DETECTING DISGUISED MISSING DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

RAHİME BELEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

FEBRUARY 2009

Approval of the Graduate School of Informatics

_____

Prof. Dr. Nazife Baykal

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Yasemin Yardımcı

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assist. Prof. Dr. Tuğba Taşkaya Temizel

Supervisor

Examining Committee Members

Assoc. Prof. Ferda Nur Alpaslan (METU, CENG) _____

Assist. Prof. Dr. Tuğba Taşkaya Temizel (METU, II) _____

Assist. Prof. Dr. Erhan Eren (METU, II) _____

Assist. Prof. Dr. Altan Koçyiğit (METU, II) _____

Assist. Prof. Dr. Pınar Şenkul (METU, CENG) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name: Rahime Belen

Signature : _____

ABSTRACT

DETECTING DISGUISED MISSING DATA

BELEN, Rahime
M.S., Department of Information Systems
Supervisor: Assist. Prof. Dr. Tuğba T. Temizel

February 2009, 59 pages

In some applications, explicit codes are provided for missing data such as NA (not available) however many applications do not provide such explicit codes and valid or invalid data codes are recorded as legitimate data values. Such missing values are known as disguised missing data.

Disguised missing data may affect the quality of data analysis negatively, for example the results of discovered association rules in KDD-Cup-98 data sets have clearly shown the need of applying data quality management prior to analysis.

In this thesis, to tackle the problem of disguised missing data, we analyzed embedded unbiased sample heuristic (EUSH), demonstrated the methods drawbacks and proposed a new methodology based on Chi Square Two Sample Test. The proposed method does not require any domain background knowledge and compares favorably with EUSH.

Keywords: Data Quality, Data Cleaning, Disguised Missing Data, Chi Square Two Sample Test

ÖZ

GİZLİ KAYIP VERİLERİN BULUNMASI

BELEN, Rahime
Yüksek Lisans, Bilişim Sistemleri
Tez Yöneticisi: Yrd. Doç. Dr. Tuğba T. Temizel

Şubat 2009, 59 sayfa

Bazı uygulamalarda kayıp veriler NA gibi özel kodlarla belirgin bir biçimde ifade edilirken, bir çok uygulamada veri aslında kayıpken veri tabanına geçerli ya da geçersiz veriler olarak kaydedilir. Bu tür kayıp verilere gizli kayıp veri denilir.

Gizli kayıp veriler veri analizinin kalitesini etkiler. Örnegin, KDD-Cup-98'de kullanılan verilerde bulunan birliktelik kurallarında analiz öncesi veri kalitesi yönetim uygulaması ihtiyacı açıkca gösterilmiştir.

Bu tezde, gizli kayıp veri sorununu çözmek için gömülü yansız örneklem buluşsali (YÖB) incelenmiş, kusurları gösterilmiş ve Ki-kare iki örneklem testi üzerine kurulu yeni bir yöntem önerilmiştir. Bu yöntem hiç bir alan bilgisine ihtiyaç duymamaktadır ve YÖB'den daha iyi performans göstermektedir.

Anahtar Kelimeler: Veri Kalitesi, Veri Temizleme, Gizli Kayıp Veri, Ki-kare İki Örneklem Testi

This thesis is dedicated to:

My grandmother, Sıdıka Koçak

ACKNOWLEDGMENTS

I would like to express my gratitude to everyone who gave me the possibility to complete this thesis. I am deeply indebted to my supervisor Assist. Prof.Dr. Tuğba Taşkaya Temizel whose help, stimulating suggestions and encouragement helped me in all the time during my research.

I want to thank all my colleagues and institute personnel for all their help, support, interest and valuable hints. Thanks go to The Scientific and Technical Research Council of Turkey (TÜBİTAK) for providing scholarship during my study.

Especially, I would like to give my special thanks to my parents, sisters and friends whose love enabled me to complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

MEUS	: Maximal Embedded Unbiased Sample
CBSQS	: Correlation Based Sample Quality Score
VCS	: Value Couple Score
EUS	: Embedded Unbiased Sample
EUSH	: Embedded Unbiased Sample Heuristic

CHAPTER 1

INTRODUCTION

Act like the sun in love and compassion!
Act like a river in friendship and fraternity!
Act like the night in covering the faults of others!
Act like the soil in humility and selflessness!
Act like a dead one in anger and fury!
Act in accordance with the way you look!
Look in accordance with the way you act!
Mawlānā Jalāl ad-Dīn Rumi

Rumi, who is one of the great spiritual masters, poetical geniuses of mankind and founder of the Mevlevi Sufi order, explained the quality of a "*human being*" as acting accordance with the way he looks and looking in accordance with the way he acts. After approximately 8 centuries from his life, in 21th century, we can explain our expectations from "*data*" as a means of quality in the same way. But reality is disappointing for both Rumi and data miners.

In many applications such as filling in a customer information form on the web, some missing values are not explicitly represented as such, but instead appear as potentially valid data values. Such missing values are known as *disguised missing data*. Because they appear as valid data values they insidiously impair quality of data analysis severely. They may cause significant biases in data analysis and cause misleading results in hypothesis tests, correlation analysis and regressions.

For example in an online application form, say a value *male* for attribute *gender*, may be set as default value. Many female customers may not want to disclose their privacy or just do not want to spend time. Consequently missing values may disguise themselves as the default value *male*.

In this thesis, we focused on a heuristic approach proposed by Ming Hua and Jian Pei in their paper [1]. We implemented their approach, analyzed the deficiencies and proposed an improvement to overcome deficiencies.

In their approach, they generate a framework for detecting disguise values based on two assumptions. Firstly, they make an assumption that a small number of values, typically one or two in an attribute, are frequently used as disguises (a value set as default in an application, first value in a drop down list or most common answer known to public). Secondly they propose that disguised missing entries are often distributed randomly.

So they formulize the definition of disguise missing values as one or two values of an attribute which are distributed randomly. This definition leads such a result that, in a dataset *D*, when the attribute *A* includes a disguise value *v*, the subset of *D* in which the all tuples have the value *v* on attribute *A* should contain a large unbiased sample of *D*. They call this definition as *The Embedded Unbiased Sample Heuristic* and generate the framework. In the framework the value *v* in an attribute which has the maximal unbiased sample is computed and assigned as disguise value.

This heuristic includes two technical challenges; how to measure whether a set of tuples is an unbiased sample of a dataset and how to compute maximal unbiased sample. They use correlations for the first challenge. If values correlated in dataset *D*, are also correlated in a subset, they assign the subset as unbiased sample. This method is called *correlation- based sample quality score* and based on joint probability and correlation difference of value couples. For the second challenge they propose a greedy method to compute approximate maximal embedded unbiased sample.

In our thesis, we observed some deficiencies in *correlation- based sample quality score* and displayed our arguments with experiment results.

Secondly we demonstrated our methodology based on *chi-square two sample tests* to compute unbiased sample and demonstrated its advantages with some experiments. Our methodology computationally performs better than *correlation- based sample quality score.*

Rest of the thesis includes four chapters. In chapter two, we explain data quality, metrics of data quality, effects of poor data quality and current data quality assessments. In chapter three, we explain disguised missing data and described the framework proposed by Ming Hua and Jian Pei. We argued the framework and demonstrated our arguments with experiments. Secondly we introduced our methodology and demonstrated our improvements. Finally in chapter four, we summarize what we have covered in this thesis. We give information about what this thesis contributes to the literature. Finally, we conclude with a discussion about possible future work in this field.

CHAPTER 2

LITERATURE REVIEW

" *I'm sorry Mr. Smith, but according to our records, you're dead."*
A customer service
representative, with any
organization, any day.

2.1 What is Data Quality and Why Care about It?

While there are many definitions of data quality, the following definition is used frequently: A collection of data X is of higher quality than a collection of data Y if X meets customer needs better than Y [2]. In another words, we can sum it up as data quality is ***the fitness of use*** which implies that data quality is inherently subjective.

Every day important decisions are made based on the data stored within databases, like understanding the behaviors of profitable customers to determining the likelihood of future loss on new business contracts. But there is a challenging detail; the decisions that are made are only as good as the data upon which they are based on [3].

Not only industry but also scientific literature is affected with bad data [4]. In their paper [4], Richard D. De Veaux and David J. Hand give examples from data quality problems discussed in scientific literature as " *In the 1978 Fisher Lecture "Statistics in Society: Problems Unsolved and Unformulated (Kruskal, 1981), Kruskal devoted much of his time to "inconsistent or clearly wrong data, especially in large data sets." He cited a 1960 census study that showed 62 women, aged 15 to 19 with 12 or more children. Coale and Stephan (1962) pointed out similar anomalies when they found a large number of 14-year-old widows. In his study Wolins (1962), a researcher attempted to obtain raw data from 37 authors of articles appearing in American Psychological Association journals. Of the seven data sets that were actually obtained, three contained gross data errors. A 1986 study by the U.S. Census estimated that between 3 and 5% of all census enumerators engaged in some form of fabrication of questionnaire responses without actually visiting the residence. This*

practice was widespread enough to warrant its own term: curbstoning, which is the "enumerator jargon for sitting on the curbstone filling out the forms with made-up information" (Wainer, 2004). While curbstoning does not imply bad data per se, at the very least, such practices imply that the data set you are analyzing does not describe the underlying mechanism you think you are describing.".

Explanation of poor data quality is quite simple. Data normally does not originate from systems that were set up with the primary goal of mining this data. So it is required to deal with operating systems that produce data only as a by-product. Even in the systems where data quality is considered as an important aspect during the design and implementation phases, these constraints are neglected in the long run. System is adapted to changing environment and data quality typically suffers [5]. In [5], Data Quality Mining, DQM, is introduced as the deliberate applications of data mining techniques for the purpose of data quality measurement and improvement that aims to detect, quantify, explain and correct data quality deficiencies in very large databases.

As stated in [6], data quality is a multidimensional, complex and morphing concept. In the last decade, it has become a burning issue in the areas of database statistics, workflow management, and knowledge engineering.

In this section, we will explain why data quality is or should be important to almost all organizations. First of all, poor data quality is pervasive. It is a plague to which industry is immune- nor is government or academia. It is very costly for any organization to have low quality data. It lowers customer satisfaction, adds expense, and makes it more difficult to run a business and pursue tactical improvements such as data warehouses and re-engineering. Not only customer, poor data quality also hurts employee's satisfaction which breeds organizational mistrust. It also impacts decision making. Implementing data warehouses with poor data quality levels is, at best, very risky. It has a very bad impact over quality of the discovered association rules [7].

While considering all these risks, it very important to think over that data quality can be a unique source of competitive advantage [2].

In order to improve data quality, it is vital to discuss the processes that have an impact on it. Understanding these processes which are sources of data inaccuracy will demonstrate the need for comprehensive program of assessment, monitoring and improvement. So in this section we will continue with classifying these processes.

We can classify these processes as *bringing data from outside*, *processes changing data within* and *processes causing data decay* [8]. Table 1 shows the main classes in detail.

Table 1: Processes that cause data inaccuracy

Initial Data Entry	**Data Decay**	**Moving and Restructuring**	**Using**
Mistakes	Decay	Extract	Faulty reporting
Data entry processes		Cleansing	Lack of understanding
Deliberate		Transformation	
System errors		Loading	
		Integration	

Initial data entry involves processes that bring data into database from outside- either manually or through various interfaces. Data entry mistakes are the most common source of data inaccuracy problems. Users enter *blue* but enter *bleu* instead; hit the wrong entry on a select list; put a correct value in the wrong field. As a brief, much of operational data originates from people and people make mistakes all the time.

Bad form designs via which data is collected also cause important data inaccuracies. For example using textboxes, where a list box can be used that includes valid values, support misspellings. Confusing fields are another common problem. They often lead users to enter wrong information. This case is handled in [9] and stated that many data quality problems arise from the "data misinterpretation"- that are problems with data semantics. The context is not captured in a form that is used by the query answering system and true answers are collected in different forms causing heterogeneities (e.g. "total sales in last year" that can be evaluated as last 12 months, last calendar year, or last fiscal year).

Another problem about form designs is that most of the form designs either mandate that a value be provided or allow it to be left blank. If left blank, it is not possible to know the difference between value-not-known and no-value-applies.

When the form requires that an entry be available and the entry person does not have the value, there is a strong tendency to fake it by putting a wrong, but acceptable value into field. This is unintentionally encouraged for selection lists that have a default value in the field to start with. Most of the times people enter wrong values on purpose which are called deliberate errors. There are three different reasons they do this.

- They do not know the correct information.
- They do not want to disclose the correct information.
- They get a benefit from entering the wrong information.

Such cases result in disguise missing data which is the topic of this thesis. A valid value is stored to the database but it is missing in real.

Not knowing the correct information occurs when the form requires a value for a field and the person wants or needs to complete the form but does not know the value to use. The form will not be complete without a value. People generally do not believe the value is important to the transaction, at least not relative to what they are trying to do. The result is that they make up a value, enter the information, and go on.

The second source of deliberate errors is caused by the people providing the data not wanting to give the correct information. This is becoming a more and more common occurrence with data coming off the internet and emergence of CRM applications. Every company wants a database on all of their customers in order to tailor marketing programs. However, they end up with a lot of incorrect data in their databases because the information they ask people for is more than people willing to provide or perceived to be an invasion of privacy.

As stated in [10], examples of fields that people tend to lie are age, weight, driver's license number, home phone number, marital status, annual income, and education level.

The third case in deliberate mistakes are made is where people obtain an advantage in entering wrong data. Such a case occurs very often in banks, manufacturers, and insurance companies when the company policy can encourage people to deliberate falsify information in order to obtain a personal benefit.

Processes that manipulate data within the organization also cause some errors. Periodic system upgrades, mass data updates, database redesign, and a variety of ad-hoc activities are examples of these processes. Lack of time and resources and unreliable meta data necessary to understand all data quality implications are the main reasons. In some cases, accurate data become inaccurate over time, without any physical changes. This usually happens when the real world object described by the data changes, but the data collection processes do not capture the change. For example, personal information in an employee database easily becomes wrong. People move, they change their marital status, they complete new education programs or they change telephone numbers. Most employees do not run into HR and fill out a form every time something changes in their life. The information in HR reflects the accuracy at the time they initially joined the company or last time an update was done.

Even ***data cleansing products*** that are commonly used to extract data from operational databases, put it into data warehouses, data marts, or operational data stores and find errors in datasets and correcting them, cause data imperfections. The problems arise since the tools used to support these processes do not help to understand the data in enough detail to effectively use and design decision support systems correctly.

The irony of it is that most projects claim to be extracting the data and cleaning it up before it is moved into the data warehouses, when in fact they are making it dirtier, not cleaner.

Poor attention paid to creating and maintaining accurate data in data dictionaries and meta data repositories is another significant reason of poor data quality. It costs companies to millions of dollars in unnecessarily complex data movement projects.

