JRC Scientific and Technical Reports

Privacy Preserving Data Mining, a Data Quality Approach

Igor Nai Fovino and Marcelo Masera

EUR 23070 EN - 2008

The Institute for the Protection and Security of the Citizen provides research-based, systems-oriented support to EU policies so as to protect the citizen against economic and technological risk. The Institute maintains and develops its expertise and networks in information, communication, space and engineering technologies in support of its mission. The strong cross-fertilisation between its nuclear and non-nuclear activities strengthens the expertise it can bring to the benefit of customers in both domains.

European Commission
Joint Research Centre
Institute for the Protection and Security of the Citizen

Contact information
Address: Via E Fermi I-21020 Ispra (VA) ITALY
E-mail: igor.nai@jrc.it
Tel.: +39 0332786541
Fax: +39 0332789576

http://ipsc.jrc.ec.europa.eu/
http://www.jrc.ec.europa.eu/

JRC JRC42700

EUR 23070 EN
ISSN 1018-5593

Luxembourg: Office for Official Publications of the European Communities

Printed in Italy

Privacy Preserving Data Mining, A Data Quality Approach

Igor Nai Fovino, Marcelo Masera

16th January 2008

Contents

Introduction

Introduction

Different approaches have been introduced by researchers in the field of privacy preserving data mining. These approaches have in general the advantage to require a minimum amount of input (usually the database, the information to protect and few other parameters) and then a low effort is required to the user in order to apply them. However, some tests performed by Bertino, Nai and Parasiliti [10,97] show that actually some problems occur when applying PPDM algorithms in contexts in which the meaning of the sanitized data has a critical relevance. More specifically, these tests (see Appendix A for more details), show that the sanitization often introduces false information or completely changes the meaning of the information. These phenomenas are due to the fact that, current approaches to PPDM algorithms do not take into account two relevant aspects:

- **Relevance of data**: not all the information stored in the database has the same level of relevance and not all the information can be dealt in the same way. An example may clarify this concept. Consider a categoric database[1] and assume we wish to hide the rule $AB \rightarrow CD$. A privacy-preserving association mining algorithm would retrieve all transactions supporting this rule and then change some items (e.g. item C) in these transactions in order to change the confidence of the rule. Clearly, if the item C has low relevance, the sanitization will not affect much the quality of the data. If, however, item C represents some critical information, the sanitization can result in severe damages to the data.

- **Structure of the database**: information stored in a database is strongly influenced by the relationships between the different data items. These relationships are not always explicit. Consider as an example an Health Care database storing patient's records. The medicines taken by patients are part of this database. Consider two different medicines, say A and B and assume that the following two constraints are defined: "The maximum amount of medicine A per day must be 10ml. If the patient also takes

---

[1] We define a categoric database a database in which attributes have a finite (but eventually large) number of distinct values with no ordering among the values. An example is the Market Basket database.

> medicine B the max amount of A must be 5ml". If a PPDM algorithm modifies the data concerning to the amount of medicine A without taking into account these constraints, the effects on the health of the patient could be potentially catastrophic.

Analyzing these observations, we argue that the general problem of current PPDM algorithms is related to Data Quality. In [70], a well known work in the data quality context, Orr notes that data quality and data privacy are in some way often related. More precisely an high quality of data often corresponds a low level of privacy. It is not simple to explain this correlation. We try in this introduction to give an intuitive explanation of this fact in order to give an idea of our intuition.

Any "Sanitization" on a target database has an effect on the data quality of the database. It is obvious to deduce that, in order to preserve the quality of the database, it is necessary to take into consideration some function $\Pi(DB)$ representing the relevance and the use of the data contained in the DB. Now, by taking another little step in the discussion, it is evident that the function $\Pi(DB)$ has a different output for different databases[2]. This implies that a PPDM algorithm trying to preserve the data quality needs to know specific information about the particular database to be protected before executing the sanitization.

We believe then, that the data quality is particularly important for the case of *critical database*, that is database containing data that can be critical for the life of the society (e.g. medical database, Electrical control system database, military Database etc.). For this reason data quality must be considered as the central point of every type of data sanitization.

In what follows we present a more detailed definition of DQ. We introduce a schema able to represent the Π function and we present two algorithms based on DQ concept.

---

[2] We use here the term "different" in its wide sense, i.e. different meaning of date, different use of data, different database architecture and schema etc.

Chapter 1

Data Quality Concepts

Traditionally DQ is a measure of the consistency between the data views presented by an information system and the same data in the real-world [70]. This definition is strongly related with the classical definition of information system as a "model of a finite subset of the real world" [56]. More in detail Levitin and Redman [59] claim that DQ is the instrument by which it is possible to evaluate if data models are well defined and data values accurate. The main problem with DQ is that its evaluation is relative [94], in that it usually depends from the context in which data are used. DQ can be assumed as a complex characteristic of the data itself. In the scientific literature DQ is considered a multi-dimensional concept that in some environments involves both objective and subjective parameters [8, 104, 105]. Table 1 lists of the most used DQ parameters [104].

Accuracy	Format	Comparability	Precision	Clarity
Reliability	Interpretability	Conciseness	Flexybility	Usefulness
Timeliness	Content	Freedom from bias	Understandability	Quantitativeness
Relevance	Efficiency	Informativeness	Currency	Sufficiency
Completeness	Importance	Level of Detail	consistence	Usableness

Table 1.1: Data Quality parameters

In the context of PPDM, DQ has a similar meaning, with however an important difference. The difference is that in the case of PPDM the real world is the original database, and we want to measure how closely the sanitized database is to the original one with respect to some relevant properties. In other words we are interested in assessing whether, given a target database, the sanitization phase will compromise the quality of the mining results that can be obtained from the sanitized database. In order to understand better how to measure DQ we introduce a formalization of the sanitization phase. In a PPDM process a given database DB is modified in order to obtain a new database DB'.

Definition 1
Let DB *be the database to be sanitized. Let* Γ_{DB} *be the set of all the aggregate information contained in* DB*. Let* $\Xi_{DB} \subseteq \Gamma_{DB}$ *be the set of the sensitive information to hide. A transformation* $\xi : D \rightarrow D$*, where* D *is the set of possible instances of a* DB *schema, is a perfect sanitization if* $\Gamma_{\xi(DB)} = \Gamma_{DB} - \Xi_{DB}$

In other words, the ideal sanitization is the one that completely removes the sensitive high level information, while preserving at the same time the other information. This would be possible in the case in which constraints and relationship between the data and between the information contained in the database do not exist, or, roughly speaking, assuming the information to be a simple aggregation of data, if the intersection between the different information is always equal to $\emptyset$. However, this hypothetical scenario, due to the complexity of modern databases in not possible. In fact, as explained in the introduction of this section, we must take into account not only the high level information contained in the database, but we must also consider the relationships between different information or different data, that is the Π function we mentioned in the previous section. In order to identify the possible parameters that can be used to assess the DQ in the context of PPDM Algorithms, we performed a set of tests. More in detail we perform the following operations:

1. We built a set of different types of database, with different data structures and containing information completely different from the meaning and the usefulness point of view.
2. We identified a set of information judged relevant for every type of database.
3. We applied a set of PPDM algorithms to every database in order to protect the relevant information.

Observing the results, we identified the following four categories of damages (or loss of quality) to the informative asset of the database:

- **Ambiguous transformation:** This is the case in which the transformation introduces some uncertainty in the non sensitive information, that can be then misinterpreted. It is for example the case of aggregation algorithms and Perturbation algorithms. It can be viewed even as a precision lack when for example some numerical data are standardized in order to hide some information.
- **Incomplete transformation:** The sanitized database results incomplete. More specifically, some values of the attributes contained in the database are marked as “Blank”. For this reason information may result incomplete and cannot be used. This is typically the effect of the Blocking PPDM algorithm class.
- **Meaningless transformation:** In this case the sanitized database contains information without meaning. That happen in many cases when

perturbation or heuristic-based algorithms are applied without taking into account the meaning of the information stored into the database.

