33, 33, 43, 43, 65, 65, 65,$\cdots$} accordingly they maintain their respective serial no. from the rule we can say if we know C then it will give us some D values. If {C}={11} then we can find {D}={33,43} we get two value for a single C value but their appearance is different, 33 appears 3 times and 43 appears 2 times. So we can say their frequency is 3 and 2 respectively.

```
Result: Inconsistent Data
select a rule, {Rl};
according to rule, {Rl};
      calculate the avgFrq from pre calculated frequency, F;
for every data do
    if frequency, F<avgFrq then
        extract data;
    end
end
```

Algorithm 4: Inconsistent Data Extraction

After calculating the frequency we have to calculate the average frequency for inconsistent data extraction. First we select a rule; basically it's a pre-selected rule which is used in previous section for frequency calculation. By comparing every frequency with the average frequency, if the frequency is less than the average frequency then we can say there are some inconsistent data and we will extract those data for further analysis.

```
Result: Top Value
select a rule, {Rl};
according to rule, {Rl};
      calculate the avgFrq from pre calculated frequency, F;
for every data do
    find the top frequent value;
    if frequency, F≤avgFrq or value is similar then
        mark as free;
    else
        select the top frequent value
    end
    or;
    set a range;
    if the value in range and return value is top frequent then
        mark as free;
    else
        if the value in range and return value is not top frequent then
            select the top frequent value;
        else
            mark as free;
        end
    end
end
```

Algorithm 5: Top Value Selection

Here we also use our pre-selected rule I previous section which is frequency calculation. According to our rule first we will find the top value. If the top value is equal to or near the low frequent value then we mark those data as not inconsistent. For example, we first select a rule {C}→{D}. there are some values in C like {11,11,11,11,11,22,22,22,. . . .} and there are some value in D like {33,33,33,43,43,65,65,65,. . . ..} accordingly, then we find for {C}={11}, there are two value for {D}={33,43}, but 43 is two times but 33 is three times. Now if we find the top frequent value that will return 33, but 43 is near about 33. So initially we say 43 is inconsistent but after further analysis 43 is almost near about 33, so we will mark this value as not inconsistent. But in case for big data like 33 is 30 times frequent and 43 is one or two time frequent the we can say that value is inconsistent. Or if we consider age with marital status then we can see some data like {age}={33,25,46,23,54,35,38,29,47,58,32,49} and {status}={m,s,m,s,m,s,m,s,m,m,m,m} where m = married , s= single. Then we can find m is the top value, s is less frequent value. Here {25,23,35,29} is single {33,46,54,38,47,58,32,49} is married. Now we cannot say single person should be married at their given age but one thing that we can say 35 should be married. For this type of data we can set a range for example, from 32 to 58 aged people should be married. We cannot mark this type of data as inconsistent with the top value.

4.9 Experiment

System Configuration

We have implemented most of the algorithm in Java and some rest of them in Python and have run our experiment on an Intel Core i5 CPU 1.60GHz, 4.00GB RAM, Windows 7 Professional 64bit.

Dataset

We have created our own dataset which contains fields named fn, ln, cc, ac, phn, street, city, zip, salary, status. Basically it represents a persons information including his address, salary, marital status.

Result

From our experiment, we have generated subset of itemset. From this subset we have generated some rules. we have selected a random rule. based on this rule, we select some particular value which gives us some return value which we want to analysis the returned result is consistent or not. Figure 4.7 is an experimental result. Here we select a rule and then we find the frequency, with this frequency we analysis the dataset for consistent data and fix the low frequent data.

Analysis

Data is a valuable thing. Data can be anything. It can be right or wrong. We experiment this with some generated rule. For this rules, some correct data may be affected. But this is not our concern, as we do not directly override the existing data rather than we suggest for the new selected data. So, this can be a hope full design for data correction.

4.10 Limitation

- As this is a rule generated system, so there are many rules that are mathematically strong but in real sense this type of rule is not relevant. For example, if there is a rule something like this {city}→{marital status} which is sometimes an irrelevant rule in real sense.
- Data is a big and expensive thing. Data can be anything. We cannot say a data is wrong or incorrect. Sometimes, in our proposed system, there might be some data is correct but for our system it is marked as incorrect. Suppose, if a person age is 45, then he should be married which is saying from our system. But a person can be single at the age of 45. This is an exceptional case. A person can also be Divorced which is also a less frequent data.

4.11 Summary

In our proposed technique mainly based on three module, Data Analysis, WWW(World Wide Web) and User Interaction. We use rule based apriori algorithm with the proposed modification, we call this 'modified apriori algorithm'. Our main approach is for error detection and correction for inconsistent data. Though there are some limitations, initially we are ignoring this.