As a brief, inaccurate data gets into databases at a number of points and for a variety of reasons. Any program to improve data accuracy must address the issues across the entire spectrum of opportunities for error.

2.2 Data Quality Metrics

It is worthwhile to discuss the metrics that are used to measure data quality. In data quality assessments each organization must determine which dimensions are important to its operations and precisely define the variables that constitute the dimensions. The data quality literature provides a classification of data quality metrics even if there are inconsistencies on the definition of most dimensions due to the contextual nature of quality [11].

1. **Accuracy (Free of error):** Accuracy of a datum refers to the nearness of a value to actual correct value. It is calculated as;

$$Accuracy = 1 - \frac{Number\ of\ data\ units\ in\ error}{Total\ number\ of\ data\ units}$$

2. **Completeness:** There are three perspectives in this dimension; schema completeness, column completeness, and population completeness. Schema completeness is the degree to which entities and attributes are not missing from the schema. Column completeness measures if any missing values exist in a column of a table. By population completeness we mean the degree to which members of the population that should be present are not present.

   Each of the completeness type can be measured by;

$$Completeness\ rating = 1 - \frac{Number\ of\ incomplete\ item}{Total\ number\ of\ items}$$

3. **Consistency:** Consistency can also be analyzed in different perspectives. Consistency of redundant data, consistency between two related data elements or consistency for the format for the same data elements are the perspectives of this dimension.

$$Consistency\ rating = 1 - \frac{Number\ of\ instances\ violating\ specific\ consistency\ type}{Total\ number\ of\ consistency\ checks\ performed}$$

4. **Timeliness:** timeliness is the extent to which data are sufficiently up-to-date for a task which may be also defined as freshness, currency and volatility.

Data quality dimensions are also classified in 4 groups [7].

1. Quality dimensions which describe quality of management of data considering the satisfaction of technical and physical constraints (e.g. accessibility, ease of maintenance, reliability)
2. Quality dimensions considering the conceptual constraints on modelling and information presentation (e.g. conformance to schema, appropriate presentation)
3. Intrinsic data quality dimensions (e.g. accuracy, uniqueness, consistency)
4. Relative data quality dimensions with dependence on the user, on the application, on time or given knowledge state.

Map of data quality is displayed in [7] as given below;

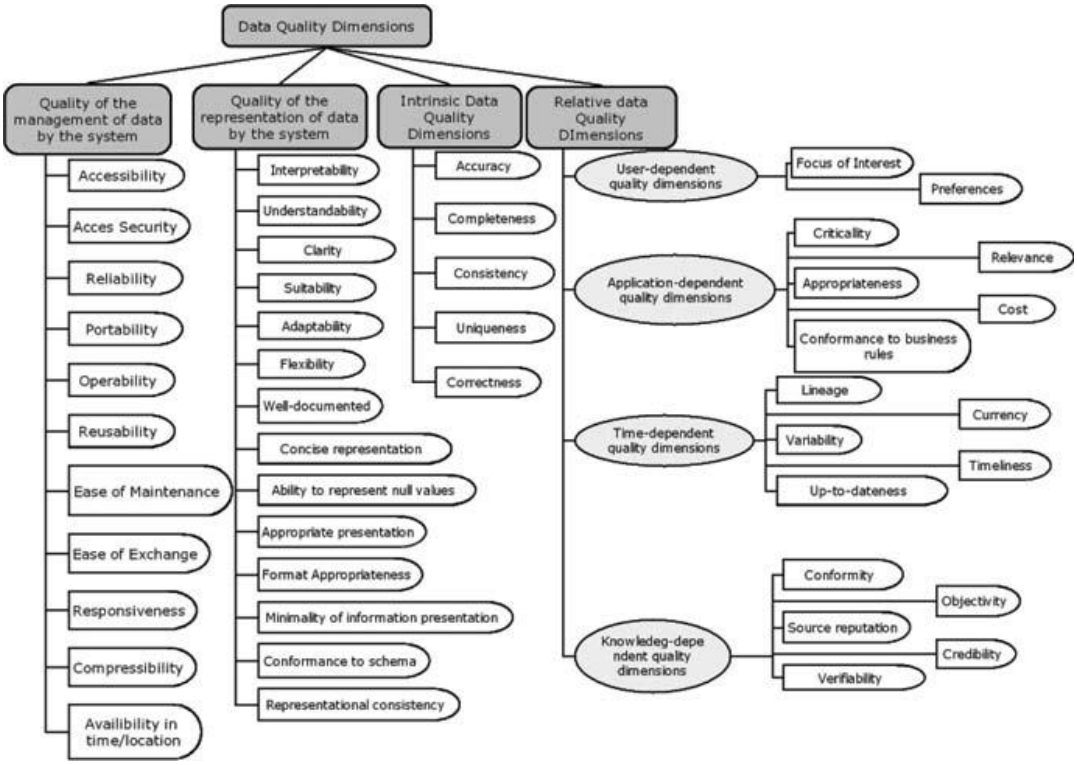

Figure 1: Map of data quality dimensions

2.3 Effects of Poor Data Quality

It is obvious that inaccurate data can lead to inaccurate results. But only 14% of public and private sector organizations worldwide have completely accurate data and 73% recognize that the data inaccuracy affects their bottom line [12].

A new airport in Hong Kong suffered catastrophic problems in baggage handling, flight information and cargo transfer because of dirty data in 1998. Flights took off without luggage, airport officials tracked flights with plastic pieces on magnetic boards, and airlines called confused ground staff on cellular phones to let them know where passengers could

find their planes. The new airport had been depending on the central database to be accurate but airport paid the price in terms of customer satisfaction and trust [13].

Independent research organization Dynamic Markets conducted a research about data quality and it is observed that fewer than two in ten US organizations claim that their data is 100% accurate, 77% admit that this problem is costing money (Table 2).

Another research by The Data Warehousing Institute shows that inaccurate data costs U.S. businesses $611 billion per year and half of the organizations surveyed plan to invest in better data management practices and use data quality to their advantage in planning for the future in global marketplace.

The rest of the research shows that 73% of organizations worldwide believe that inaccurate and incomplete data costs them money in terms of wasted resources; lost productivity and/or wasted marketing spend, compared to 66% of public sector organizations. 75% admit that potential revenue is lost through missed business opportunities from poorly profiled customer and prospect databases. Taking a global view, the average amount of waste due to inaccurate or incomplete data is perceived to 6%.

Table 2: Countries and their approaches to data quality [12]

Country	**Organizations claiming 100% accurate data**	**Admit to losing revenue due to poor data quality**
Singapore	36%	45%
France	22%	60%
Germany	22%	89%
Spain	18%	66%
United States	13%	77%
United Kingdom	8%	71%
Australia	7%	65%
Benelux	6%	86%

There are many drivers to data integrity for different organization from different countries. Most common ones are protection of the organization's reputation and enhancing customer satisfaction. Others are as listed in Table 3.

Table 3: Key drivers to data integrity [14]

Key drivers to keeping data clean	% of sample
Protection of the organizations' reputation	58
Cost savings reducing wastage and inefficiency	51
Capitalizing on market opportunities through customer profiling	49
Providing better service for citizens through profiling data	50
Compliance with regulations	35
Compliance with the Data Preference Services	20
Enhancing citizen/customer satisfaction	61

In the same research, barriers to maintaining accurate customer information are inquired. The top three are cited as: lack of time and internal resources (65%), keeping track of a transient customer base (55%), and lack of available budget (49%).

Table 4: Barriers to maintaining accurate databases

Barriers to maintaining accurate databases	% of sample
Time and internal resources	65
Available budget	49
Senior management support	9
Transient customers/prospects	55
Deceased customers/prospects	31
None – there are no barriers	22

As a result, despite organizations desire for clean and up-to-date databases, only 14% of them claim to have 100% accurate databases.

2.4 Data Quality Assessments

In previous sections, we discussed definitions, causes and effects of data quality. In this section we will explain current assessments in data quality and common data quality tools and their abilities and limitations.

Maydanchik handles the data quality in three main steps in [8] : *identification of data quality rules, design* and *implementation* of them. Data profiling is handled before the data

quality assessment in which basic statistics like value frequencies and distribution charts are produced. This step enables data quality experts to learn what the data looks like. So it is very important for an efficient data quality assessment. Because actual data is very different from what theoretically expected. Data models and dictionaries become inaccurate over time.

There are many data profiling techniques. Most useful ones are depicted as [8]:

Attribute profiling examines the values of individual data attributes and searches basic aggregate statistics, frequent values, and value distribution.

Relationship profiling in which entity keys and relationships and counting occurrences for each relationship in the data model are identified.

State- transition model profiling is a collection of techniques for analysis of the lifecycle of state-dependent objects and provision of actual information about the order and duration of states and actions.

Dependency profiling uses computer programs to look for hidden relationships between attribute values.

2.4.1 Designing Data quality Rules

Maydanchik suggests designing quality rules based on the constraints that validate data relationships [8]. He proposes five main categories: "***Attribute domain constraints***", "***Relational integrity rules***", "***Rules for historical data***", "***Rules for state-dependent objects***" and "***Attribute dependency rules***".

Attribute domain constraints restrict the allowed values on attributes. The simplest kind of attribute domain constraint is stated as *optionality constraint* which prevents attribute from taking Null, or missing value. For example, social security number should not be missing in a census dataset.

There are many approaches to handle missing values and it is very easy to detect explicitly missing values during data profiling. But main problem arises when default values given in a form are recorded into datasets when user does not provide an answer for many reasons as mentioned in the previous section. Database designers are often unaware of such default values, and data dictionaries rarely list them. Maydanchik suggests analyzing frequent values or detecting outliers to detect disguise missing data. He states the most typical disguise missing value for numeric attributes is 0 but he also admits that other values are also assigned as default values in many systems especially in legacy systems.

Another attribute constraint is *format constraint* which handles the valid representations of attributes. In databases some attributes have a variety of representations. For example, date 11/15/06 can also be displayed as 15-Nov-06 or 11152006, or in many other ways. In order to handle this variety, valid representation is given in data dictionaries as a value mask. For example, a value mask for an attribute *date* may be given as MMDDYYYY. If no mask

is provided, frequent formats are detected via data profiling tools and an appropriate mask is generated.

The third attribute constraint is *valid value constraint* which limits permitted attribute values to such a prescribed list. *Precision constraint* is another constraint frequently used in data quality assessments. They require all values of an attribute to have the same precision, granularity, and unit of measurement. Existing data profiling tools can handle this constraint generating precision frequency charts.

Constraint that can be evaluated in a quality assessment can also be derived from relations that tie the parts of the same object. Rules that are derived from these constraints are called ***Relational integrity rule.*** *Identity rule* is the simplest one which validates that every record in a database table corresponds to one real world entity and that no two records reference the same entity. Any duplicate values are erroneous. There are various tools on the market that can handle duplicate values.

Reference rule is another rule explained by Maydanchik as *"every reference made from one entity occurrence to another entity occurrence can be successfully resolved*". Each reference rule represented in relational data is modeled by a ***foreign key*** that holds the database together. Here the aim is to "avoid that navigation of a reference across entities does not result in a *dead end*".

Cardinal rules define the constraints on relationship cardinality which defines the allowed number of occurrences of relationships. Relational cardinality is often represented incorrectly in relational data models. The reasons are stated by Maydanchik as "optionality is sometimes built into the entity-relationship diagrams simply because real data is imperfect. Strong entities are routinely allowed to have no corresponding weak entity record simply because database designers expect bad and missing data". This rule can be handled by existing data profiling tools that counts actual occurrences for each relationship in the data model.

Historical data is the most error prone data but it is very efficient to derive rules from. Easiest assessment of historical data is working on *time-dependent attributes* which hold object characteristic that changes over time. Rules derived from such data are broken down into four categories: *currency*, *retention*, *continuity*, and *granularity*.

Currency rules enforce the desired freshness of the historical data identifying the youngest record in the historical data and comparing its timestamp to a pre-defined threshold. *Retention rules* deal with depth of the historical data which often reflects common retention policies and regulations requiring data to be stored for a certain period of time before it can be discarded. For example, a bank may be required to keep data of all customer transactions for several years.

Values of some attributes are more meaningful when accumulated over a period of time. For example, it may be pertinent to collect weekly, monthly, or quarterly cumulative sales volumes. Such series of cumulative time period measurements are called as *accumulator history* and it leads two main rules on historical data: *granularity rules* and *continuity rules*. *Granularity rules* require all measurement periods to have the same size and *Continuity rules* prohibit gaps and overlaps in accumulator histories. These constraints can easily be designed as rules to check data quality on this respect.

The fourth kind of data quality rules discussed by Maydanchik is the rule for ***state-dependent objects***. In this assessment, states of an historical object and actions that changes states of it are defined via data profiling or data dictionaries and both actions and states are validated. In order to achieve these steps, state-transitions diagrams of valid states and actions are developed and validation of each action and state is checked according to this diagram.

State-transition model profiling is generally more complex than regular attribute profiling.

Maydanchik points out that there is no tool that specifically addresses *State-transition model profiling* but some existing tools can be used on that purpose with some changes. Also it can be achieved by some advance queries and data manipulation techniques.

Most challenging data quality assessment is based on general attribute dependency rules without knowing the type of the dependency. "Two attributes are called dependent when the value of first attribute influences possible values of the second attribute" [8] . Attribute dependencies fall into five broad categories: ***redundant attributes***, ***derived attributes***, ***partially dependent attributes***, ***attributes with dependent optionality***, and ***correlated attributes***. Data quality rules are derived differently for each dependency category.

Redundant attributes are data elements which refer to the same attribute of a given entity and used for specific purposes especially in legacy systems. They provide an important constraint that value of an attribute must be compatible with the value of its redundant attribute. A special kind of redundant attribute is derived attributes whose values are calculated based on some other attributes. So it is very easy to detect quality on these attributes generating a formula for the rule like:

Attribute1 = Func(Attribute2, Attribute3,...,AttributeN)

Dependency may occur partially in datasets and the value of one attribute may just restrict possible values of another attribute to a smaller subset, but not to a single value as in derived attributes. These attributes are called as ***partially dependent***. Different rules are derived from these attributes based on the content of the dependency. For example, in a census dataset, age of a mother is partially dependent to age of her child that age of mother is higher than the age of her child.

Up to here, we have discussed dependency in datasets which leads to clear-cut solutions. But ***correlation*** is another important aspect which provides many data quality rules. A correlation value of Attribute1 may influence the expectations for Attribute2 and vice versa. In such cases, it is suggested to build a chart of likely value pairs and treating the other value pairs as potentially erroneous. This rule may yield some false positives, but it will also catch many errors. Many existing tools execute dependency profiling for hidden relationships using complex statistical models and pattern recognition techniques.