- **Implicit Constraints Violation:** Every database is characterized by some implicit constraints derived from the external world that are not directly represented in the database in terms of structure and relation (the example of medicines A and B presented in the previous section could be an example). Transformations that do not consider these constraints risk to compromise the database introducing inconsistencies in the database

The results of these experiments magnify the fact that, in the context of privacy preserving data mining, only a little portion of the parameters showed in table 1.1 are of interest. More in details, are relevant the parameters allowing to capture a possible variation of meaning or that are indirectly related with the meaning of the information stored in a target database. Therefore, we identify as most appropriate DQ parameters for PPDM Algorithms the following dimensions:

- Accuracy: it measures the proximity of a sanitized value *a'* to the original value a. In an informal way, we say that a tuple t is accurate if the sanitization has not modified the attributes contained in the tuple t
- Completeness: it evaluates the percentage of data from the original database that are missing from the sanitized database. Therefore a tuple t is complete if after the sanitization every attribute is not empty.
- Consistency: it is related to the semantic constraints holding on the data and it measures how many of these constraints are still satisfied after the sanitization.

We now present the formal definitions of those parameters for use in the remainder of the discussion.

Let OD be the original database and SD be the sanitized database resulting from the application of the PPDM algorithm. Without loosing generality and in order to make simpler the following definitions, we assume that OD (and consequently SD) be composed by a single relation. We also adopt the positional notation to denote attributes in relations. Thus, let od_i (sd_i) be the i-th tuple in OD (SD), then od_{ik} (sd_{ik}) denotes the k^{th} attribute of od_i (sd_i). Moreover, let n be the total number of the attributes of interest, we assume that attributes in positions $1, \ldots, m$ ($m \leq n$) are the primary key attributes of the relation.

Definition 2
Let sd_j *be a tuple of* SD. *We say that* sd_j *is* ***Accurate*** *if* $\neg\exists od_i \in OD$ *such that* $((od_{ik} = sd_{jk})\forall k = 1..m \wedge \exists(od_{if} \neq sd_{jf}), (sd_{jf} \neq NULL), f = m + 1, .., n))$.
Definition 3
A sd_j *is* ***Complete*** *if* $(\exists od_i \in OD$ *such that* $(od_{ik} = sd_{jk})\forall k = 1..m) \wedge$

$(\neg\exists(sd_{jf} = NULL), f = m+1,..,n)$.

Where we intend for "NULL" value a value without meaning in a well defined context.

Let C be the set of the constraints defined on database OD, in what follows we denote with c_{ij} the j^{th} constraint on attribute i. We assume here constraints on a single attribute, but, it is easily possible to extend the measure to complex constraints.

Definition 4
A tuple sd_k *is* ***Consistent*** *if* $\neg\exists c_{ij} \in C$ *such that* $c_{ij}(sd_{ki}) = false, i = 1..n$
Starting from these definitions, it is possible to introduce three metrics (reported in Table 1.2), that allow us to measure the lack of accuracy, completeness and consistency. More specifically, we define the lack of accuracy as the proportion of non accurate items (i.e. the amount of items modified during the sanitization), with respect to the number of items contained in the sanitized database. Similarly, the lack of completeness is defined as the proportion of non complete items (i.e. the items substituted with a NULL value during the sanitization) with respect to the total number of items contained in the sanitized database. The consistency lack, is simply defined as the number of constraints violation in SD due to the sanitization phase.

Name	**Short Explanation**	**Expression**
Accuracy Lack	The proportion of non accurate items in SD	$\lambda_{\mathrm{SD}} = \frac{\|SD_{\mathrm{nacc}}\|}{\|SD\|}$
Completeness Lack	The proportion of non complete items in SD	$\vartheta_{\mathrm{SD}} = \frac{\|SD_{\mathrm{nc}}\|}{\|SD\|}$
Consistency Lack	The number of constraint violations in SD	$\varpi_{\mathrm{SD}} = Nc$

Table 1.2: The three parameters of interest in PPDM DQ evaluation. In the expressions SD_{nacc} denoted the set of not accurate items, SD_{nc} denotes the set of not complete items and N_C denotes the number of constraint violations

Chapter 2

The Data Quality Scheme

In the previous section, we have presented a way to measure DQ in the sanitized database. These parameters are, however, not sufficient to help us in the construction of a new PPDM algorithm based on the preservation of DQ. As explained in the introduction of this report, the main problem of actual PPDM algorithms is that they are not aware of the relevance of the information stored in the database, nor of the relationships among these information.

It is necessary to provide then a formal description that allow us to magnify the aggregate information of interest for a target database and the relevance of DQ properties for each aggregate information (for some aggregate information not all the DQ properties have the same relevance). Moreover, for each aggregate information, it is necessary to provide a structure in order to describe the relevance (in term of consistency, completeness and accuracy level required) at the attribute level and the constraints the attributes must satisfy. The Information Quality Model (IQM) proposed here addresses this requirement.

In the following, we give a formal definition of Data Model Graph (DMG) (used to represent the attributes involved in an aggregate information and their constraints) and Aggregation Information Schema (AIS). Moreover we give a definition for an Aggregate information Schema Set (ASSET). Before giving the definition of DMG, AIS and ASSET we introduce some preliminary concepts.

Definition 5
An Attribute Class is defined as the tuple $AT_C =< name, AW, AV, CW, CV, CSV, Slink >$ *where:*

- *Name is the attribute id*
- AW *is the accuracy weight for the target attribute*
- AV *is the accuracy value*
- CW *is the completeness weigh for the target attribute*
- CV *is the completeness value*

- *CSV is the consistency value*
- *Slink is list of simple constraints.*

An *attribute class*, represents an attribute of a target database schema, involved in the construction of a certain aggregate information for which we want to evaluate the data quality. The attributes, however, have a different relevance in an aggregate information. For this reason a set of weights is associated to every attribute class, specifying, for example, if for a certain attribute a lack of accuracy must be considered as relevant damage or not.

The attributes are the bricks of our structure. However, a simple list of attributes in not sufficient. In fact, in order to evaluate the consistency property, we need to take in consideration also the relationships between the attributes. The relationships, can be represented by logical constraints, and the validation of these constraints give us a measure of the consistency of an aggregate information. As in the case of the attributes, also for the constraints we need to specify their relevance (i.e. a violation of a constraint cause a negligible or a severe damage?). Moreover there exists constraints involving at the same time more then two attributes. In what follows the definitions of simple constraint and complex constraint are showed.

Definition 6
A Simple Constraint Class is defined as the tuple $SC_C =< name, Constr, CW, Clink, CSV >$ where:

- *Name is the constraint id*
- *Constraint describes the constraint using some logic expression*
- *CW is the weigh of the constraint. It represents the relevance of this constraint in the AIS*
- *CSV is the number of violations to the constraint*
- *Clink it is the list of complex constraints defined on SC_C.*

Definition 7
A Complex Constraint Class is defined as the tuple $CC_C =< name, Operator,$ $CW, CSV, SC_C\_link >$ where:

- *Name is the Complex Constraint id*
- *Operator is the "Merging" operator by which the simple constraints are used to build the complex one.*
- *CW is the weigh of the complex constraint*
- *CSV is the number of violations*

- *$SC_C link$ is the list of all the SC_C that are related to the CC_C.*

We have now the bricks and the glue of our structure. A DMG is a collection of attributes and constraints allowing one to describe an aggregate information. More formally, let D be a database:

Definition 8
*A **DMG** (Data Model Graph) is an oriented graph with the following features:*

- *A set of nodes N_A where each node is an Attribute Class*
- *A set of nodes SC_C where each node describes a Simple Constraint Class*
- *A set of nodes CC_C where each node describes a Complex Constraint Class*
- *A set of direct edges $L_{Nj,Nk}$: $L_{Nj,Nk} \in ((N_A X SC_C) \cup (SC_C X CC_C) \cup (SC_C X N_A) \cup (CC_C X N_A))$.*

As is the case of attributes and constraints, even at this level, there exists aggregate information more relevant than other. We define then an AIS as follows:

Definition 9
*An **AIS** ϕ is defined as a tuple $< \gamma, \xi, \lambda, \vartheta, \varpi, W_A IS >$ where: γ is a name, ξ is a DMG, λ is the accuracy of AIS, ϑ is the completeness of AIS, ϖ is the consistency of AIS and W_{AIS} represent the relevance of AIS in the database.*
In a database there are usually more than an aggregate information for which one is interested to measure the quality.