Chapter 5

Conclusion

Web based error detection and correction process is composed of Inaccuracy detection and correction, duplication, incompleteness and data currency parts. In our ongoing work we were done with inaccuracy detection and correction part. We used distance and dependency based modified Apriori algorithm in lieu of confidence and support count based Apriori algorithm to generate frequent item sets. We have generated rules from frequent item sets and the rule generated data table with frequency. We have analyzed data through frequency value to detect errors according to the top frequency and then proposed WWW retrieval phase which is based on term frequency and inverse document frequency and User survey phase to correct the information. Moreover, our aim is to enhance data quality of database.

5.1 Contribution

First we apply modified apriori algorithm for frequent itemset and rule generation. The main apriori algorithm use support count and confidence based pruning. As our research work and dataset type is different from genetic or transactions based work, so the existing support count and confidence is not suitable for our dataset. So, we thought these two point differently which is distance and dependency. By using these two terms we got a preliminary result which is satisfying our research work though there are some limitations between these two terms. Initially we can ignore this limitation. Then we have generate frequency by which we can approach in our correction. We suggest 3 way to correction process, data analysis, www and user and then we generate a rank and then suggest the top ranked value. As data can be any thing, any type, so we suggest the value rather directly correcting them.

5.2 Future Work

Out of datedness is one of the most interesting parts of our research work. Machine learning is one of the most useful techniques for out of datedness detection. Our next step will be go through with machine learning and find an efficient way for out of datedness detection. Web scraping is another interesting part. Web scraping go through machine learning, artificial intelligence, computer vision, natural language processing. We intend to go with natural language processing for web scraping and find an effective way for required data fetching from WWW. WWW is a vast thing like a sea, so we want to fetch our exactly needed information and store them in a database for improving data quality. For this purpose we will extract top ranked documents from the web. From this top ranked documents we will use string matching criterion to find out the related sentence which will be used to detect out of datedness of information in the database.

Bibliography

[1] S. Ma N. Tang W. F. Fan, J. Z. Li and W. Y. Yu. Towards certain fixes with editing rules and master data in Proceedings of the VLDB Endowment. 3(1-2), 2010.

[2] Cheqing Jin Hailong Liu, Zhanhuai Li and Qun Chen. Web-based Techniques for Automatically Detecting and Correcting Information Errors in a Database. *2016 IEEE, BigComp*, 2016.

[3] L. Sitbon S. Sadiq M. Indulska Z. X. Li, M. A. Sharaf and X. F. Zhou. "Webput: efficient web-based data imputation in Web Information Systems Engineering. *Springer*, 2012.

[4] M. Magnani. Techniques for dealing with missing data in knowledge discovery tasks. 15(1), 2004.

[5] W. F. Fan and F. Geerts. Foundations of data quality management in Synthesis Lectures on Data Management. 4(5), 2012.

[6] Wikipedia. Web scraping.

[7] XIAO-HUI XU YI ZHANG JUN-QING CHEN XIN WANG, LIN-PENG HUANG. A Solution for Data Inconsistency in Data Integration. *JISE*, 2011.

[8] J. Neville M. Ouzzani M. Yakout, A. K. Elmagarmid and I. F. Ilyas. Guided data repair in Proceedings of the VLDB Endowment. 4(5), 2011.

[9] C. J. Zhang Y. Li Y. Tong, C. C. Cao and L. Chen. Crowd-cleaner: Data cleaning for multi-version data on the web via crowdsourcing in ICDE. *IEEE*, 2014.

[10] W. Y. Meng X. Li and C. Yu. T-verifier: Verifying truthfulness of fact statements in ICDE. *IEEE*, 2011.

[11] www.columbia.edu/ jwp2128/Teaching/W4721/papers/ch6.pdf. Association analysis: Basic concept and algorithms.

[12] Sanjeev Rao Robert Baumgartner Jyoti Arora, Nidhi Bhalla. A Review on Association Rule Mining Algorithms. *IJIRCCE*, 1, 2013.

[13] Wikipedia. Association rule learning.

[14] Weining Zhang Ke Wang. Detecting Data Inconsistency for Multidatabases.

[15] Pasquale De Meo Giacomo Fiumara Robert Baumgartner Emilio Ferrara, . Web Data Extraction, Applications and Techniques: A Survey.

[16] http://www.tfidf.com/. Tf-idf :: A single-page tutorial - information retrieval and text mining.

[17] Steven Loria. Finding important words in text using tf-idf.

[18] Son Doan Kevin B Johnson Lemuel R Waitman Joshua C Denny Hua Xu, Shane P Stenner. MedEx: a medication information extraction system for clinical narratives. 17(1), 2010.

[19] Ewan Klein Steven Bird and Edward Loper. Natural language processing with python.

Appendix

The code is available to download freely from the website:
https://github.com/shacho0011/ErrorDetectAndCorrect