Value clustering is another concept in dependency profiling. It occurs when the distribution of attribute values fall into two or more clusters depending on the values of another attribute and indicates that partially dependent attributes that can be translated into conditional data quality rules.

In summary, Maydanchik lists the assessments that can be achieved using existing data quality rules or some database queries. He points only two concepts in which current tools are incapable: detecting disguised missing data and state transition model profiling. It is also important to emphasize that most of the approaches depend on domain knowledge like business rules, regulations, valid value lists or constraints provided by the database administrator.

In [15], Leo L. Pipino et.al. handle the data quality assessments in two aspects: the subjective data quality assessments and the objective data quality assessments. They refer to needs and experiences of stakeholders like the collectors, custodians, and consumers of data products as subjective aspect of data quality. They mention about questionnaires via which data quality can be measured. They emphasize that questionnaires are frequently used in healthcare, finance, and customer product companies.

Object quality assessment is classified into two groups in [15] as task-independent assessments and task-dependent assessments. In task-independent assessment, task-independent metrics used reflect states of the data without the domain knowledge of the application, and can be applied to any data set, regardless of the tasks and in task-dependent assessments, task-dependent metrics used include the organization's business rules, company and government regulations, and constraints provided by the database administrator.

Three pervasive functional forms are given [15] as ***ratio***, ***min or max operation***, and ***weighted average*** which are common methods that are used with quality metrics. The simple ratio measures the ratio of desired outcomes to total outcomes. It is used with the metrics such as *free-of-error*, *completeness*, and *consistency.*

Other metrics like *believability* and *appropriate amount of data* are preferred to be indicated with the lower bound that can be acceptable in quality assessment. Metrics like timeliness and accessibility are preferred to be mentioned with the highest bounds. For such metrics, ***min or max operations*** are used.

In believability metric, one may want to specify the degree of importance of each of the variables to the overall believability measure. In such a case the ***weighted average*** is an appropriate form to use.

In [15], they present an approach that combines the subjective and objective assessments of data quality, and illustrate how it has been used in practice.

In order to use both the subjective and objective metrics to improve organizational data quality, they define three steps:

1. performing subjective and objective data quality assessments,
2. comparing the results of the assessments, and identifying discrepancies, and determining root causes of discrepancies; and
3. determining and taking necessary actions for improvement.

While evaluating both the subjective and objective assessments of a specific dimension, the outcome of the analysis will fall into one of four quadrants as given below;

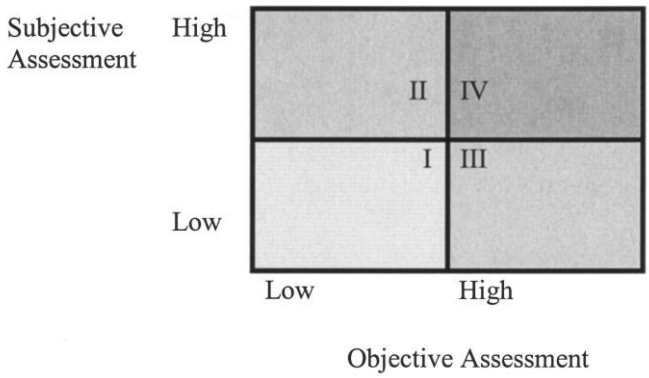

Figure 2: Subjective and Objective Assessment [15]

The goal is to achieve a data quality state that falls into Quadrant IV in which both of the assessments are high. So in this approach both the objective and subjective assessments are taken into account as indicated before.

Richard D. De Veaux et.al. [4] propose that the best way to improve the quality of data is to improve things in the data collection phase and they emphasize the importance of prevention and detection of errors from arising in the first place. They suggest the usage of data dictionaries to check the validity of records. They also propose *Pareto principle* which declares that most of the errors are attributable to just a few variables since some variables are intrinsically less reliable (and important) than others. So they propose to improve the overall level of quality significantly by removing just a few of these low quality variables.

Another paper [9] published by Stuart Madnick, Hongwei Zhu defines the reason of many ‘data quality problems as data misinterpretation problems’—that is, problems caused by heterogeneous data semantics. They propose a framework called COntext INterchange

(COIN) which is a knowledge-based mediation technology that enables meaningful use of heterogeneous databases where there are semantic differences. COIN identifies the semantic differences and reconciles them by its mediation service. The overall COIN also wraps technology and middleware services for accessing the source information and facilitates the integration of the mediated results into end-user applications where the wrappers are physical and logical gateways providing a uniform access to the disparate sources over the network.

In another paper [16], authors summarize processes of data quality assessments as (i) auditing data to identify the types of anomalies reducing the data quality, (ii) choosing appropriate methods to automatically detect and remove them, and (iii) applying the methods to the tuples in the data collection and add another task (iv), the post-processing or control step where they exam the results and perform exception handling for the tuples not corrected within the actual processing. They define the last step as semi-automatic process where the control of its execution is done by one or more domain experts, i.e., experts with knowledge about the mini-world and its regularities and peculiarities.

CHAPTER 3

DISGUISED MISSING DATA

3.1 Overview

An extremely common anomaly in large datasets is that of missing data, which corresponds to legitimate data values in a dataset but do not exist for numerous reasons. Missing data arise quite frequently in practice, but as stated by Pearson [17], this problem can manifest itself in at least three different ways: the desired record *x* can simply be missing from the dataset; it can be coded as a special missing data value like NA, NaN, or ? ; or it can be disguised as a valid data value with no indication that the correct value of x is unknown or indefinable. This last case is particularly insidious because it can convert missing data values into multivariate outliers that may be difficult to detect [17].

There are two different ways to detect disguise values in a data set: The first one is to use a semi-autonomous approach that depends on domain background knowledge [1]. A domain expert can filter entries with suspicious values that contradict with the semantics of the attributes. Another semi-autonomous way is to detect distribution anomalies which require knowing the expected distribution. They all depend on domain knowledge and cannot detect inliers.

The second type is to use a full autonomous approach that does not require using any background information. In this thesis, we have investigated the latter one as it has been demonstrated that the former method fails when disguise values are inliers [1].

In this thesis, we particularly worked on the method proposed by Ming Hua and Jian Pei [1]. In this section, first we are going to explain their approach, then demonstrate the drawbacks and finally we are going to present our improvement.

3.2 Definition of Disguised Missing Data

Attribute A in a tuple *t* is denoted by *t.A* and called entry. At the end of the data collection three situations may arise for the entry *t.A*.

Case 1: The user enters a value that reflects the fact and *t.A* is not missing.

Case 2: The user does not provide a value and *t.A* is explicitly missing.

Case 3: The user does not prefer to enter the factual value but a valid value of A is recorded due to some data collection mistakes. Although the entry value is missing in its nature, a fake value is recorded and *t.A* is disguised missing.

Let T be the truth table and $\check{T}$ be the recorded table. Here, T contains the data that should be recorded and $\check{T}$ contains the data that is recorded. Particularly an entry is called disguised missing if $t.A=null$ but $\tilde{t}.A \neq null$.

Example 1: Consider an attribute Gender on an online application form on a frequent flyer program. It has two choices: male or female. System may set a value, say male in this example, as default value and some women may not want to reveal this information or want to skip filling this attribute. As a consequence missing values may disguise themselves as the default value, male. In such a case, in truth table T, values of attribute Gender for these women are *female* but in reality they are recorded as *male* in the table $\check{T}$.

Default values support the occurrence of disguise values but they are not the only reason. For example attribute birth date is generally wanted to be disclosed. January 1 (the first value in the pop-up lists of month and day, respectively) is chosen in order to pass.

Consequently, disguised missing data is much more challenging than explicitly missing values. In explicitly missing values, entries that are missing are known and strategies can be developed to handle these entries. For disguised missing data we do not know even which entries are missing and, for example in example 1, we cannot decide which entries as male are real and which are missing.

In [1], published by Ming Hua and Jian Pei, in order to detect disguise values, they analyzed the distribution of disguise missing values and utilized the embedded unbiased sample (EUS) heuristic that often holds for disguised missing values. According to EUS, the projected database of a disguise value often contains a large unbiased sample of the whole dataset. Based on this property, they proposed a general framework to identify suspicious frequently used disguised values.

3.3 General Approach

This approach is based on the heuristic that random disguising often does not happen extensively in practice. A small number of values are frequently used as the disguises (one or two in an attribute). They make the following assumption:

> **Assumption 1 (Frequently Used Disguises):** On an attribute, there often exist only a small number of disguises that are frequently used by the disguised missing data.

Example 2: (EUS Heuristic)

Let's analyze the case given in Example 1. $\check{T}_{male}$ is the set of tuples carrying value male on attribute gender. $\check{T}_{male}$ can be divided into two exclusive subsets as shown below.

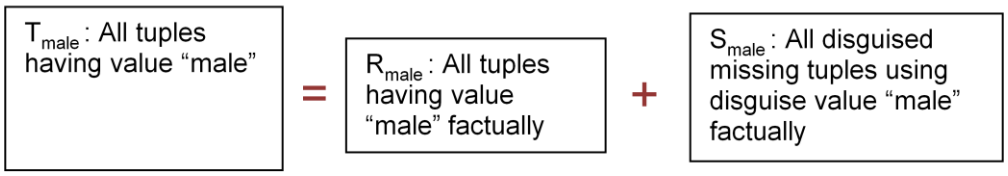

Figure 3: The EUS Heuristic

Here, depending on the heuristic that random disguising often does not happen extensively in practice, we can say that the set S_{male} is an unbiased sample of the truth table except for attribute gender itself (all tuples in S_{male} take value male on gender). Similarly, we can also divide T_{female}, the set of tuples having value female on attribute gender, into two subsets R_{female} and S_{female}. If value male is used more frequently as the disguise value on attribute gender, then S_{male} from T_{male} is larger than S_{female} from T_{female}. According to Assumption 1, on each attribute, there are only a very small number of values that are used as disguises. In other words, it is likely those disguise values contain subsets of tuples that are unbiased samples of the whole data set. As a heuristic, if a value contains a large subset of tuples that is an unbiased sample of the whole data set, this value is suspicious of a disguise value. It is possible to clarify the proposal with following heuristic for disguise missing values.

The Embedded Unbiased Sample Heuristic: *If v is a frequently used disguise value on attribute A, then* $T_{A=v}$ *contains a large subset* $S_v \subseteq \check{T}_{A=v}$ *such that* S_v *is an unbiased sample of* $\check{T}$ *except for attribute A where* $\check{T}_{A=v} = \{ \tilde{t} \in \check{T} | \tilde{t}.A=v \}$.

You can see the frequently used notations in the Table 5.

Table 5: Frequently used notations

Symbol	Explanation
T	The truth table
Ť	The recorded table
Ť'	A subset of Ť
t̃	A tuple in recorded table
t̃.A	An entry in the recorded table
$\check{T}_v$	The projected database of value *v*
S_v	The disguised missing set of *v*
M_v	The maximal embedded unbiased sample of *v*
$\phi(\tilde{T}, \tilde{T}')$	The correlation-based sample quality score

3.3.1 General Framework

For each value *v* on attribute *A*, let T_v be the set of tuples carrying value *v* in the truth table. Clearly, $T_v \subseteq \check{T}_v$. Then, $S_v = (\check{T}_v - T_v)$ is the set of tuples using *v* as the disguise on attribute *A*. We call S_v the disguised missing *set* of *v*. According to the EUS heuristic, S_v is an unbiased sample of Ť. The larger the size of S_v, the more frequently *v* is used as the disguise value. A value *v* is called a frequent disguise value if it is frequently used as disguises.

In order to find disguise values on an attribute A, it is required to find small number of attribute values whose projected databases contain a large subset as an unbiased sample of the whole table. Such attribute values are suspects of frequently used disguise values. The larger the unbiased sample subset, the more likely the value is a disguise value. So it is required to find maximal unbiased subset Mv, maximal embedded unbiased sample, or MEUS for short. While analyzing the subsets, two measures of Mv must be considered; size and quality. Quality can be explained as how well the subset resembles the distribution of the whole dataset. The values with large and high quality MEUS should be reported as the suspicious frequent disguise values.

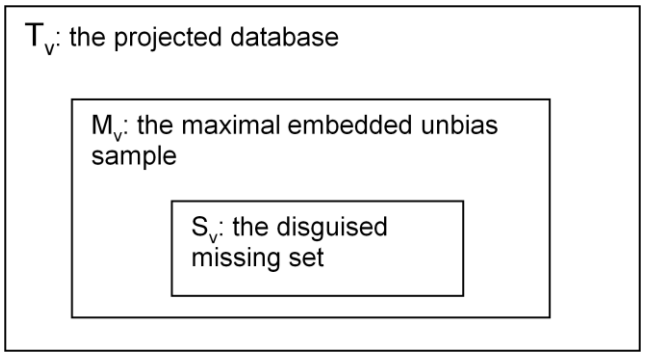

Figure 4: The relationship among concepts

This heuristic fails in some cases. For example, many people may submit their tax returns on the deadline day. So on attribute **submission-date**, the projected database of the dates right before the deadline may likely contain large unbiased samples of all tax returns whereas these dates are not frequently used as disguise values. So it is reasonable to ensure that EUS heuristic fits a data set, before applying the approach**. If most of the projected databases are unbiased samples, the heuristic should not be applied.** Also**,** if the disguised missing values are independent and are random values in the domain of the attribute, it is very hard to unmask them.

Based on these discussions, a general framework is generated as given below [1].

Phase 1: Mining candidates of frequent disguise values

Input: A table *T* and a threshold on the number of candidates of frequent disguise values *k*;

Output: for each attribute, *k* candidates of frequent disguise values;

Method:

1: FOR each attribute *A* DO

2: // applicability test
check whether the projected databases of most
(frequent) values on *A* are unbiased samples of *T*,
if so, **break**;

3: **FOR** each value *v* on *A* DO derive M_v;

4: find the value(s) with the best and largest M_v's;

END FOR

Phase 2: postprocessing: verify the candidates of frequent disguise values;

Two important challenges arise here;

1. How to measure whether a set of tuples is an unbiased sample of a table? (3.3.2 Correlation Based Sample Quality Score (CBSQS))
2. How to compute a maximal embedded unbiased sample M_v from the projected database $\check{T}_v$? (3.3.3 Finding Approximate MEUS'S)

3.3.2 Correlation Based Sample Quality Score (CBSQS)

In order to measure whether Ť' is a good sample of Ť, they propose to use correlation analysis with the assumption that correlations can capture the distribution of a data set nicely. The approach is straightforward: if the values correlated in Ť are also correlated in Ť' and vice versa, then the possibility of Ť' and Ť having similar distribution will be high.

Based on the paper [1], the correlation based sample quality score was given as follows: Given table Ť, on attributes A1…..An and subset Ť' ⊂ Ť, we want to measure whether Ť' is a good sample of Ť. If values which are correlated in Ť are also correlated in Ť' and vice versa, then likely Ť' and Ť are of similar distribution.