Definition 9
*With respect to D we define **ASSET** (Aggregate information Schema Set) as the collection of all the relevant AIS of the database.*

The DMG completely describes the relations between the different data items of a given AIS and the relevance of each of these data respect to the data quality parameter. It is the “road map” that is used to evaluate the quality of a sanitized AIS

As it is probably obvious, the design of a good IQM schema and more in general of a good ASSET, it is fundamental to characterize in a correct way the target database.

2.0.1 An IQM Example

In order to magnify which is the real meaning of a data quality model, we give in this section a brief description of a possible context of application and we then describe an example of IQM Schema.

The Example Context

Modern power grid starting today to use Internet, and more generally public network, in order to monitor and to manage remotely electric apparels (local

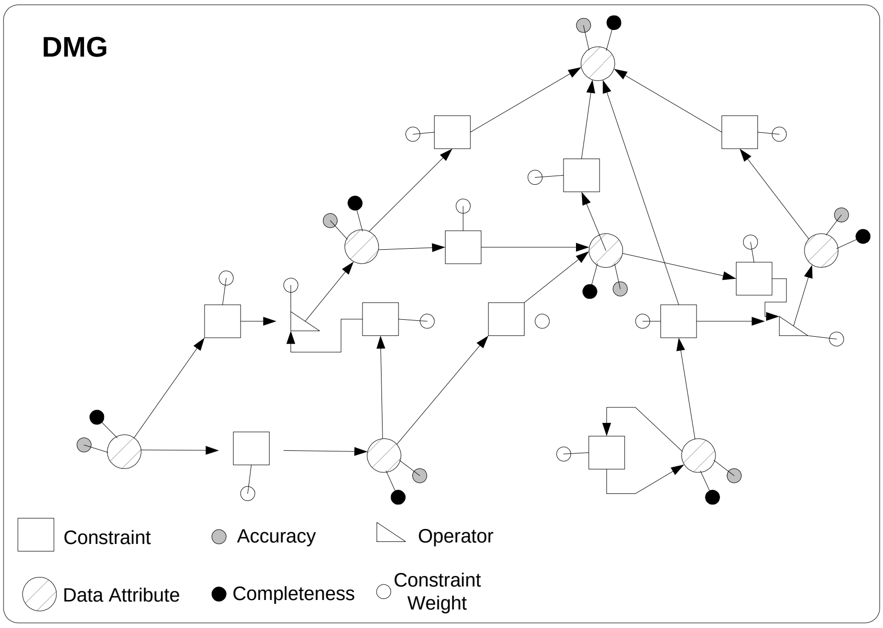

Figure 2.1: Example of DMG

energy stations, power grid segments etc.) [62]. In such a context, there exist a lot of problems related to system security and to the privacy preservation.

More in details, we take as example a very actual privacy problem caused by the new generation of *home electricity meters*. In fact, such a type of new meters, are able to collect a big number of information related to the home energy consumption. Moreover, they are able to send the collected information to a central remote database. The information stored in such database can be easily classified as sensitive.

For example, the electricity meters can register the energy consumption per hour and per day in a target house. From the analysis of this information it is easy to retrieve some behavior of the people living in this house (i.e. from the 9 a.m. to 1 p.m. the house is empty, from the 1 p.m. to 2.30 p.m. someone is at home, from the 8 p.m. the whole family is at home etc.). This type of information is obviously sensitive and must be protected. On the other hand, information like the maximum energy consumption per family, the average consumption per day, the subnetwork of pertinence and so on, have an high relevance for the analysts who want to correctly analyze the power grid to calculate for example the amount of power needed in order to guarantee an acceptable service over a certain energy subnetwork.

This is a good example of a situation in which the application of a privacy preserving techniques without knowledge about the meaning and the relevance

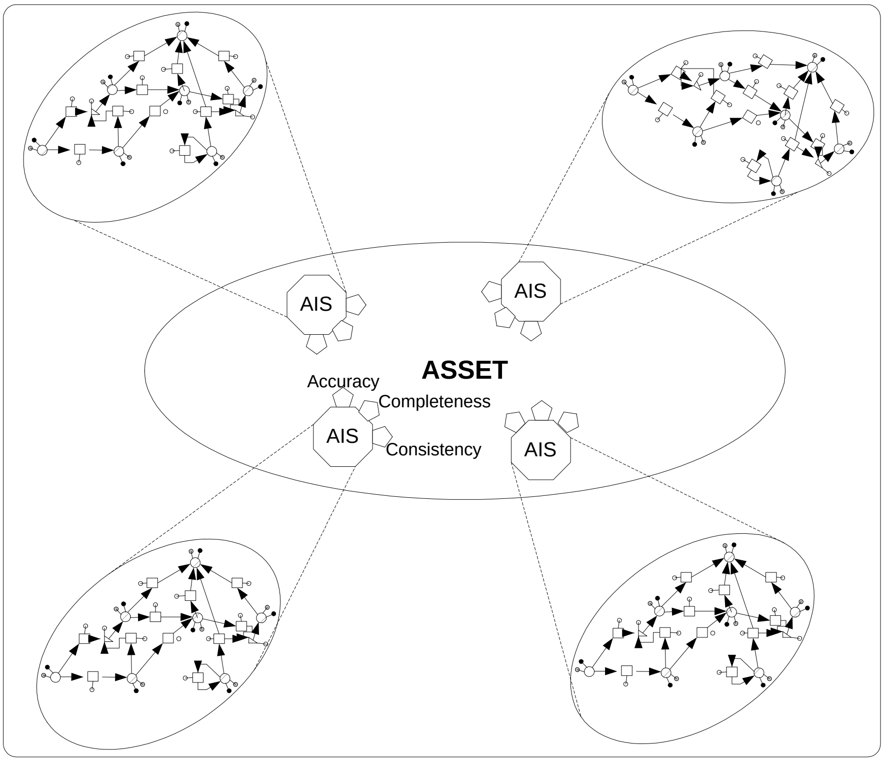

Figure 2.2: Example ASSET: to every AIS contained in the Asset the dimensions of Accuracy, Completeness and Consistency are associated. Moreover, to every AIS the proper DMG is associated. To every attribute and to every constraint contained in the DMG, the three local DQ parameters are associated

of the data contained in the database could cause a relevant damage (i.e. wrong power consumption estimation and consequently blackout).

An IQM Example

The *electric meters database* contains several information related to energy consumption, average electric frequency, fault events etc. It is not our purpose to describe here the whole database. In order to give an example of IQM schema we describe two tables of this database and two DMG of a possible IQM schema.

In our example, we consider the tables related to the *localization information* of the electric meter and the information related to the *per day consumption* registered by a target electric meter.

More in details, the two table we consider have the following attributes:
Localization Table

- *Energy_Meter_ID*: it contains the ID of the energy meter.
- *Local Code*: it is the code of the city/region in which the meter is.
- *City*: it contains the name of the city in which the home hosting the meter is.
- *Network*: it is the code of the electric network.
- *Subnetwork*: it is the code of the electric subnetwork.

Per Day Data consumption table

- *Energy_Meter_ID*: it contains the ID of the energy meter.
- *Average_Energy_Consumption*: it contains the average energy used by the target home during a day.
- *Date*: It contains the date of registration.
- *Energy_consumption_per_hour*: it is a vector of attributes containing the average energy consumption per hour.
- *Max_Energy_consumption_per_hour*: it is a vector of attributes containing the maximum energy consumption per hour.
- *Max_Energy_Consumption*: it contains the maximum energy consumption per day measured for a certain house.