The correlation between v_i and v_j is given by;

$$Corr(v_i, v_j) = \frac{P(v_i, v_j)}{P(v_i) * P(v_j)} = \frac{P(v_i|v_j)}{P(v_j)} \quad \text{(Equation 1)}$$

Based on this approach, they use correlations of pairs of values to measure how good a sample Ť' is with respect to Ť. They compare the correlations in Ť' and Ť and calculate correlation-based sample quality score (***CBSQS)***, denoted by $\phi(\widetilde{T}, \widetilde{T'})$.

$$\sum_{P_{\check{T}'}(v_i, v_j) > 0} \left(\frac{P_{\check{T}'}(v_i, v_j)}{1 + |Corr_{\check{T}}(v_i, v_j) - Corr_{\check{T}'}(v_i, v_j)|^q} \right) \quad \text{(Equation 2)}$$

The score returned form CBSQS is a non-negative number. The higher the score, the better Ť' is an unbiased sample of Ť.

In CBSQS, quality of the MEUS is computed. Recall that to measure whether a value *v* is a frequent disguise value; we consider both quality of MEUS and relative size of M_v with respect to T_v. So they define the disguise value score (DV-score for short) of a subset U⊆Ť as;

$$\text{dv(v, U)} = \frac{|U|}{|\check{T}|} * \boldsymbol{\phi}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}'}) \quad \text{(Equation 3)}$$

Based on this formula, frequent disguise value score is defined as;

$$\text{dv(v)} = \max_U\{\text{dv(v, U)}\} = \max_U\left\{\frac{|U|}{|\check{T}|} * \boldsymbol{\phi}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}'})\right\} \quad \text{(Equation 4)}$$

M_v is selected as the subset maximizing the DV-score. That is,

$$Mv = \arg\max\left\{\frac{|U|}{|\check{T}|} * \boldsymbol{\phi}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}'})\right\} \quad \text{(Equation 5)}$$

3.3.3 Finding Approximate MEUS'S

The biggest challenge in finding approximate MEUS is that DV-score is not monotonic with respect to the set containment relation. For a subset U⊆$\check{T}_v$ and W⊆U, dv(v,W) may be higher or lower than dv(v,U). So especially in large datasets, computation is too costly. For example in a set with ***n*** tuples, there are many nonempty subsets ($\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n}$) and CBSQS should be measured for all these subsets to find the optimal good sample which is a very costly task.

They adopt a greedy approach to deal with the problem. The algorithm starts with the Ť as the initial sample U and in each iteration they remove a tuple t̃ and compute DV score for (U-{ t̃ }). A tuple with the largest positive DV-score gain is removed from the current sample set as the result of current iteration. The iteration keeps removing the tuples until DV-score cannot be improved by removing any tuple from the current sample.

Input: a table *T* and a value *v* on attribute *A*;
Output: approximate *Mv*;
Method:
1: $U \leftarrow Tv$;
2: REPEAT
3: FOR EACH tuple e $\tilde{t} \in U$
compute the DV-score gain of (U -{ $\tilde{t}$ }) over U;
4: remove a tuple $\tilde{t}_0$ with the largest DV-score gain if the gain is positive;
5: UNTIL no tuple can be removed;
6: RETURN *U*;

We can sum up the described approachs with an example. While collecting a dataset via a web form in which the field *Gender* is selected from radio buttons let's assume that the value *Male* is selected as default value. In this form, most of the women will just skip the attribute and disguise their gender as *Male*. In such a case the projected table of *Male*, $T_{Male,}$ will involve a large unbiased sample of the whole dataset, Ť. Here the aim is to reach to S_{Male} where the tuples include *Male* as disguise missing value. It is very difficult to reach these tuples exactly, but is possible to converge them leading to the set M_{Male} that covers and converges to S_{Male}. The larger the size of M_{Male}, the more suspicious the value Alabama is. It is also considered how much M_{Male} is represents the whole dataset Ť. In order to reach $M_{Male,}$ a greedy algorithm is used and for the subsets that are determined with this algorithm, CBSQS is calculated. The value, here *Male*, whose projected table returns the largest CBSQS, is returned as disguise value.

3.3.4 Analysis of CBSQS

In order to analyze effectiveness of CBSQS, the equation will be detailed as follows:
Let's assume that we analyze a dataset with four attributes; A1, A2, A3 and A4 whose value ranges are as given below.

$A1=\{v1_{A1}, v2_{A1}, v3_{A1}, v4_{A1}\}$

$A2=\{v1_{A2}, v2_{A2}, v3_{A2}, v4_{A2}\}$

$A3=\{v1_{A3}, v2_{A3}\}$

$A4=\{v1_{A4}, v2_{A4}, v3_{A4}\}$

Let's name the score of one value couple depicted in as ***value couple score***, ***VCS,*** and start to analyze the method:

$$VCS(v_i, v_j) = \frac{P_{\check{T}}(v_i, v_j)}{1 + \left|Corr_{\check{T}}(v_i, v_j) - Corr_{T'}(v_i, v_j)\right|^q} \quad \text{(Equation 6)}$$

Let's consider that we want to measure the disguise value for attribute A1 and compute unbiased sample within $\check{T}_{v1}$'. We can rewrite as the summation of the scores of the attribute couples A2&A3 and A2&A4 and A3&A4. Φ_{A2A3} returns the score of value couples of A2 and A3, Φ_{A2A4} returns the score of value couples of A2 and A4 and Φ_{A3A4} returns the score of value couples of A3 and A4.

$$\boldsymbol{\phi}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') = \boldsymbol{\phi A2}\&\boldsymbol{A3}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') + \boldsymbol{\phi A2}\&\boldsymbol{A4}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') + \boldsymbol{\phi A3}\&\boldsymbol{A4}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') \quad \text{(Equation 7)}$$

According to EUS heuristic score of each attribute couple in must be high enough to classify a sample as unbiased sample of a dataset. Here, the attribute couple score for A2 and A3 is calculates as;

$$\boldsymbol{\phi A2}\&\boldsymbol{A3}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') = \text{VCS}(v1_{A2}, v1_{A3}) + \text{VCS}(v1_{A2}, v2_{A3}) + \text{VCS}(v2_{A2}, v1_{A3}) + \text{VCS}(v2_{A2}, v2_{A3}) + \text{VCS}(v3_{A2}, v1_{A3}) + \text{VCS}(v3_{A2}, v2_{A3}) + \text{VCS}(v4_{A2}, v1_{A3}) + \text{VCS}(v4_{A2}, v2_{A3})$$

(Equation 8)

Also in order to support EUS heuristic score of each value couple in must be high enough to classify a sample as unbiased sample of a dataset. Here, the attribute couple scores for "A2 and A3" and "A3 and A4" are calculated as;

$$\begin{aligned}\boldsymbol{\phi A2}\&\boldsymbol{A4}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') = {} & \text{VCS}(v1_{A2}, v1_{A4}) + \text{VCS}(v1_{A2}, v2_{A4}) + \text{VCS}(v1_{A2}, v3_{A}) + \\ & \text{VCS}(v2_{A2}, v1_{A4}) + \text{VCS}(v2_{A2}, v2_{A4}) + \text{VCS}(v2_{A2}, v3_{A}) + \\ & \text{VCS}(v3_{A2}, v1_{A4}) + \text{VCS}(v3_{A2}, v2_{A4}) + \text{VCS}(v3_{A2}, v3_{A}) + \\ & \text{VCS}(v4_{A2}, v1_{A4}) + \text{VCS}(v4_{A2}, v2_{A4}) + \text{VCS}(v3_{A2}, v3_{A})\end{aligned}$$

(Equation 9)

$$\begin{aligned}\boldsymbol{\phi A3}\&\boldsymbol{A4}(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}') = {} & \text{VCS}(v1_{A3}, v1_{A4}) + \text{VCS}(v1_{A3}, v2_{A4}) + \text{VCS}(v1_{A3}, v3_{A}) + \\ & \text{VCS}(v2_{A3}, v1_{A4}) + \text{VCS}(v2_{A3}, v2_{A4}) + \text{VCS}(v2_{A3}, v3_{A})\end{aligned} \quad \text{(Equation 10)}$$

Given the new equations forms and new terms, in the rest of this section we analyzed Equation 1 and made some assumptions in which cases this formula may fail. To demonstrate the drawbacks, we use real world and synthetic datasets.

Given table $\tilde{T}$ on attributes A1, …, An and a subset $\widetilde{T}' \subset \tilde{T}$, let v_{i_j} be the i^{th} value on attribute Aj.

Assumption 1:

If a subset includes most of the value couples that occur frequently in the dataset than the score will return high. Because, for a frequent value couple, $P_{\check{T}}(v_i,v_j)$ will be high and also $Corr_{\check{T}}(v_i,v_j)$ and $Corr_{\check{T}'}(v_i,v_j)$ will be high and similar which leads $|Corr_{\check{T}}(v_i,v_j) - Corr_{\check{T}'}(v_i,v_j)|$ to converge to zero. When scores of such frequent values are summed, $\phi(\widetilde{T}, \widetilde{T}')$, the total score will be high enough to select the subset as an unbiased sample.

This approach was adopted in the paper "Cleaning Disguised Missing Data: A Heuristic Approach" by Ming Hua, Jian Pei [1].

Assumption 2:

High dependency between attribute values can bias the result and return high sample quality score although the subset T_v excludes many value couples in dataset *D* and distribution of T_v cannot represent the distribution of *D.*

Assume that in a dataset with four attributes A1, A2, A3, and A4, there is a depedency between the values $v1_{A1}$, $v1_{A2}$ and $v1_{A3}$ as;

$A1=v1_{A1} \rightarrow A2 = v1_{A2}$

$A1=v1_{A1} \rightarrow A3 = v1_{A3}$

While computing disguise value(s) for the attribute A1, $T_{v=A1}$ will include the following tuples into the calculation ;

Table 6: $T_{v=A1}$

A2	A3	A4
v1A2	v1A3	…
v1A2	v1A3	…
v1A2	v1A3	…
v1A2	v1A3	…
v1A2	v1A3	…

While computing the ***CBSQS*** for this subset, , $P_{\check{T}}(v1_{A2},v1_{A3})$ will be very high if $A1=v1_{A1}$ is observed frequently. Correlation between these values will also be high and $|\mathrm{Corr}\check{T}(v1_{A2}, v1_{A3}) - \mathrm{Corr}_{T'}(v1_{A2}, v1_{A3})|$ will converge to zero in . So VCS($v1_{A2}$, $v1_{A3}$) will return a very high value. But, because all other value couples of A2 and A3 are excluded, VCS($v1_{A2}$, $v2_{A3}$) , VCS($v2_{A2}$, $v1_{A3}$), VCS($v2_{A2}$, $v2_{A3}$) , VCS($v3_{A2}$, $v1_{A3}$), VCS($v3_{A2}$, $v2_{A3}$), VCS($v4_{A2}$, $v1_{A3}$) and VCS($v4_{A2}$, $v2_{A3}$) will all return zero.

We propose that if $v1_{A1}$ is observed frequently in the dataset, the result of VCS($v1_{A2}$, $v1_{A3}$) can dominate the low score of other value couples in $\phi A2\&A3(\tilde{T}, \tilde{T}')$. Such a dependency can also **cause** $\boldsymbol{\phi A2}\&A3(\tilde{\boldsymbol{T}}, \tilde{\boldsymbol{T}}')$ to dominate the low results of $\boldsymbol{\phi A2}\&A4(\tilde{\boldsymbol{T}}, \tilde{\boldsymbol{T}}')$ and $\boldsymbol{\phi A3}\&A4(\tilde{\boldsymbol{T}}, \tilde{\boldsymbol{T}}')$.

Such a case results in T_{vA1} being found as an unbiased sample of D which is exactly unreasonable.

To illustrate it, we generated a synthetic data set in 3.4.1 CASE 1: Dependency Effect.

Assumption 3:

We remarked that EUS heuristic will not work when we want to detect disguise value for a random attribute. But randomization may bias the results even computing disguised value for a nonrandom attribute if the dataset includes random attributes.

Assume that the attributes A2 and A3 are totally independent attributes and values of them are randomly distributed. While computing disguise value(s) for attribute A1, the projected database T_{vA1} is most likely to include these random value couples of A2 and A3 and as result of their semantics, $\boldsymbol{\phi A2}\&A3(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}')$ will return a high value. This score can lead a high total score by itself even $\boldsymbol{\phi A2}\&A4(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}')$ and $\boldsymbol{\phi A3}\&A4(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}')$ return low scores.
To illustrate it, we generated a synthetic data set in 3.4.2 CASE 2: Independent Attributes Effect.

Assumption 4:

If a subset includes derived attributes, in which value of an attribute A2 is derived from another attribute(s) A3, specific values of A2 will be frequently observed with the specific values of A3. In such a case, joint probability between these specific value couples will be very high and $\boldsymbol{\phi A2}\&A3(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}')$ will return a high value. Another problem will occur when score of $\boldsymbol{\phi A2}\&A3(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}')$ dominates the scores of $A2\&A4(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}')$.

To illustrate it, we generated a synthetic data set in 3.4.3 CASE 3: Derived Attributes Effect.

3.4 Problems with the Correlation-Based Sample Quality Score

In this section, we analyze the effect of summation of VCSs while computing attribute couple scores in order to demonstrate the deficiencies of CBSQS score.

In order to clarify the assumptions, we generated synthetic datasets and computed disguise values. In datasets we aimed a simple approach; in the generated dataset, there should be T_{vi} which support EUS heuristic and there should be T_{vj} which does not support EUS heuristic but defines our assumptions 2, 3 or 4. The synthetic datasets are inspired from real world data sets such as Turkish Census Data Set for year 2001. We focused on census datasets and computed disguise value for the attribute "Age" in all datasets. While generating datasets, we selected tuples that form T_{vi} and then injected the tuples that form T_{vj}. We provided a framework in which we can compete those projected databases in being most unbiased sample with given specific support values.

3.4.1 CASE 1: Dependency Effect

In [3], it has been stated that two attributes are dependent when the value of the first attribute influences possible values of the second attribute.

We expect a value that follows a dependency rule not to be detected as a suspicious disguise value because such rule compliance suggests that this value is entered by the user consciously considering the semantic of the attribute. It also opposes to EUS heuristic which proposes that a disguise value has an unbiased sample of the dataset.

We have generated a synthetic census dataset to demonstrate our argument.

3.4.1.1 Data Set Characteristics:

We have generated a dataset comprising 1000 tuples with 3 dimensions indicating "Age", "Literacy" and "School" attributes respectively. The attributes and their value ranges are given below. Both attributes and value ranges are taken from a real world data set which is Turkish Census Data set [18].

Table 7: Values of "Age" [18]

Value	Age Range
1	0--5
2	6--14
3	15--17
4	18-35
5	Over 35

Table 8: Values of "Literacy"

Value	Meaning
1	Literate
2	Illiterate

Table 9: Values of "Graduated school"

Value	Meaning
0	Illiterate
1	Primary school
2	Secondary school
3	High school
4	University

The four attributes are dependent on each other. For example, "literacy" attribute value "illiterate" is only observed when "graduated school" is set to 0. Likewise, specific values in "Age" attribute can only be seen with particular values in "Literacy" and "School" attributes. When the value of "Age" is 1, it restricts the value of "Literacy" to 2 and "School" to 0.

By taking into account the semantics of the attributes, we have generated a list of possible attribute triplets which can be seen in Table 10. We have discarded other attribute triplets as they cannot be observed such as "*a toddler cannot be literate or occupied and go to a university*".

Triplets are compatible with the semantics of the attributes. For example, when the age range is between 6 and 14, the person can be literate and go to a primary school. In this experiment, we have generated no disguise values. It appears to be a perfectly normal data set, in which no disguises should be found by the algorithm.

Table 10: List of possible attribute triplets

Age	Literacy	School
1	***2***	***0***
2	1	2
2	1	4
2	2	0
2	1	2
2	1	4
2	2	0
3	1	2
3	1	4
3	1	6
3	2	0
3	1	2
3	1	4
3	1	6
3	2	0
4	1	2
4	1	4
4	1	6
4	1	8
4	2	0
4	1	2
4	1	4
4	1	6
4	1	8
4	2	0
5	1	2
5	1	4
5	1	6
5	1	8
5	1	2
5	1	4
5	1	6
5	1	8
5	2	0

3.4.1.2 Experiment:

As we wanted to avoid any bias towards the selection of a particular attribute value in Table 10: List of possible attribute triplets, we initially replicated some tuples until the frequency of each value is almost equal to the frequency of other values for the same attribute. Then, we have randomly selected the tuples from the rule set using Gaussian distribution (except the first tuple, **[1 2 0]**) and constructed our dataset. Afterwards, we have

inserted the **tuple [1 2 0]**. We have always ensured that the size of the data set does not exceed 1000 tuples. The ratio of the tuple **[1 2 0]** changed between 10% and 35% with an increment of 5. (Up to 10% no significant biased result occurred and after the ratio 28% all the results were biased). We have randomly generated 100 sample data sets for each specified ratio.

When the algorithm calculates the disguise value for the attribute "Age", it produces 5 subsets: $T_{v=1}$, $T_{v=2}$, $T_{v=3}$, $T_{v=4}$, $T_{v=5}$. We expect values 4 or 5 to be computed as disguise value because disguise values are expected to be randomly distributed in a data set and these values appear with most of the other attribute values.

As a result of Rule 1, $T_{v=1}$ will consist of ["Literacy"=2 "School=0"]. So it is very important observe in which frequency $T_{v=1}$ will be computed as unbiased sample while measuring effectiveness of CBSQS.

3.4.1.3 Results:

The support of "Age=1" and the number of experiments where the disguise value is computed as 1 is given in the figure below.

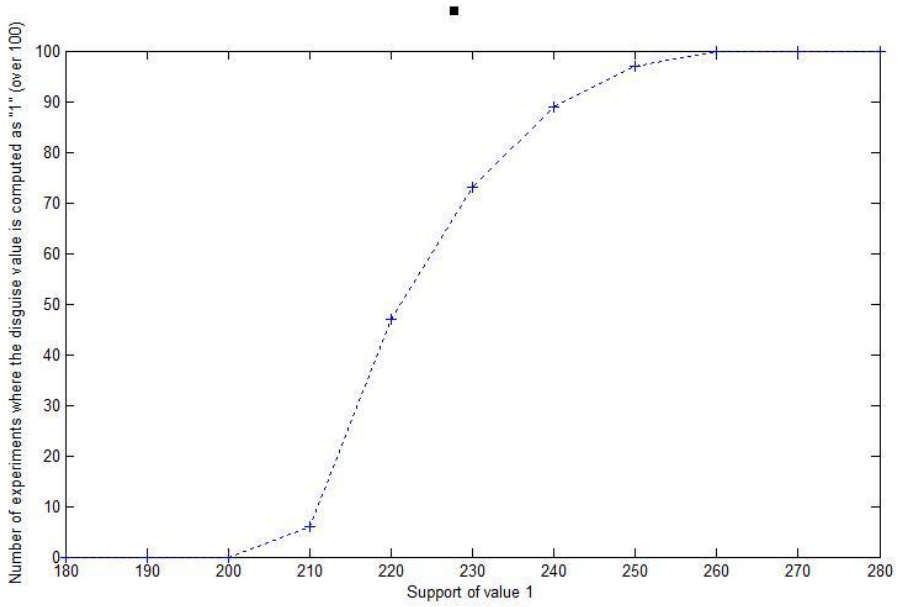

Figure 5: The support versus number of experiments where "Age=1" is found as disguise

As shown in Figure 5, when the support exceeds 200, value 1 starts to be computed as a disguise value. It dominates all other values when the support surpasses 260. This result shows that strong dependency biases the results after a threshold.

According to the proposal, the value 5 and 4 are the most appropriate candidate for the disguised missing values. These values appear in each combination of "Literacy" and "School" attributes, which mean that they are evenly distributed in the data set.

Here are the contingency tables for a sample dataset with 260 tuples following the Rule 1 and its subsets.

Table 11: Contingency table of dataset *D*

Literacy	School=0	School=2	School=4	School=6	School=8
1	0	203	163	128	68
2	438	0	0	0	0

Table 12: Contingency table of $T_{v=1}$

Literacy	School=0	School=2	School=4	School=6	School=8
1	0	0	0	0	0
2	260	0	0	0	0

Table 13: Contingency table of $T_{v=2}$

Literacy	School=0	School=2	School=4	School=6	School=8
1	0	74	40	0	0
2	44	0	0	0	0

Table 14: Contingency table of $T_{v=3}$

Literacy	School=0	School=2	School=4	School=6	School=8
1	0	54	60	58	0
2	44	0	0	0	0

Table 15: Contingency table of $T_{v=4}$

Literacy	School=0	School=2	School=4	School=6	School=8
1	0	47	44	43	47
2	39	0	0	0	0

Table 16: Contingency table of $T_{v=5}$

Literacy	School=0	School=2	School=4	School=6	School=8
1	0	28	19	27	21
2	51	0	0	0	0

It is obvious that the values 4 and 5 in attribute "Age" are seen with each value of each attribute which supports EUS heuristic. But in such a ratio, in all the datasets value 1 dominates the scores and is computed as disguise value.

Since in $\phi(\widetilde{T}, \widetilde{T}')$ formula, only the couples that appear in the subset are included, while working on $T'_{v=1}$ the formula turns out to be;

$$\phi(\widetilde{T}, \widetilde{T'}) = \frac{\mathrm{P}_{\tilde{T}}(\mathrm{Literacy} = 2, \mathrm{School} = 0)}{1 + |\mathrm{CorrT}(2,0) - \mathrm{CorrT}'(2,0)|^2}$$

In a dataset where ratio of people younger than 5 ("Age"=1) is over 20%, P(Literacy=2,School=0) will be directly over '0.20' which is a high nominator. In the denominator, the difference between the correlation in the *D* and correlation in the $T'_{v=1}$ will be computed where the correlation function is defined as;

$$Corr(2,0) = \frac{P(2|0)}{P(0)}$$

Then the correlation will be calculated as 1 for the $T'_{v=1}$ because both the nominator and denominator is 1. So the equation turns out to be;

$$\text{CBSQS} = \frac{P_{\tilde{T}}(\text{Literacy} = 2, \text{School} = 0)}{1+|1-1|} = P_{\tilde{T}}(\text{Literacy}=2, \text{School}=0)$$

So the result of $\phi(\tilde{T}, \tilde{T}')$ will be high enough to select value 1 as disguised missing.

Here the problem arises because of summing up scores of value couples. EUS heuristic can be realized by considering ***how many value couples*** in attributes A1 and A2 are both correlated in a subset T_v and main dataset *D*.

Pima data set [19] as experimented in [1] well demonstrates the problem which contains records about Pima Indian females who are at least 21 year old and tested either positive or negative for diabetes. On the attribute "***diastolic blood pressure***", method based on CBSQS detects 0 as the most frequent disguise value. The result agrees with our domain knowledge. But while analyzing $T_{v=0}$ you see the dependency effect. There are 35 tuples having value 0 in this attribute and each of these tuples have value 0 in the attribute "***2 hour serum insulin***" and 33 of them have 0 in "***triceps skin fold thickness***" In another words;

P(**"*2 hour serum insulin*" = 0)** = 0.487

P("***2 hour serum insulin*" = 0 | "*diastolic blood pressure*" = 0**) = 1 and,

P("***triceps skin fold thickness*" = 0 | "*diastolic blood pressure*" = 0**) = 0.9429.

Similar issues are observed for the attributes "***plasma glucose concentration at 2 hours***" and "***body mass index***". In these attributes values '91' and '0' are detected as disguise values respectively. When we analyzes the projected tables of these values, $T_{\text{Plasma-glucose = 91}}$ and $T_{\text{Body Mass Index = 0}}$, these values are highly dependent with some values of attributes and these projected tables exclude most of the value couples that are correlated in the dataset.

This result means that there may be a dependency between these attributes (regardless of the domain knowledge) and it is mentioned in previous section that dependency is an important aspect while measuring data quality. If value couples, that are found dependent in data profiling, obey the dependency in the whole dataset it is empowers the quality of data. So it is not reasonable to assume these values as missing regardless of the domain knowledge.

3.4.2 CASE 2: Independent Attributes Effect

In Case 1, we discussed the effect of a high VCS in an attribute couple score and demonstrated the effect of a ***single*** high VCS in and the effect of this high attribute couple score in . Such a result suggested us to make experiments to observe in which cases similar biased results are obtained.

We decided to work with datasets in which ***all VCSs*** in a specific ***attribute couple***s are high where other attribute couples return very low scores even zero.

Such situations can appear when the distribution of two attributes in T_v and in *D* are similar but distribution of others are not. For example, in a dataset with four attributes A1, A2, A3 and A4, assume that A2 and A3 are totally independent from each other where A3 take values regardless values of A2. In such a dataset, $\boldsymbol{\phi A2}\&A\boldsymbol{3}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$ returns high score because it is very probable for totally independent value couples to have similar distribution in dataset *D* and subset T_v.

We indicated that distributions of other attribute couples are not similar in our scenario. In such a case, $\boldsymbol{\phi A1}\&A\boldsymbol{2}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$, $\boldsymbol{\phi A1}\&A\boldsymbol{3}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$, $\boldsymbol{\phi A1}\&A\boldsymbol{4}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$, $\boldsymbol{\phi A2}\&A\boldsymbol{4}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$, and $\boldsymbol{\phi A3}\&A\boldsymbol{4}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$ will return low scores. Here we aimed to observe in which cases $\boldsymbol{\phi A2}\&A\boldsymbol{3}(\widetilde{\boldsymbol{T}},\widetilde{\boldsymbol{T}}')$ will dominate.

In order to simulate this issue, we have generated synthetic datasets and observed in which frequencies independence factor dominates the results.

3.4.2.1 Data Set Characteristics:

As we wanted to avoid any bias towards the selection of a particular attribute value in , we initially replicated some tuples until the frequency of each value is almost equal to the frequency of other values for the same attribute. We have generated a dataset comprising 1000 tuples with 4 dimensions indicating “Age”, “Custodian” “School“ and “Marital Status”. Custodian is used to call the person/people who has care or custody for the ones who are younger than 18, it takes the values “parent”, “family member other than parents” or “other”.

Our aim to generate this dataset is obtain a T_v in which two attributes are totally independent and most of the value couples in *D* are present but value of third attribute is fixed. So we can measure the power of independent couples. The value ranges of the attributes and possible valid tuples are defined as given below.

Table 17: Values of "Age"

Value	Age Range
1	0—14
2	15—17
3	Above 18

Table 18: Values of “Custodian

Value	Meaning
1	Self
2	Parent
3	Member of a family other than parent
4	Other

Table 19: Values of "School"

Value	School
1	Primary School
2	Secondary School
3	High School
4	University
5	Ms/PhD

Table 20: Values of "Marital Status"

Value	Meaning
1	Single
2	Married
3	Widow

For the subset of attribute "Age=3", the value of "Custodian" is fixed to 'self' but "School" and "Marital Status" take all the values independently. Because for the people who are self custodian, there is no age restriction so dependency between "Marital Status" and "School" caused by "Age" disappears.

In the light of these assumptions, we have generated a list of possible tuples which can be observed in a data set in .

Table 21: List of possible tuples

Rule ID	Age	Custodian	School	Marital Status
Rule: 1	1	2	1	1
Rule: 2	1	2	2	1
Rule: 3	1	3	1	1
Rule: 4	1	3	2	1
Rule: 5	1	4	1	1
Rule: 6	1	4	2	1
Rule: 7	2	2	1	1
Rule: 8	2	2	1	2
Rule: 9	2	2	1	3
Rule: 10	2	2	2	1
Table 22: List of possible tuples (Cont.)				
Rule: 11	2	2	2	2
Rule: 12	2	2	2	3
Rule: 13	2	2	3	1
Rule: 14	2	2	3	2
Rule: 15	2	2	3	3

Rule: 16	2	3	1	1
Rule: 17	2	3	1	2
Rule: 18	2	3	1	3
Rule: 19	2	3	2	1
Rule: 20	2	3	2	2
Rule: 21	2	3	2	3
Rule: 22	2	3	3	1
Rule: 23	2	3	3	2
Rule: 24	2	3	3	3
Rule: 25	2	4	1	1
Rule: 26	2	4	1	2
Rule: 27	2	4	1	3
Rule: 28	2	4	2	1
Rule: 29	2	4	2	2
Rule: 30	2	4	2	3
Rule: 31	2	4	3	1
Rule: 32	2	4	3	2
Rule: 33	2	4	3	3
Rule: 34	3	1	1	1
Rule: 35	3	1	1	2
Rule: 36	3	1	1	3
Rule: 37	3	1	2	1
Rule: 38	3	1	2	2
Rule: 39	3	1	2	3
Rule: 40	3	1	3	1
Rule: 41	3	1	3	2
Rule: 42	3	1	3	3
Rule: 43	3	1	4	1
Rule: 44	3	1	4	2
Rule: 45	3	1	4	3
Rule: 46	3	1	5	1
Rule: 47	3	1	5	2
Rule: 48	3	1	5	3

Tuples are compatible with the semantics of the attributes. For example Rule 38 shows that, when the age is greater than 18, the person can be married and can be graduated from secondary school.