As we have already underlined, from this information it is possible to guess several sensitive information related to the behavior of the people living in a target house controlled by an electric meter. For this reason it is necessary to perform a sanitization process in order to guarantee a minimum level of privacy.

However, a sanitization without knowledge about the relevance and the meaning of data can be extremely dangerous. In figure 2.3 two DMG schemas associated to the data contained in the previously described tables are showed. In what follows, we give a brief description about the two schema.

AIS_Loc

The AIS_Loc is mainly related to the data contained in the localization table. As it is possible to see in figure 2.3 there exist some constraints associated with the different items contained in this table. For example we require the local code be in a range of values between 1 and 1000 (every different code is meaningless and cannot be interpreted; moreover a value out of this range reveals to malicious user that the target tuple is sanitized). The weight of this constraint is then high (near to 1).

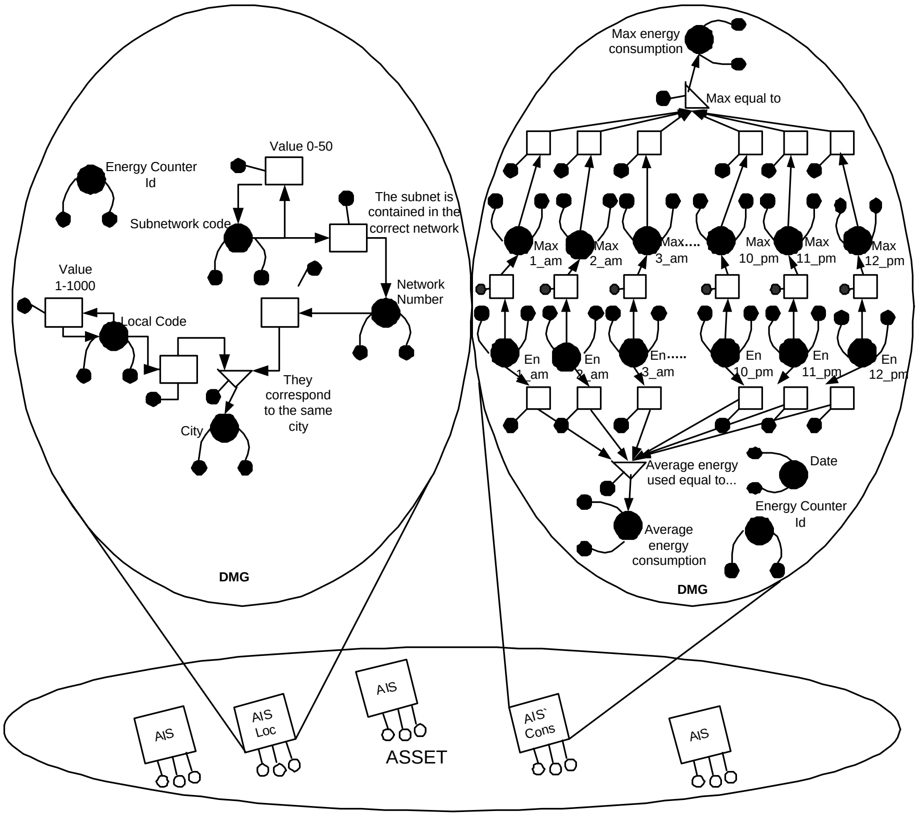

Figure 2.3: Example of two DMG schemas associated to a National Electric Power Grid database

Moreover, due to the relevance of the information about the regional localization of the electric meter, even the weights associated with the completeness and the accuracy of the local code have an high value.

A similar constraint exists even over the subnetwork code. With regard to the completeness and the accuracy of this item, a too high weight may compromise the ability to hide sensitive information. In other words, considering the three main localization items, local code, network number, subnetwork number, it is reasonable to think about a decreasing level of accuracy and completeness. In this way, we try to guarantee the preservation of information about the area of pertinence, avoiding the exact localization of the electric meter (i.e. “I know that it is in a certain city, but I do not know the quarter in which it is”).

AIS_Cons

In figure 2.3 is showed a DMG (AIS_Cons) related to the energy consumption table. In this case the value of the information stored in this table, is mainly related to the maximum energy consumption and the average energy consumption. For this reason their accuracy and completeness weights must be high (to magnify their relevance in the sanitization phase). However, it is even necessary to preserve the coherence between the values stored in the *Energy_consumption_per_hour* vector and the average energy consumption per day. In order to do this, a complex constraint is used. In this way we describe the possibility to permute or to modify the values of energy consumption per hour (in the figure En 1_am ..En 2_am etc.) maintaining however the average energy consumption consistent.

At the same way it is possible to use another complex constraint in order to express the same concept for the maximum consumption per day in relation with the maximum consumption per hour. Exist finally a relation between the maximum energy consumption per hour and the correspondent average consumption per hour. In fact the max consumption per hour cannot be smaller than the average consumption per hour. This concept is represented by a set of simple constraints. Finally the weight of accuracy and completeness of the Energy_Meter_ID are equal to 1, because a condition needed to make consistent the database is to maintain at least the individuality of the electric meters.

By the use of these two schemas, we described some characteristics of the information stored in the database, related with their meaning and their usage. In the following chapters, we show some PPDM algorithms realized in order to use these additional schema in order to perform a softer (from the DQ point of view) sanitization

Chapter 3

Data Quality Based Algorithms

A large number of Privacy Preserving Data Mining algorithms have been proposed. The goal of these algorithms is to prevent privacy breaches arising from the use of a particular Data Mining Technique. Therefore algorithms exist designed to limit the malicious use of Clustering Algorithms, of Classification Algorithms, of Association Rule algorithms.

In this report, we focus on Association Rule Algorithms. We choose to study a new solution for this family of algorithms for the following motivations:

- Association Rule Data Mining techniques has been investigated for about 20 years. For this reason exists a large number of DM algorithms and application examples exist that we take as a starting point to study how to contrast the malicious use of this technique.

- Association Rule Data Mining can be applied in a very intuitive way to categoric databases. A categoric database can be easily synthesized by the use of automatic tools allowing us to quickly obtain for our tests a big range of different type of databases with different characteristics.

- Association Rules seem to be the most appropriate to be used in combination with the IQM schema.

- Results of classification data mining algorithms can be easily converted into association rules. This it is in prospective very important, because it allows us to transfer the experience in association rule PPDM family to the classification PPDM family without much effort.

In what follows, we present before some algorithms developed in the context of Codmine project [97] (we remember here that our contribution in the Codmine Project was related to the algorithm evaluation and to the testing framework and not with the Algorithm development). We present then some considerations about these algorithms and finally we present our new approach to association rule PPDM algorithms.

3.1 ConMine Algorithms

We present here a short description of the privacy preserving data mining algorithms, which have been developed in the context of Codmine project. In the association rule hiding context, two different heuristic-based approaches have been proposed: one is based on data deletion according to which the original database is modified by deleting the data values; the other is based on data fuzzification, thus hiding the sensitive information by inserting unknown values to the database. Before giving more details concerning the proposed approaches, we briefly recall the problem formulation of association rule hiding.