In this rule set we wanted to detect in which frequency value 3 for the attribute “Age” is computed as disguise value. So in the first part of generation, we have randomly selected the

tuples from the rules Rule1-Rule 33. Afterward we inserted these tuples from the rules Rule 34-Rule 48 in specific ratios.

We have selected these tuples from Rule: 1 – Rule: 33 based on normal random distribution. We selected the rules for a dataset and shuffled the rules before selecting the rules for the second dataset. So we achieved to work with as different datasets as possible.

After generating first part of the dataset, we have inserted the tuples that follow the Rules 34-Rules 48. We have generated 100 sample data sets for each ratio range between 10% and 40% with an increment of 10 which are observed as boundary ratios for significant results.

3.4.2.2 Experiment:

When the algorithm calculates the disguise value for the attribute Age, 3 subsets will be analyzed; $T_{v=1}$, $T_{v=2}$, $T_{v=3}$. This time the subsets consists of attributes "Custodian", "School" and "Marital Status".

We expect value 2 to be computed as disguise value because disguise values are expected to be randomly distributed in a data set and this value appears with most of the other attribute values. It does not appear in very few cases but it can be ignored since the majority of the cases includes.

As a result of the semantics of the attributes, "School" and "Marital Status" take values independent of each other in $T_{v=3}$. So these two attributes will represent a good sample of the dataset and score of them in $\phi A2\&A3(\widetilde{T},\widetilde{T'})$ will be high. But the score of "Custodian" with "Marital Status", $\phi A2\&A4(\widetilde{T},\widetilde{T'})$, and score of "Custodian" with "School", $\phi A2\&A3(\widetilde{T},\widetilde{T'})$ will be low because value of "Custodian" is fixed to 1 so $T_{v=3}$ will exclude many value couples of dataset.

3.4.2.3 Results:

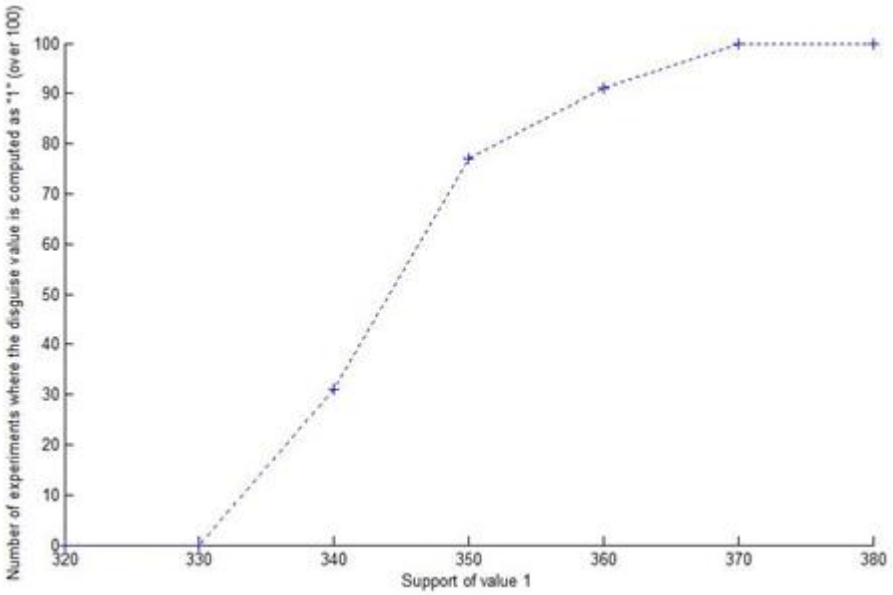

Figure 6: The support versus number of experiments where "Age=3" is found as disguise

As seen above, when the support of "Age=3" exceeds the support 330 over 1000, it starts to be computed as disguise and it dominates the others in each experiments after the support 370. This result shows that when the support of value 3 exceeds the support of other values; 1 and 2, although $T_{v=3}$ does not support EUS heuristic, $\phi A2\&A3(\widetilde{T},\widetilde{T}')$ can dominate the summation in .

The contingency table of a sample dataset *D*, which is composed of 370 tuples where "Age"=1, and its subsets are demonstrated below to clarify the point.

Table 23: Contingency table of *D* between "Custodian" and "School"

Custodian	School=1	School=2	School=3	School=4	School=5
1	80	66	69	80	75
2	99	92	36	0	0
3	86	76	26	0	0
4	75	88	52	0	0

Table 24: Contingency table of *D* between "Custodian" and "Marital status"

Custodian	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	134	78	158
2	158	30	39
3	129	126	33
4	127	45	43

Table 25: Contingency table of *D* between "School" and "Marital status"

School	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	230	49	61
2	213	51	58
3	63	48	72
4	25	17	38
5	17	14	44

Table 26: Contingency table of $T_{v=1}$ between "Custodian" and "School"

Custodian	School=1	School=2	School=3	School=4	School=5
1	0	0	0	0	0
2	65	51	0	0	0
3	51	52	0	0	0
4	47	44	0	0	0

Table 27: Contingency table of $T_{v=1}$ between "Custodian" and "Marital status"

Custodian	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	0	0	0
2	116	0	0
3	103	0	0
4	91	0	0

Table 28: Contingency table of $T_{v=1}$ between "School" and "Marital status"

School	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	163	0	0
2	147	0	0
3	0	0	0
4	0	0	0
5	0	0	0

Table 29: Contingency table of $T_{v=2}$ between "Custodian" and "School"

Custodian	School=1	School=2	School=3	School=4	School=5
1	0	0	0	0	0
2	34	41	36	0	0
3	35	24	26	0	0
4	28	44	52	0	0

Table 30: Contingency table of $T_{v=2}$ between "Custodian" and "Marital status"

Custodian	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	0	0	0
2	42	30	39
3	26	26	33
4	36	45	43

Table 31: Contingency table of $T_{v=2}$ between "School" and "Marital status"

School	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	29	34	34
2	40	37	32
3	35	30	49
4	0	0	0
5	0	0	0

Table 32: Contingency table of $T_{v=3}$ between "Custodian" and "School"

Custodian	School=1	School=2	School=3	School=4	School=5
1	80	66	69	80	75
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Table 33: Contingency table of $T_{v=3}$ between "Custodian" and "Marital status"

Custodian	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	134	78	158
2	0	0	0
3	0	0	0
4	0	0	0

Table 34: Contingency table of $T_{v=3}$ between "School" and "Marital status"

School	Marital Status = 1	Marital Status = 2	Marital Status = 3
1	38	15	27
2	26	14	26
3	28	18	23
4	25	17	38
5	17	14	44

Given in tables Table 32 and Table 33, the subset $T_{v=3}$ excludes many value couples and because of independency between "School" and "Marital status" when the value of "Age" is 3, Table 34 contains all of the value couples. So while computing CBSQS for $T_{v=3}$ $\phi A3\&A4(\widetilde{T},\widetilde{T'})$ can dominate the low scores of $\phi A2\&A3(\widetilde{T},\widetilde{T'})$ and $A2\&A4(\widetilde{T},\widetilde{T'})$.

Recall that EUS heuristic fails if we want to compute a disguise value for an independent attribute. But this result shows that it also fails while working on a dependent attribute if the rest of the dataset includes independent attributes.

This result highlights the importance of considering ***how many attribute couples*** resemble the distribution of the dataset and avoiding success of an attribute couple to hide the failure of another.

3.4.3 CASE 3: Derived Attributes Effect

In this case we will analyze the power of ***derived attribute couples,*** where the value of A1 is derived from the value of A2, in the summation.

In [8], it has been stated that values of derived attributes are calculated based on the values of some other attributes. This approach is very common when the calculation is rather complex and stores data stored in multiple records of multiple entities. Performing the calculation on the fly is then very inefficient.

In a dataset which includes derived attributes A1 and A2 where value of A2 is derived from value of A1, a value of A2 will be observed with a certain value of A1 because of derivation. So while computing $\phi(\widetilde{T}, \widetilde{T}')$, $\phi A1\&A2(\widetilde{T}, \widetilde{T}')$ will be very high. Similar to problems discussed in case 1 and 2, such a high score can lead to high total score although other attributes are not correlated.

3.4.3.1 Data Set Characteristics:

In order to simulate the issue, we have generated a synthetic census dataset with 4 attributes; “Age“, “Number of Estate”, “Income” and “Tax”. Here, “Income” and “Tax” are derived attributes where the value of “Tax” is derived from value of “Income”.

In our dataset we included employees in different ages. “Income” refers to earnings of the employee per month and “Tax” refers to tax paid per month. If an employee has no estate, the value of “Tax” is directly 10% of the value of “Income”. For the employees who have estate(s), value of “Tax” is greater than 10% of the “Income”. The value ranges of attributes and list of valid tuples are given below.

Table 35: Values of “Age”

Value	Age Range
1	6--14
2	15--17
3	18--24
4	Above 24

Table 36: Values of "Number of Estate"

Value	Number of Estate Range
0	0
1	1
2	2
3	3
4	4
5	Above 4

Table 37: Values of "Income"

Value	Income Range
1	500-1000 YTL
2	1100-2000 YTL
3	2100-3000 YTL
4	3100-4000 YTL
5	Above 4000 YTL

Table 38: Values of "Tax"

Value	Tax Range
1	50-100 YTL
2	110-200 YTL
3	210-300 YTL
4	310-400 YTL
5	Above 400 YTL

Table 39: List of possible tuples

RULE ID	Age	#Estate	Income	Tax
Rule 1	2	0	1	1
Rule 2	2	0	2	2
Rule 3	2	0	3	3
Rule 4	2	0	4	4
Rule 5	2	0	5	5
Rule 6	3	0	1	1
Rule 7	3	0	2	2
Rule 8	3	0	3	3

Table 40: List of possible tuples (Cont.)				
Rule 9	3	0	4	4
Rule 10	3	0	5	5
Rule 11	4	0	1	1
Rule 12	4	0	2	2
Rule 13	4	0	3	3
Rule 14	4	0	4	4
Rule 15	4	0	5	5
Rule 16	4	1	1	2
Rule 17	4	1	2	3
Rule 18	4	1	3	4
Rule 19	4	1	4	5
Rule 20	4	1	5	5
Rule 21	4	2	1	2
Rule 22	4	2	2	3
Rule 23	4	2	3	4
Rule 24	4	2	4	5
Rule 25	4	2	5	5
Rule 26	4	3	1	2
Rule 27	4	3	2	3
Rule 28	4	3	3	4
Rule 29	4	3	4	5
Rule 30	4	3	5	5
Rule 31	4	4	1	2
Rule 32	4	4	2	3
Rule 33	4	4	3	4
Rule 34	4	4	4	5
Rule 35	4	4	5	5
Rule 36	4	5	1	2
Rule 37	4	5	2	3
Rule 38	4	5	3	4
Rule 39	4	5	4	5
Rule 40	4	5	5	5

Tuples are compatible with the semantics of the attributes. For example, children who are between 6 and 17 years old cannot involve in transactions about deeds alone but require a custodian. Also children younger than 15 cannot have jobs.

3.4.3.2 Experiment:

As can be seen in
Table 39, the employees younger than 17 has no estate. So in the tuples where "Age" is 2, and 3, such a dependency occurs;

(Age=2 |Age=3) → Income = 1 →Tax = 1

Income = 2 →Tax = 2

Income = 3 →Tax = 3

So while computing disguise value for the attribute "Age", $T_{v=2}$ and $T_{v=3}$ exclude most of the value couples of "Income" and "Tax" (("Income"=1, "Tax"=2), ("Income"=1, "Tax"=3), ("Income"=1, "Tax"=4), ("Income"=1, "Tax"=5), ("Income"=2, "Tax"=3), ("Income"=2, "Tax"=4), ("Income"=2, "Tax"=5), ("Income"=3, "Tax"=4), ("Income"=3, "Tax"=5), ("Income"=4, "Tax"=5)). Such a circumstance breaks the EUS heuristic. But because of the dependency, as discussed in 3.4.1 CASE 1: Dependency Effect, $\phi A3\&A4(\widetilde{T},\widetilde{T'})$ will return high scores in these samples which may lead a high CBSQS.

Value 4 is the randomly distributed value in the attribute "Age". So we expect this value to be computed as disguise value which supports EUS heuristic. We aimed to observe success of EUS to capture this value despite the dependency in projected databases of other values 1, 2 and 3.

We have selected the tuples that follow Rule: 11 to Rule: 45 randomly and generated the first form of the dataset. So we formed a dataset including the tuples with "Age=4". Afterwards we have selected the tuples that follow Rule1-Rule 10 and injected into the dataset in specific ratios. So we aimed to observe the ratio in which the values 2 or 3 can dominate the randomly distributed value 4.

We have done the insertion in specific ratios: Number of tuples where "Age"=2 and "Age"=3 varied between [550:700] with an increment of 10. We used these boundaries because no change has been observed when the number of tuples are out of this range.

Here our aim is to detect in which ratios the dependency between "Incomes" and "Tax" will favor the result. Again we generated and computed 100 datasets for each ratio.

3.4.3.3 Results:

Results are displayed in Figure 7.

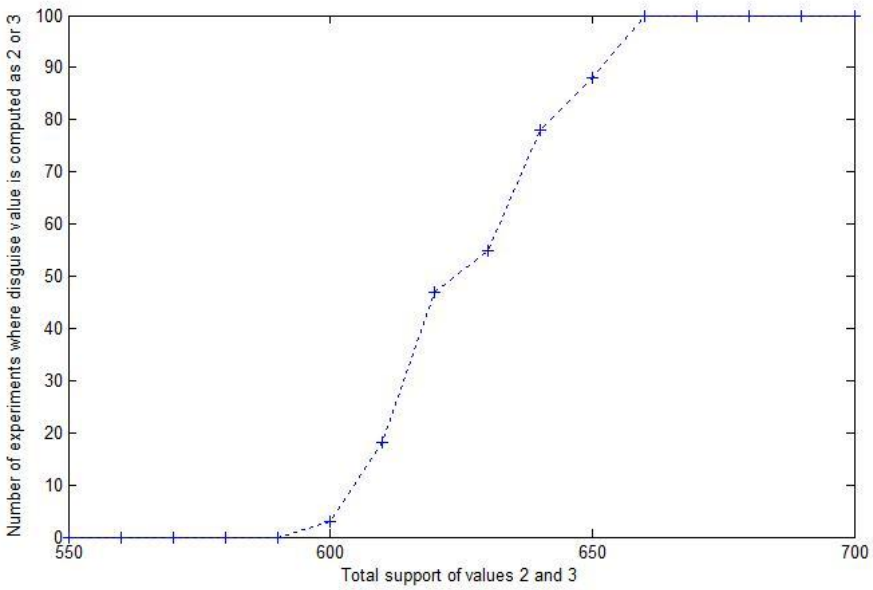

Figure 7: The support versus total number of experiments where "Age=2" or "Age=3" is found as disguise

It is observed that when the total number of these tuples exceeds 600, where approximately 300 tuples appear per each interval, values 2 and 3 start to be computed as disguise missing values and after 660, none of the experiments return 4.

In order to clarify the point, the contingency table for a dataset *D* which includes 340 tuples with "Age"=4 and its subsets are given below;

Table 41: Contingency table of *D* between "# Estate" and "Income"

# Estate	Income =1	Income =2	Income =3	Income =4	Income=5
0	161	142	115	147	144
1	10	4	17	16	11
2	15	14	16	16	12
3	4	15	10	13	27
4	11	15	8	11	8
5	8	3	8	5	14

Table 42: Contingency table of *D* between "# Estate" and "Tax"

# Estate	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
0	161	142	115	147	144
1	0	10	4	17	27
2	0	15	14	16	28
3	0	4	15	10	40
4	0	11	15	8	19
5	0	8	3	8	19

Table 43: Contingency table of *D* between "Income" and "Tax"

Income	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
1	161	48	0	0	0
2	0	142	51	0	0
3	0	0	115	59	0
4	0	0	0	147	61
5	0	0	0	0	216

Now, let's continue with $T_{v=2}$

Table 44: Contingency table of $T_{v=2}$ between "# Estate" and "Income"

# Estate	Income =1	Income =2	Income =3	Income =4	Income=5
0	62	53	54	100	75
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Table 45: Contingency table of $T_{v=2}$ between "# Estate" and "Tax"

# Estate	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
0	62	53	54	100	75
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Table 46: Contingency table of $T_{v=2}$ between "Income" and "Tax"

Income	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
1	62	0	0	0	0
2	0	53	0	0	0
3	0	0	54	0	0
4	0	0	0	100	0
5	0	0	0	0	75

The results of Tv=2 and Tv=3 are similar. Because they follow the similar rules and none of them are unbiased sample of *D*.

Table 47: Contingency table of $T_{v=3}$ between "# Estate" and "Income"

# Estate	Income =1	Income =2	Income =3	Income =4	Income=5
0	78	84	52	41	61
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Table 48: Contingency table of $T_{v=3}$ between "# Estate" and "Tax"

# Estate	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
0	78	84	52	41	61
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Table 49: Contingency table of $T_{v=3}$ between "Income" and "Tax"

Income	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
1	78	0	0	0	0
2	0	84	0	0	0
3	0	0	52	0	0
4	0	0	0	41	0
5	0	0	0	0	61

After these biased samples which are computed as unbiased according to CBSQS method let's analyze Tv=4 which is not computed as unbiased sample in any of 100 experiments given the datasets with the same ratios.

Table 50: Contingency table of *D* between "# Estate" and "Income"

# Estate	Income =1	Income =2	Income =3	Income =4	Income=5
0	21	5	9	6	8
1	10	4	17	16	11
2	15	14	16	16	12
3	4	15	10	13	27
4	11	15	8	11	8
5	8	3	8	5	14

Table 51: Contingency table of *D* between "# Estate" and "Tax"

# Estate	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
0	21	5	9	6	8
1	0	10	4	17	27
2	0	15	14	16	28
3	0	4	15	10	40
4	0	11	15	8	19
5	0	8	3	8	19

Table 52: Contingency table of *D* between "Income" and "Tax"

Income	Tax =1	Tax =2	Tax =3	Tax =4	Tax=5
1	21	48	0	0	0
2	0	5	51	0	0
3	0	0	9	59	0
4	0	0	0	6	61
5	0	0	0	0	80

You see that $T_{v=4}$ covers most of the value couple in *D*. But in this ratio none of the dataset returns 4 as disguise value.

In this experiment, it is observed that a derived attribute in which all the values strictly depend on the values of other attribute return high in method $\phi(\widetilde{T}, \widetilde{T}')$ which biases the results.

As a result, it is essential to avoid high score of a value or attribute couple to favor the result and a better algorithm must be handled that avoids such dominations.

CHAPTER 4

PROPOSED METHOD

In the current approach [1], sample quality is calculated using $\phi(\widetilde{T}, \widetilde{T}')$ which returns non-negative number. It is based on the summation of score of attribute couples and score of attribute couples are calculated by summing up the score of value couples without any normalization. We analyzed the possible conditions that cause biased results and explained the results.

In order to eliminate the deficiency of the current approach, we have decided to redesign the $\phi(\widetilde{T}, \widetilde{T}')$.

We focused on two issues;

1. Score of a value couple must not dominate the score of other couples while computing score of an attribute couple.
2. Score of an attribute couple must not dominate the score of other attribute couples.

In the light of this information, we have decided to use such a formula, to measure sample quality score, that returns a score in a specific interval for an attribute couple unlike $\phi(\widetilde{T}, \widetilde{T}')$. So when scores of attribute couples added together, it is impossible for one to dominate the others. In order to guarantee that scores of dependent value couples do not bias the results, we preferred to use a distribution hypothesis tests which are not sensitive to such cases that $\phi(\widetilde{T}, \widetilde{T}')$ fails.

We decided on **Chi Square Two Sample Test** which checks whether two data samples come from the same distribution without specifying what that common distribution is. The chi-square two sample test is based on binned data. Binning for both data sets should be the same. The basic idea behind the chi-square two sample test is that the observed number of points in each bin (this is scaled for unequal sample sized) should be similar if the two data samples come from common distributions. More formally, the chi-square two sample test statistic can be defined as follows.

H0: The two samples come from a common distribution.

Ha: The two samples do not come from a common distribution.

Test Statistic: For the chi-square two sample tests, the data is divided into k bins and the test statistic is defined as:

$$x^2 = \sum_{i=1}^{k} \left(\frac{(K_1 R_i - K_2 S_i)^2}{R_i + S_i} \right)$$

where the summation is for bin 1 to k, R_i is the observed frequency for bin i for sample 1, and S_i is the observed frequency for bin i for sample 2. K_1 and K_2 are scaling constants that are used to adjust for unequal sample sizes. Specifically,

$$K_1 = \sqrt{\frac{\sum_{i=1}^{k} S_i}{\sum_{i=1}^{k} R_i}}$$

$$K_2 = \sqrt{\frac{\sum_{i=1}^{k} R_i}{\sum_{i=1}^{k} S_i}}$$

This test is sensitive to the choice of bins. Most reasonable choices should produce similar, but not identical, results [20].

Our aim is to measure the distribution similarity between attribute couples of the dataset and the subset. In order to achieve this, we decided to represent the dataset as a means of value couples they include. So, for attributes A1 and A2 we classified the value pairs that appear together and generated a new column indicating the class the value couples belong to. We did this data transformation for every attribute couple and for a dataset of n attributes we generated a new dataset containing $\binom{n}{2}$ attributes.

As statistical toolbox, we used Multivariate Analysis Toolbox for Matlab® written by Liran Carmel [21].

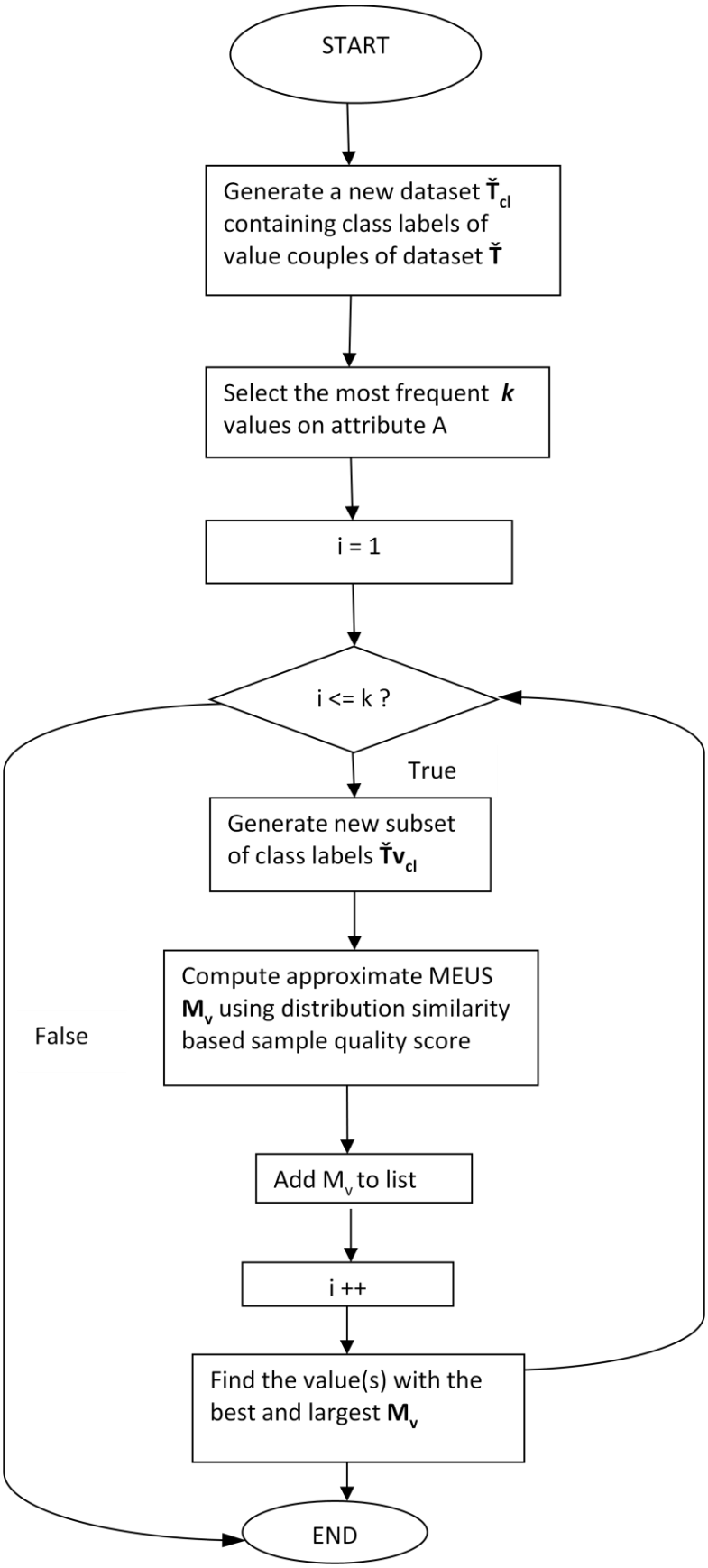

Figure 8: Main Framework

The detail of the transformation is displayed in Figure 9 and details of sample quality method in Figure 10.

Input: Dataset T with 2 attributes A1, A2
Output: Column C with 1 attribute A1A2
Method:

1. List the value couples that appear in T
2. FOR EACH tuple $t \in T$

   Classify *t* and insert the class to *C*
   END
3. RETURN C;

Figure 9: A method to generate transformed column

While generating transformed dataset each column couple is sent to the method given in Figure 9. In case user wants to wants to transform columns triple instead of couples, program also allows classifying the triples and transforming each triplet to a column. Similarly, it allows quartets or more columns. So nonlinear relations can also be captured, like XOR, which is impossible in CBSQS.

Input: T_c, U_c, attribute ***a*** which is being analyzed for disguise missing value
Output: Sample quality score *sqs*

1. *sqs=0*
2. FOR EACH attribute couple A1&A2 $\in U_c$
   (A1!=***a*** && A2!=***a***)
   Generate class label column $\mathbf{Cl_1}$ for A1A2 in T_c ;
   Generate class label column $\mathbf{Cl_2}$ for A1A2 in U_c ;
   Measure the distribution similarity ***s*** between Cl_1 and Cl_2 using "**Chi Square Two Sample Test**"
   *sqs=sqs+**s**;*
3. END
4. RETURN *sqs*

Figure 10: A method to compute sample quality

We kept the approach of computing approximate MEUS by removing the tuples until the sample quality score cannot increase. The only difference is the input of the method. Details are given in Figure 11.

Phase 1: Classifying the value couples in dataset T and generating new dataset T_c
Input: T_c, value *v* on attribute *A*
Output: approximate M_v

1. $U \leftarrow T_v$
2. Classify U and generate U_c
3. REPEAT
4. FOR EACH tuple $\tilde{t} \in U_c$
   compute DV-score gain of $(U_c - \{\tilde{t}\})$ over U_c;
   remove a tuple $\tilde{t}_0$ with the largest DV-score gain if the gain is positive;
5. UNTIL no tuple can be removed;
6. RETURN U;

Figure 11: A method to compute approximate MEUS

4.1 Experimental Results

We have tested our new approach in different cases using real datasets and synthetic datasets. The first experiment was conducted on Pima Indians Diabetes data set [19], which was also used in Ming…et.al [1]. The results are tabulated in Table 53.

Table 53: The comparison between CBSQS and chi-square sample test approaches

	Ming Hua.et.al.'s Approach based on CBSQS		**Our approach**	
ATTRIBUTE	**Most Frequent Disguise Value**	**Number of tuples in the approximate MEUS**	**Most Frequent Disguise Value**	**Number of tuples in the approximate MEUS**
Number of times pregnant	0	110 / 111	0	110 / 111
Plasma glucose concentration at 2 hours	91	9 / 9	100	16 / 17
Diastolic blood pressure (mm Hg)	0	35 / 35	70	55 / 57
Triceps skin fold thickness (mm)	0	227 / 227	0	226 / 227
2-Hour serum insulin (mu U/ml)	0	374 / 374	0	373 / 374
Body mass index (weight in kg/(height in m)^2)	0	11 /11	32	12 / 13
Diabetes pedigree function	No	No	No	No
Age (years)	21	57 / 63	21	62 / 63

The first difference appears in the third attribute "Diastolic blood pressure (mm Hg)". Our method detects the value "70" as disguise missing. Recall that we have discussed that issue in Case 2.1 and underlined that while computing approximate MEUS for this attribute, there are 35 tuples having value 0 in this attribute and each of these tuples have value 0 in the attribute "***2 hour serum insulin***" and 33 of them have 0 in "***triceps skin fold thickness***".

It is stated in the paper that the value 70 which is the normal blood pressure may be correlated with some other attribute values for a person in good health, which makes the value 70 not evenly distributed. In order to clarify the point, we have consulted to a medical doctor and learnt that while this can be true, it does not necessarily mean that the normal blood pressure should always be observed within the normal range of the rest of the attributes. There can be other attributes such as whether the person is taking any medication which may affect the blood pressure but not included in the data set.

We also tested our approach in the datasets in which the current approach fails; datasets that include dependent value couples, datasets that include random attributes and datasets that include derived attributes.

4.1.1 CASE 1: Dependency Effect

We used the same valid tuples given in Table 10 and generated the dataset following the same steps given in 3.4.1 CASE 1: Dependency Effect in 3.4. The results of experiments are displayed below. Dotted line represents the results of CBSQS and solid line represents the results of our methodology.