3.2 Problem Formulation

Let D be a transactional database and $I = \{i_1, \ldots, i_n\}$ be a set of items, which represents the set of products that can be sold by the database owner. Each transaction can be considered as subset of items in I. We assume that the items in a transaction or an itemset are sorted in lexicographic order. According to a bitmap notation, each transaction t in the database D is represented as a triple $< TID, values\_of\_items, size >$, where TID is the identifier of the transaction t, $values\_of\_items$ is a list of values, one value for each item in I, associated with transaction t. An item is supported by a transaction t if its value in the $values\_of\_items$ is 1, and it is not supported by t if its value in the $values\_of\_items$ is 0. $Size$ is the number of 1 values which appear in the $values\_of\_items$, that is the number of items supported by the transaction. An association rule is an implication of the form $X \Rightarrow Y$ between disjoint itemsets X and Y in I. Each rule is assigned both to a support and to a confidence value. The first one represents the probability to find in the database transactions containing all the items in $X \cup Y$, whereas the confidence is the probability to find transactions containing all the items in $X \cup Y$, once we know that they contain X. Note that while the support is a measure of a frequency of a rule, the confidence is a measure of the strength of the relation between the antecedent X and the consequent Y of the rule. Association rule mining process consists of two steps: 1) the identification of all the frequent itemsets, that is, all the itemsets, whose supports are bigger than a pre-determined minimum support threshold, *min_supp*; 2) the generation of strong association rules from the frequent itemsets, that is those frequent rules whose confidence values are bigger than a minimum confidence threshold, min_conf. Along with the confidence and the support, a *sensitivity level* is assigned only to both frequent and strong rules. If a strong and frequent rule is above a certain sensitivity level, the hiding process should be applied in such a way that either the frequency or the strength of the rule will be reduced below the *min_supp* and the *min_conf* correspondingly.

The problem of association rule hiding can be stated as follows: given a database D, a set R of relevant rules that are mined from D and a subset R_h of R, we want to transform D into a database D' in such a way that the rules in R can still be mined, except for the rules in R_h.

3.3 Rule hiding algorithms based on data deletion

Rule hiding approach based on data deletion operates by deleting the data values in order to reduce either the support or the confidence of the sensitive rule below the corresponding minimum support or confidence thresholds, MST or MCT, which are fixed by the user, in such a way that the changes introduced in the database are minimal. The decrease in the support of a rule $X \Rightarrow Y$ is done by decreasing the support of its generating itemset $X \cup Y$, that is the support of either the rule antecedent X, or the rule consequent Y, through transactions that fully support the rule. The decrease in the confidence $Conf(X \Rightarrow Y) = \frac{Supp(X \Rightarrow Y)}{Supp(X)}$ of the rule $X \Rightarrow Y$, instead, can be accomplished: 1) by decreasing the support of the rule consequent Y through transactions that support both X and Y; 2) by increasing the support of the rule antecedent through transactions that do not fully support Y and partially support X. Five data deletion algorithms have been developed for privacy preserving data mining, whose pseudocodes are presented in Figures 3.1, 3.2 and 3.3. Algorithm 1 decreases the confidence of a sensitive rule below the min_conf threshold by increasing the support of the rule antecedent, while keeping the rule support fixed. Algorithms 2 reduces the rule support by decreasing the frequency of the consequent through transactions that support the rule. Algorithm 3 operates as the previous one with the only exception that the choice of the item to remove is done according to a minimum impact criterion. Finally, Algorithms 4 and 5 decrease the frequency of large itemsets from which sensitive rules can be mined.

3.4 Rule hiding algorithms based on data fuzzification

According to the data fuzzification approach, a sequence of symbols in the new alphabet of an item {0,1,?} is associated with each transaction where one symbol is associated with each item in the set of items I. As before, the i^{th} value in the list of values is 1 if the transaction supports the i^{th} item, the value is 0 otherwise. The novelty of this approach is the insertion of an uncertainty symbol, a question mark, in a given position of the list of values which means that there is no information on whether the transaction supports the corresponding item. In this case, the confidence and the support of an association rule may be not unequivocally determined, but they can range between a minimum and a maximum level. The minimum support of an itemset is defined as the per-

```
INPUT:   the source database $D$, the number $|D|$ of transactions in $D$, a set $R_h$ of rules to hide, the $min\_conf$ thr
OUTPUT: the database $D$ transformed so that the rules in $R_h$ cannot be mined

Begin
    Foreach rule $r$ in $R_h$ do
    {
        1. $T'_{l_r} = \{t \in D \mid t$ does not fully supports $r_r$, and partially supports $l_r\}$
        foreach transaction $t$ in $T'_{l_r}$ do
        {
            2. t.num_items = $|I| - |l_r \cap t.values\_of\_items|$
        }
        3. Sort($T'_{l_r}$) in descending order of number of items of $l_r$ contained
        Repeat until($conf(r) < min\_conf$)
        {
            4. $t = T'_{l_r}[1]$
            5. set_all_ones(t.values_of_items, $l_r$)
            6. $N_{l_r} = N_{l_r} + 1$
            7. $conf(r) = N_r / N_{l_r}$
            8. Remove $t$ from $T'_{l_r}$
        }
        9. Remove $r$ from $R_h$
    }
End
```

Figure 3.1: Algorithm 1 for Rule Hiding by Confidence Decrease

```
INPUT:   the source database $D$, the size of the
database $|D|$, a set $R_h$ of rules to hide, the
min_supp threshold, the min_conf threshold
OUTPUT: the database $D$ transformed so that
the rules in $R_h$ cannot be mined

Begin
  Foreach rule $r$ in $R_h$ do
  {
    1. $T_r = \{t \in D \mid t$ fully supports $r\}$
    foreach transaction $t$ in $T_r$ do
    {
      2. t.num_items = count($t$)
    }
    3. Sort($T_r$) in ascending order of transaction size
    Repeat until ($supp(r) < min\_sup$)
              or ($conf(r) < min\_conf$)
    {
      4. $t = T_r[1]$
      5. j = choose_item($r_r$)
      6. set_to_zero(j,t.values_of_items)
      7. $N_r = N_r - 1$
      8. $supp(r) = N_r/|D|$
      9. $conf(r) = N_r/N_{l_r}$
      10. Remove $t$ from $T_r$
    }
    11. Remove $r$ from $R_h$
  }
End
```

```
INPUT:   the source database $D$, the size of the
database $|D|$, a set $R_h$ of rules to hide, the
min_supp threshold, the min_conf threshold
OUTPUT: the database $D$ transformed so that the
rules in $R_h$ cannot be mined

Begin
  Foreach rule $r$ in $R_h$ do
  {
    1. $T_r = \{t \in D \mid t$ fully supports $r\}$
    foreach transaction $t$ in $T_r$ do
    {
      2. t.num_items = count($t$)
    }
    3. Sort($T_r$) in decreasing order of transaction size
    Repeat until ($supp(r) < min\_sup$)
              or ($conf(r) < min\_conf$)
    {
      4. $t = \mathrm{T}_r[1]$
      5. j = choose_item($r$)
      6. set_to_zero(j,t.values_of_items)
      7. $N_r = N_r - 1$
      8. Recompute $N_{l_r}$
      9. $supp(r) = N_r/|D|$
      10. $conf(r) = N_r/N_{l_r}$
      11. Remove $t$ from $T_r$
    }
    12. Remove $r$ from $R_h$
  }
End
```

Figure 3.2: Algorithms 2 and 3 for Rule Hiding by Support Decrease

centage of transactions that certainly support the itemset, while the maximum support represents the percentage of transactions that support or could support the itemset. The minimum confidence of a rule is obviously the minimum level of confidence that the rule can assume based on the support value, and similarly for the maximum confidence. Given a rule r, $minconf(r) = \frac{minsup(r)*100}{maxsup(l_r)}$ and $maxconf(r) = \frac{maxsup(r)*100}{minsup(l_r)}$, where l_r denotes the rule antecedent.

Considering the support interval and the minimum support threshold, MST, we have the following cases for an itemset A:

- A is *hidden* when $maxsup(A)$ is smaller than MST;
- A is *visible* with an *uncertainty level* when $minsup(A) \leq MST \leq maxsup(A)$;
- A is *visible* if $minsup(A)$ is greater than or equal to MST.

The same reasoning applies to the confidence interval and the minimum confidence threshold (MCT).