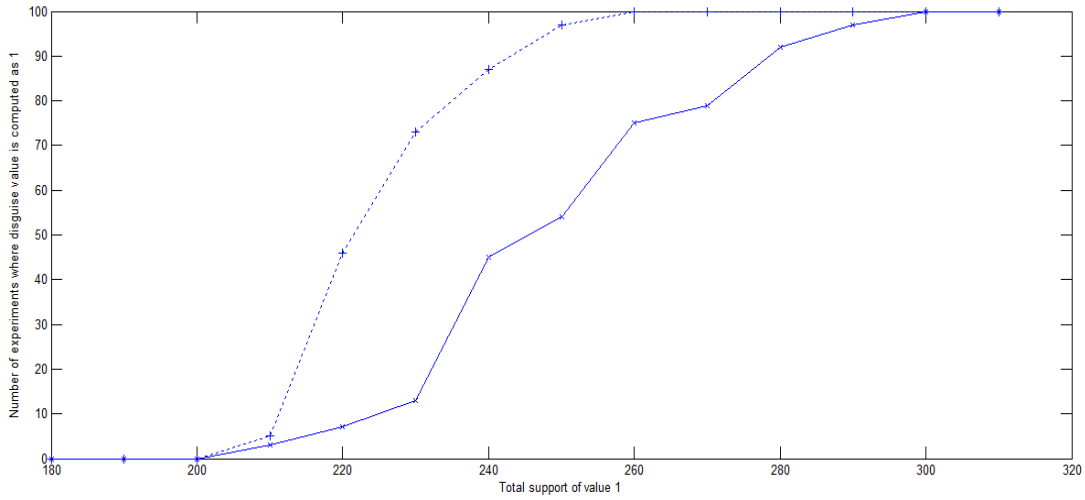

Figure 12: The support versus total number of experiments where "Age=1" is found as disguise using our approach

Recall that after a ratio of 26% value "1" dominates all other values of "Age" and is computed as disguise value in the current approach. But in our approach it shifts to 30% which explains that our approach is less sensitive to dependency.

4.1.2 CASE 2: Independent Attribute Effect

We used the same valid tuples given in and generated the dataset following the same steps given in 3.4.2 CASE 2: Independent Attributes Effect in 3.4. The results of experiments are displayed below. Dotted line represents the results of CBSQS and solid line represents the results of our methodology.

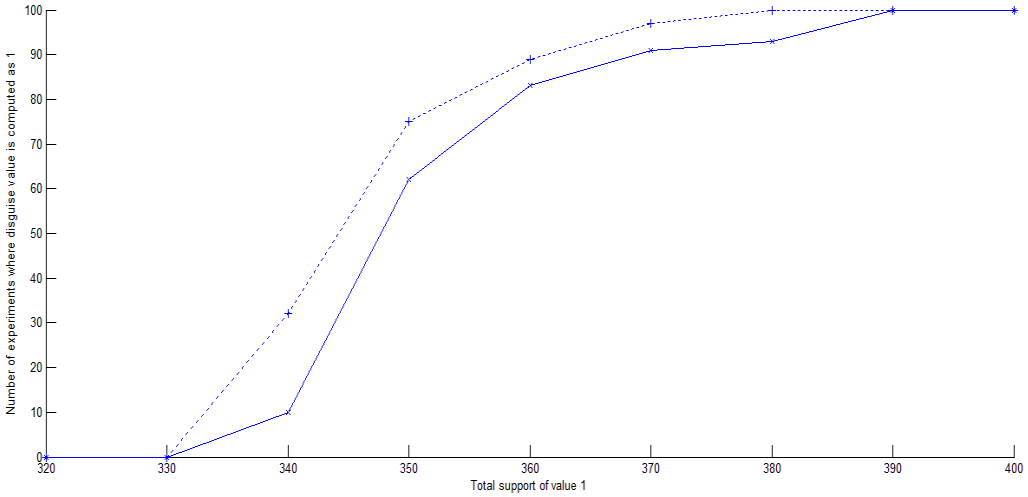

Figure 13: The support versus total number of experiments where "Age=3" is found as disguise

In CBSQS based algorithm, recall that when the support of "Age=3" exceeds the ratio 33% approximately in 30 out of 100 trials, the value 3 is computed as disguise value but in our method this number decreases to 10 trials. Also in the current approach when the ratio of tuples where Age=3 is 37%, this value dominates all other scores and computed as disguise value in each of the 100 datasets. This ratio increases to 39% in our method.

These results display that random attributes are the most important challenge while computing disguise values in a dataset. Our approach makes the results a bit better but it is not reasonable to detect disguise values in a dataset which includes totally independent attributes. Detecting and eliminating these independent attributes may solve the problem.

4.1.3 CASE 3: Derived Attribute Effect

We used the same tuples given in
Table 39 and generated the dataset following the same steps given in 3.4.3 CASE 3: Derived Attributes Effect in 3.4. The results of experiments are displayed below. Dotted line represents the results of CBSQS and solid line represents the results of our methodology.

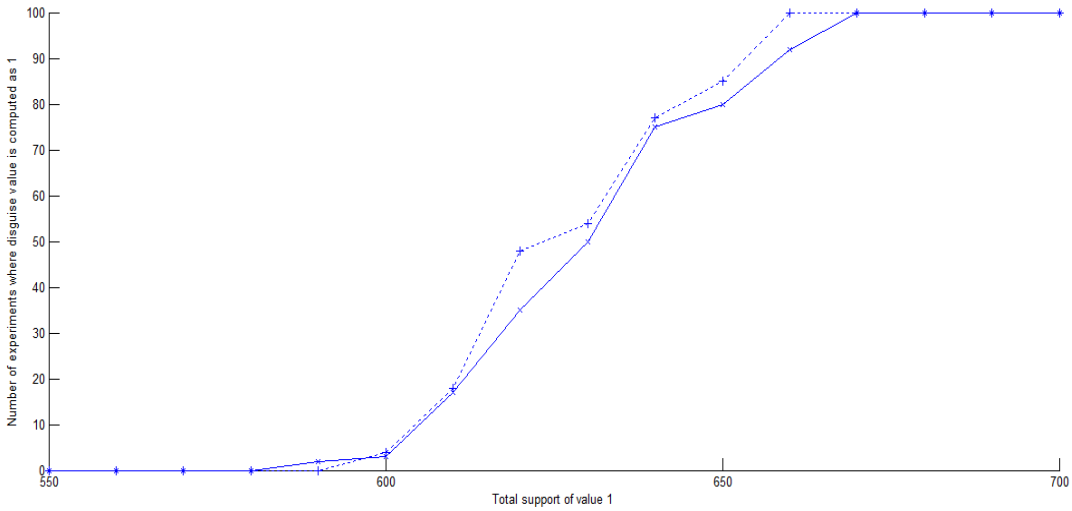

Figure 14: The support versus total number of experiments where "Age=1" is found as disguise

This result displays that domination in (summation of scores of value couples while computing attribute couple score) can be handled using our approach. But domination in (summation of scores of attribute couples) still causes problems.

CHAPTER 5

CONCLUSION

In this thesis we have focused on detecting disguise missing data. In particular, we have analyzed the approach proposed by Ming Hua and Jian Pei [1] in terms of its deficiencies and capabilities and proposed an improvement based on chi-square two sample test.

In this approach [1], the value whose projected database is the unbiased sample of whole dataset is computed as disguise missing value. In order to measure whether a set of tuples is an unbiased sample of a dataset, a method called *correlation-based sample quality score, CBSQS*, is used which is based on joint probability and correlation differences. Because it is assumed that if values that are correlated in a dataset are also correlated in its subset, the subset can be considered as an unbiased sample of the dataset.

The main idea in CBSQS is to calculate the score of each value couple and to sum them up to lead a score of an attribute couple. Finally scores of attributes couples are summed to lead total score which is used to detect how representative a subset is.

In this thesis, we focused on two main issues: challenges in summing up value couple scores to get an attribute couple score and challenges in summing up attribute couple scores to get total score.

The actual expectation behind the summation in *CBSQS* is the idea that summation of high values results in a high score and summation of low values results in a low score. So score of an *attribute couple* is expected to be low when score of each *value couple* is low and *total score* is expected to be high when score of each *attribute couple* is high.

But in reality many cases may emerge which do not meet aforementioned expectations. When a summation includes many very low values but only one very high value, the result can still be relatively high value that contradicts with the expectation.

As a consequence, we focused on conditions which affect the summation function and handled the problem as a first issue.

Recall that CBSQS is based on joint probability. So *dependent values* that are frequently observed together will have very high value couple scores. Even though rest of the value couples for a specific attribute couple returns very low scores, sum of them may still be high

because of just one very high score. In order to clarify our argument we generated synthetic datasets which include dependent attributes and displayed the biased results and explained reasons.

Secondly we worked on summation of attribute couple scores and figured out the cases where an attribute couple may return such a high score that can dominate the low scores of other attribute couples. We pointed out two cases: *totally independent attributes* and *derived attributes*.

When an attribute A1 is totally independent with another attribute A2, their score in CBSQS, will be very high because it is very probable for totally independent value couples to have similar distribution in dataset and its subset. Therefore, even the rest of the attributes return low scores, the total score will manage to be relatively high.

If an attribute A1 is a derived attribute of A2, than specific values of A1 will be observed with specific values of A2 which will yield a high attribute couple score because values of these attributes are highly correlated. So even the other attribute couples return low scores, score of A1 and A2 will be very high and will be able to manage to dominate the other low scores.

In the light of observed deficiencies, we generated a new methodology to measure sample quality score based on *chi-square two sample tests* which checks whether two data samples come from the same distribution without specifying what that common distribution is. In this methodology, we represented the dataset as a means of value couples they include. So, for attributes A1 and A2 we classified the value couples that appear together and generated a new column A1A2 indicating the class the value couples belongs to. At the end of generation, we had a new data set containing $\binom{n}{2}$ attributes that hold value classes of value couples. After this process, we measured the similarity of distributions in the main dataset and subsets using *chi-square two sample tests.*

In order to measure score of an attribute couple we tested the distribution of generated columns indicating the class labels of attributes A1 and A2 in main dataset and the subset. So summation effect in the calculation of value couple score is directly eliminated. However, while computing total score of CBSQS, we kept summing up attribute couple scores since a better algorithm has not been implemented yet.

We showed that results of the experiments on real datasets (Pima Indian Diabetes [19] Dataset and AERS Data [23]) compare favorable with CBSQS. Our method computationally performs better than CBSQS. Results show that our approach solves the deficiency in dependent value couples because calculation of attribute couple score is not dependent on summation. Another improvement is that value that returns from chi-square two sample tests is either 0 or 1. So each attribute couple score is 0 or 1 which eliminates the domination

effect. But summation can still cause problems. So we had *minor* improvements in experiments in which data sets include derived or totally independent values.

In the future, we aim to implement a new algorithm which is not based on summation of scores. When we get rid of this summation effect, we want to clean a dataset from detected disguise missing values based on CBSQS and then apply a data mining technique. We are planning to clean the disguise missing values detected based on our methodology in the same dataset and apply the same data mining techniques and compare the results.

We will investigate other two sample tests and compare their effectiveness. One of our aims is to generate a framework which computes whether EUS heuristic fits a dataset that can be used before rushing into computing disguise missing data for the dataset and also in which ignorable attributes can be eliminated which provides computational efficiency.

REFERENCES

[1] Hua Ming, Jian Pei., "Cleaning Disguised Missing Data: A Heuristic Approach." California : KDD' 07, 2007.

[2] "Quality, Data, and Data Quality." [book auth.] Thomas C. Redman. *Data Quality for the Information Age.* London : Artech House, 1996, p. 19.

[3] Jason D. Van Hulse, Taghi M. Khoshgoftaar, Haiying Huang., "The Pairwise Attribute Niose Detection Algorithm." London : Springer, 2006, Issue Konowledge and Information Systems.

[4] Richard D. De Veaux, David J. Hand., "How to Lie with Bad Data." Massachusettes : Statistical Science, 2005, Vol. 20.

[5] Jochen Hipp, Ulrich Güntzer, Udo Grimmer., "Data Quality Mining: Making a Virtue Neccessity." Tübingen : 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2001.

[6] Dasu T., Johnson T., *Exploratory Data Mining and Data Cleaning.* New York : Wiley-Interscience, 2003.

[7] Berti-Equille, Laure., "Data Quality Awareness: A case study for cost-optimal association rule mining." London : Springer, 2006.

[8] "Data Quality for Practitioners: Data Quality Assessment." [book auth.] Arkady Maydanchik. 2007, pp. 146-148.

[9] Stuart Madnick, Hongwei Zhu., "Improving Data Quality Through Effective Use of Data Semantics." Cambridge : ScienceDirect, 2005.

[10] Olson, Jack E., *Data Quality: The Accuracy Dimesion.* San Francisco : Morgan Kaufmann, 2003. P:47.

[11] Cinzia Cappiello, Chiara Francalanci, Barabara Pernici., "Data Qaulity Assessment form the User's perspective." Paris : ACM, 2004.

[12] "Inaccurate Customer Data Results In Lost Revenue." *Experian QAS.* [Online] 06 09, 2005. http://www.qas.com/display-news.htm?id=4731.

[13] STATE UNIV OF NEW YORK AT ALBANY., "Data Quality Tools for Data Warehousing- A Small Sample Survey." Albany : Defense Technical Information Center, 1998.

[14] Markets, Independent research by Dynamic., *How well do you know your customers?* s.l. : Experian Press, 2005.

[15] Leo L. Pipino, Yang W. Lee, Richard Y. Wang., "Data Quality Assessment." New York : ACM, 2002, Issue 4, Vol. 45. 0001-0782.

[16] Heiko Müller, Johann-Christoph Freytag., "Problems, Methods, and Challenges in Data Cleansing." Berlin : HUB-IB-164, 2003.

[17] Pearson, Ronald K., *The Problem of Disguised Missing Data.* New York : ACM SIGKDD Explorations Newsletter, 2006.

[18] Databank. *Turkish Statistical Institute.* [Online] 1 2006. http://www.tuik.gov.tr/jsp/duyuru/upload/vt/vt.htm.

[19] Pima Indians Diabetes Data Set . *UCI Machine Learning Repository.* [Online] http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes.

[20] Heckert, Alan., "CHI SQUARE TWO SAMPLE." *http://www.itl.nist.gov.* [Online] 2006. http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/chi2samp.htm.

[21] Carmel, Liran., Multivariate Analysis Toolbox for Matlab® . [Online] 5 2008. http://www.ncbi.nlm.nih.gov/CBBresearch/Fellows/Carmel/software/MVA/mva.html.

[22] Pima Indians Diabetes Data Set. *UCI Machine Learning Repository.* [Online] http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes.

[23] Adverse Event Reporting System (AERS). *U.S. Food and Drug Administration.* [Online] 1 2009. http://www.fda.gov/cder/aers/default.htm.