INPUT: the source database D, the size of the database $|D|$, a set L of large itemsets, the set L_h of large itemsets to hide, the *min_supp* threshold
OUTPUT: the database D modified by the hiding of the large itemsets in L_h

```
Begin
  1. Sort($L_h$) in descending order of size and support
  2. $T_h = \{T_Z \mid Z \in L_h$ and $\forall t \in D :$
     $t \in T_Z \Longrightarrow t$ supports $Z\}$
  Foreach $Z$ in $L_h$
  {
    3. Sort($T_Z$) in ascending order of transaction size
    Repeat until $(supp(r) < min\_sup)$
    {
      4. t = popfrom($T_Z$)
      5. a = maximal support item in $Z$
      6. $a_S = \{X \in L_h | a \in X\ \}$
      Foreach $X$ in $a_S$
      {
        7. if $(t \in T_X)$ delete$(t, T_X)$
      }
      8. delete$(a, t, D)$
    }
  }
End
```

INPUT: the source database D, the size of the database $|D|$, a set L of large itemsets, the set L_h of large itemsets to hide, the *min_supp* threshold
OUTPUT: the database D modified by the hiding of the large itemsets in L_h

```
Begin
  1. Sort($L_h$) in descending order of support and siz
  Foreach $Z$ in $L_h$ do
  {
    2. i = 0
    3. j = 0
    4. $< T_Z > =$ sequence of transactions supporti
    5. $< Z > =$ sequence of items in $Z$
    Repeat until $(supp(r) < min\_sup)$
    {
      6. $a =< Z >$[i]
      7. $t = < T_Z >$[j]
      8. $a_S = \{X \in L_h | a \in X\ \}$
      Foreach $X$ in $a_S$
      {
        9. if $(t \in T_X)$ then delete$(t, T_X)$
      }
      10. delete$(a, t, D)$
      11. i = (i+1) modulo size($Z$)
      12. j = j+1
    }
  }
End
```

Figure 3.3: Algorithms 4 and 5 for Rule Hiding by Support Decrease of large itemsets

Traditionally, a rule hiding process takes place according to two different strategies: decreasing its support or its confidence. In this new context, the adopted alternative strategies aim to introduce uncertainty in the frequency or the importance of the rules to hide. The two strategies reduce the minimum support and the minimum confidence of the itemsets generating these rules below the minimum support threshold (MST) and minimum confidence threshold (MCT) correspondingly by a certain safety margin (SM) fixed by the user.
The Algorithm 6 for reducing the support of generating large itemset of a sensitive rule replaces 1's by " ? " for the items in transactions supporting the itemset until its minimum support goes below the minimum support threshold MST by the fixed safety margin SM. Algorithms 7 and 8 operate reducing the minimum confidence value of sensitive rules. The first one decreases the minimum support of the generating itemset of a sensitive rule by replacing items of the rule consequent by unknown values. The second one, instead, increases the maximum support value of the antecedent of the rule to hide via placing question marks in the place of the zero values of items in the antecedent.

```
INPUT:   the database $D$, a set $L$ of large itemsets,
the set $L_h$ of large itemsets to hide, $MST$ and $SM$
OUTPUT: the database $D$ modified by the fuzzification
of large itemsets in $L_h$

Begin
  1. Sort $L_h$ in descending order of size and
     minimum support of the large itemsets
  Foreach $Z$ in $L_h$
  {
    2. $T_Z = \{t \in D \mid t$ supports $Z\}$
    3. Sort the transaction in $T_Z$ in
       ascending order of transaction size
    Repeat until $(minsup(r) < MST - SM)$
    {
      4. Place a ? mark for the item with the
         largest minimum support of $Z$ in the
         next transaction in $T_Z$
      5. Update the supports of the affected
         itemsets
      6. Update the database $D$
    }
  }
End
```

Figure 3.4: Algorithm 6 for Rule Hiding by Support Reduction

3.5 Considerations on Codmine Association Rule Algorithms

Based on results obtained by the performed tests (See Appendix A for more details), it is possible to make some considerations on the presented algorithms. All these algorithms are able in any case (with the exclusion of the first algorithm) to hide the target sensitive information. Moreover the performance results are acceptable. The question is then: “Why there is the necessity to improve these algorithms if they all work well?” There exist some coarse parameters allowing us to measure the impact of the algorithms on the DQ of the released database. Just to understand here what is the problem, we say that during the tests we discovered in the sanitized database a considerable number of new rules (artifactual rules) not present in the original database. Moreover we discovered that some of the not sensitive rules contained in the original database, after the sanitization were erased (Ghost rules). This behavior in a critical database could have a relevant impact. For example, as usual let us consider the Health Database. The introduction of artifactual rules may deceive a doctor that, on the basis of these rules, can choose a wrong therapy for a patient. In

```
INPUT: the source database $D$, a set $R_h$
of rules to hide, $MST$, $MCT$ and $SM$
OUTPUT: the database $D$ transformed so that
the rules in $R_h$ cannot be mined

Begin
  Foreach $r$ in $R_h$ do
  {
    1. $T_r$={$t$ in $D$ | $t$ fully supports $r$}
    2. for each $t$ in $T_r$ count the number of items in $t$
    3. sort the transactions in $T_r$ in ascending order
       of the number of items supported
    Repeat until $(minsup(r) < MST - SM)$
                 or $(minconf(r) < MCT - SM)$
    {
      4. Choose the first transaction $t \in T_r$
      5. Choose the item $j$ in $r_r$ with the
         highest $minsup$
      6. Place a ? mark for the place of $j$ in $t$
      7. Recompute the $minsup(r)$
      8. Recompute the $minconf(r)$
      9. Recompute the $minconf$ of other
         affected rules
      10. Remove $t$ from $T_r$
    }
    11. Remove $r$ from $R_h$
  }
End
```

```
INPUT: the source database $D$, a set $R_h$
of rules to hide, $MCT$, and $SM$.
OUTPUT: the database $D$ transformed so that
the rules in $R_h$ cannot be mined

Begin
  Foreach $r$ in $R_h$ do
  {
    1. $T'_{l_r}$= {$t$ in $D$ | $t$ partially supports $l_r$
             and $t$ does not fully support $r_r$}
    2. for each $t$ in $T'_{l_r}$ count the number of items of
    3. sort the transactions in $T'_{l_r}$ in descending order
       the number of items of $l_r$ supported
    Repeat until $(minconf(r) < MCT - SM)$
    {
      4. Choose the first transaction $t \in T'_{l_r}$
      5. Place a ? mark in $t$ for the items in $l_r$
         that are not supported by $t$
      6. Recompute the $maxsup(l_r)$
      7. Recompute the $minconf(r)$
      8. Recompute the $minconf$ of other affected
      9. Remove $t$ from $T'_{l_r}$
    }
    10. Remove $r$ from $R_h$
  }
End
```

Figure 3.5: Algorithms 7 and 8 for Rule Hiding by Confidence Decrease

the same way, if some other rules are erased, a doctor could have a not complete picture of the patient health, generating then some unpleasant situation. This behavior is due to the fact that the algorithms presented act in the same way with any type of categorical database, without knowing nothing about the use of the database, the relationships between data etc. More specifically we note that the key point of this uncorrect behavior is in the selection of the item to be modified. In all these algorithm in fact, the items are selected randomly, without considering their relevance and their impact on the database. We then introduce an approach to select the proper set of items minimizing the impact on the database

3.6 The Data Quality Strategy (First Part)

As explained before, we want to develop a family of PPDM algorithms which take into account, during the sanitization phase, aspects related to the particular database we want to modify. Moreover we want such a PPDM family driven by the DQ assurance. In the context of Association Rules PPDM algorithm, and more in detail in the family of Perturbation Algorithms described in the previous section, we note that the point in which it is possible to act in order to preserve the quality of the database, is the selection of the Item. In other words the problem we want to solve is

How we can select the proper item to be modify, in order to limit the impact of this modification on the quality of the database?

The DQ structure we introduced previously can help us in achieving this goal. The general idea we suggest here it is relatively intuitive. In fact, a target DMG contains all the attributes involved in the construction of a relevant aggregate information. Moreover a weight is associated with each attribute denoting its accuracy and completeness. Every constraint involving this attribute is associated to a weight representing the relevance of its violation.

Finally, every AIS has its set of weights related to the three DQ measures and it is possible to measure the global level of DQ guaranteed after the sanitization. We are then able to evaluate the effect of an attribute modification on the Global DQ of a target database. Assuming that we have available the Asset schema related to a target database, our Data Hiding strategy can be described as follows:

1. A sensitive rule s is identified
2. The items involved in s are identified
3. By inspecting inside the Asset schema, the DQ damage is computed simulating the modification of these items
4. An item ranking is built considering the results of the previous operation
5. The item with a lower impact is selected
6. The sanitization is executed by modifying the chosen item

The proposed strategy can be improved. In fact, to compute the effect of the modifications, in case of very complex information, could be onerous in term of performances.

We note that, in a target association rule, a particular type of items may exist, that is not contained in any DMG of the database Asset. We call this type of Items **Zero Impact Items**. The modification of these particular items has no effect on the DQ of the database, because it was, in the Asset description phase, judged not relevant for the aggregate information considered. The improvement of our strategy is then the following: before starting the Asset inspection phase we search for the existence of Zero-Impact Items related to the target rules. If they exist, we do not perform the Asset inspection and we use one of these Zero-Impact items in order to hide the target rule.

3.7 Data Quality Distortion Based Algorithm

In the Distortion Based Algorithms, the database sanitization (i.e. the operation by which a sensitive rule is hidden), it is obtained by the distortion of some values contained in the transactions supporting the rule (partially or fully, depending from the strategy adopted) to be hidden. In what we propose, assuming to work with categorical databases like the classical Market Basket database [1], the distortion strategy we adopt is similar to the one presented in the Codmine algorithms. Once a good item is identified, a certain number of transactions supporting the rule are modified changing the item value from 1 to 0 until the confidence of the rule is under a certain threshold. Figure 3.6 shows the flow diagram describing the algorithm.

As it is possible to see, the first operation required is the identification of the rule to hide. In order to do this, it is possible to use a well known algorithm, named *APRIORI* algorithm [5], that, given a database, extract all the rules contained that have a confidence over a Threshold level [2]. Detail on this algorithm are described in the Appendix B. The output of this phase is a set of rules. From this set, a sensitive rule will be identified (in general this depends from its meaning in a particular context, but it is not in the scope of our work to define when a rule must be considered *sensitive*). The next phase implies the determination of the **Zero_Impact** Items (if they exist) associated wit the target rule.

If the **Zero_Impact** Itemset is empty, it means no item exists contained in the rule that can be assumed to be non relevant for the DQ of the database. In this case it is necessary to simulate what happens to the DQ of the database, when a non zero impact item is modified. The strategy we can adopt in this case is the following: we calculate the support and the confidence of the target rule before the sanitization (we can use the output obtained by the Apriori Algorithm).Then we compute how many steps are needed in order to downgrade the confidence of the rule under the minimum threshold (as it is possible to see from the algorithm code, there is little difference between the case in which the item selected is part of the antecedent or part of the consequent of the rule). Once we know how many transactions we need to modify, it is easy, inspecting the Asset, to calculate the impact on the DQ in terms of accuracy and consistency. Note that in the distortion algorithm evaluating the completeness is without meaning, because the data are not erased, but modified. The *Item Impact Rank* Algorithm is reported in Figure 3.8.
The Sanitization algorithm works as follows:

1. It selects all the transactions fully supporting the rule to be hidden.
2. It computes the Zero Impact set and if there exist such type of items, it randomly selects one of these items. If this set is empty, it calculates

---

[1] A market Basket Database is a database in which each transaction represents the products acquired by the customers.

[2] This Threshold level represents the confidence over which a rule can be identified as relevant and sure.

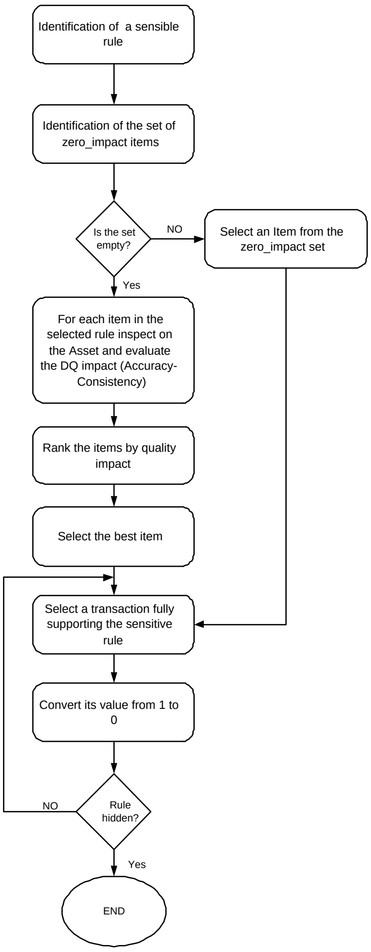
Identification of a sensible rule
Identification of the set of zero_impact items
Is the set empty?
NO
Select an Item from the zero_impact set
Yes
For each item in the selected rule inspect on the Asset and evaluate the DQ impact (Accuracy-Consistency)
Rank the items by quality impact
Select the best item
Select a transaction fully supporting the sensitive rule
Convert its value from 1 to 0
NO
Rule hidden?
Yes
END

Figure 3.6: Flow Chart of the DQDB Algorithm

the ordered set on Impact Items and it selects the one with the lower DQ Effect.

3. Using the selected item, it modifies the selected transactions until the rule is hidden.

The final algorithm is reported in Figure 3.9. Given the algorithm, for sake of completeness we try to give an idea about its complexity.

- The search of all zero impact items, assuming the items to be ordered according to a lexicographic order, can be executed in $N * O(lg(M))$ where N is the size of the rule and M is the number of different items contained in the Asset.

```
INPUT: the Asset Schema $A$ associated to the target database,
the rule $Rh$ to be hidden
OUTPUT: the set (possibly empty) $Zitem$ of zero Impact items discovered

Begin
zitem=$\emptyset$;
Rsup=Rh;
Isup=list of the item contained in Rsup;
While ($Isup \neq \emptyset$)
{
    res=0;
    Select item from Isup;
    res=Search(Asset, Item);
    if (res==0)then zitem=zitem+item;
    Isup=Isup-item;
}
return(zitem);
End
```

Figure 3.7: Zero Impact Algorithm

```
INPUT: the Asset Schema $A$ associated
to the target database,the rule $Rh$ to be hidden,
the $sup(Rh)$, the $sup(ant(Rh))$, the Threshold $Th$
OUTPUT: the set $Qitem$ ordered by Quality Impact

Begin
Qitem=$\emptyset$;
Confa=$\frac{Sup(Rh)}{Sup(ant(Rh))}$
Confp=Confa;
Nstep_if_ant=0;
Nstep_if_post=0;
While ($Confa > Th$) do
{
    Nstep_if_ant++;
    Confa=$\frac{Sup(Rh)-Nstep\_if\_ant}{Sup(ant(Rh))-Nstep\_if\_ant}$;
}
While ($Confb > Th$) do
{
    Nstep_if_post++;
    Confa=$\frac{Sup(Rh)-Nstep\_if\_post}{Sup(ant(Rh))}$;
}
For each item in $Rh$ do
{
    if ($item \in ant(Rh)$) then N=Nstep_if_ant;
        else N=Nstep_if_post;
    For each $AIS \in Asset$ do
    {
        node=recover_item(AIS,item);
        Accur_Cost=$node.accuracy\_Weight * N$;
        Constr_Cost=Constr_surf(N,node);
        item.impact=$item.impact+$
                $(Accur\_Cost * AIS.Accur\_weight)+$
                $+(Constr\_Cost * AIS.COnstr\_weight)$;
    }
}
sort_by_impact(Items);
Return(Items);
End
```

Figure 3.8: Item Impact Rank Algorithm for the Distortion Based Algorithm (we remember here that ant(Rh) indicates the antecedent component of a rule)

- The Cost of the *Impact Item Set* computation is less immediate to formulate. However, the computation of the sanitization step is linear $(O(\lfloor Sup(Rh) \rfloor))$. Moreover the constraint exploration in the worst case has a complexity in $O(NC)$ where NC is the total number of constraints contained in the Asset. This operation is repeated for each item in the rule Rh and then it takes $|Items|O(NC)$. Finally in order to sort the Itemset, we can use the MergeSort algorithm, and then the complexity is $|itemset|O(lg(|itemset|))$.

- The sanitization has a complexity in $O(N)$ where N is the number of transaction modified in order to hide the rule.

```
INPUT:   the Asset Schema $A$ associated to the target database,
the rule $Rh$ to be hidden, the target database $D$
OUTPUT: the Sanitized Database

Begin
Zitem=DQDB_Zero_Impact(Asset,Rh);
if ($Zitem \neq \emptyset$) then item=random_sel(Zitem);
    else
    {
        Impact_set=Items_Impact_rank(Asset,Rh,Sup(Rh),Sup(ant(Rh)), Threshold);
        item=best(Impact_set);
    }
Tr={$t \in D | t$fully support $Rh$}
While (Rh.Conf >Threshold) do
sanitize(Tr;item);
End
```

Figure 3.9: The Distortion Based Sanitization Algorithm

3.8 Data Quality Based Blocking Algorithm

In the Blocking based algorithms the idea is to substitute the value of an item supporting the rule we want to hide with a meaningless symbol. In this way, we introduce an uncertainty related to the question "How to interpret this value?". The final effect is then the same of the Distortion Based Algorithm. The strategy we propose here, as showed in Figure 3.10 is very similar to the one presented in the previous section; the only difference is how to compute the Impact on the Data Quality. For this reason the *Zero_Impact Algorithm* is exactly the same presented before for the DQDB Algorithm. However, some adjustments are necessary in order to implement the *Impact Item Rank Algorithm.* The majors modifications are related to the Number of Step Evaluation (that is of course different) and the Data Quality evaluation. In fact, in this case it does not make sense to evaluate the Accuracy Lack, because every time an item is changed, the new value is meaningless. For this reason in this case we consider the pair (Completeness, Consistency) and not the pair (Accuracy, Consistency). Figure 3.11 reports the new *Impact Item Rank Algorithm*.

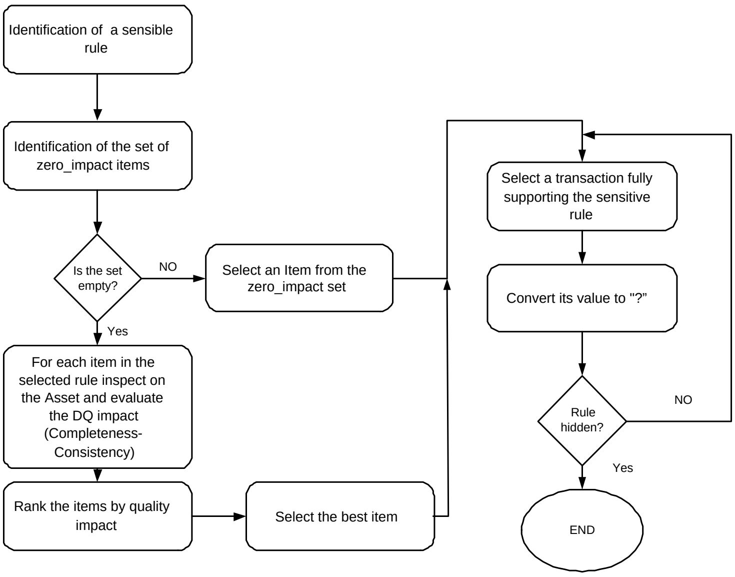
Identification of a sensible rule
Identification of the set of zero_impact items
Is the set empty?
NO
Select an Item from the zero_impact set
Yes
For each item in the selected rule inspect on the Asset and evaluate the DQ impact (Completeness-Consistency)
Rank the items by quality impact
Select the best item
Select a transaction fully supporting the sensitive rule
Convert its value to "?"
Rule hidden?
NO
Yes
END

Figure 3.10: Flow Chart of BBDB Algorithm

The last part of the algorithm is quite similar to the one presented for the DBDQ Algorithm. The only relevant difference is the insertion of the “?” value instead of the 0 value. From the complexity point of view, the complexity is the same of the complexity of the previous algorithm.

3.9 The Data Quality Strategy (Second Part)

Preliminary tests performed on these algorithms show that the impact on the DQ of the database is reduced. However, we note an interesting phenomenon. In what follows we give a description of this phenomena, an interpretation and a first solution

Consider a DataBase D to be sanitized. D contains a set of four sensitive rules. In order to protect these rules we normally execute the following steps:

1. We select a rule Rh from the pool of sensitive rules.
2. We apply our algorithm (DQDB or BBDQ).
3. We obtain a sanitized database in which the DQ is preserved (with respect to the rule Rh).

```
INPUT: the Asset Schema $A$ associated
to the target database,the rule $Rh$ to be hidden,
the $sup(Rh)$, the $sup(ant(Rh))$, the Threshold $Th$
OUTPUT: the set $Qitem$ ordered by Quality Impact

Begin
Qitem=$\emptyset$;
Confa=$\frac{Sup(Rh)}{Sup(ant(Rh))}$
Nstep.ant=countstep_ant(confa,Thresholds (Min..Max))
Nstep.post=countstep_post(confa,Thresholds (Min..Max))
For each item in $Rh$ do
{
    if ($item \in ant(Rh)$) then N=Nstep.ant;
        else N=Nstep.post;
    For each $AIS \in Asset$ do
    {
        node=recover_item(AIS,item);
        Complet_Cost=$node.Complet\_Weight * N$;
        Constr_Cost=Constr_surf(N,node);
        item.impact=$item.impact+$
            $+(Complet\_Cost * AIS.Complet\_weight)+$
            $+(Constr\_Cost * AIS.Constr\_weight)$;
    }
}
sort_by_impact(Items);
Return(Items);
End
```

Figure 3.11: Item Impact Rank Algorithm for the Blocking Based Algorithm

4. We reapply the process for all the rules contained in the set of *sensitive rules*.

What we observed in a large number of tests is that even if after every sanitization the DQ related to a single rule is preserved (or at least the damage is mitigated), at the end of the whole sanitization (every rule hidden), the database had a lower level of DQ that the one we expected.

Reflecting on the motivation of this behavior, we argue that the sanitization of a certain rule *Ra* may damage the DQ related to a previously sanitized rule *Rp* having for example some items in common.

For this reason here we propose an improvement to our strategy to address (but not to completely solve) this problem. It is important to notice that this is a common problem for all the PPDM Algorithm and it is not related only to the algorithm we proposed in the previous sections.

The new strategy is based on the following intuition: The Global DQ degradation is due to the fact that in some sanitization items are modified that were involved in other already sanitized rules. If we try to limit the number of different items modified during the whole process (i.e. during the sanitization of the whole rules set), we obtain a better level of DQ. An approach to limit the number of modified items is to identify the items supporting the biggest number of sensitive rules as the most suitable to be modified and then performing in a burst the sanitization of all the rules. The strategy we adopt can be summarized as follows:

1. We classify the items supporting the rules contained in the *sensitive set* analyzing how many rules simultaneously contain the same item (we count

the occurrences of the items in the different rules).

2. We identify, given the *sensitive set*, the Global Set of Zero_Impact Items.
3. We order the Zero_Impact Set by considering the previously counted occurrences.
4. The sanitization starts by considering the Zero_Impact Items that support the maximum number of Rules.
5. Simultaneously all the rules supported by the chosen item are sanitized and removed from the *sensitive set*, when they result hidden.
6. If the Zero_Impact Set is not sufficient to hide every sensitive rule, for the remaining items their DQ Impact is computed.
7. From the resulting classification, if several items exist with the same impact, the one supporting the highest number of rules is chosen.
8. The process continues until all the rules are hidden.

From a methodological point of view, we can divide the complete strategy we proposed into three main blocks:

- Pre Hiding phase
- Multi Zero Impact Hiding
- Multi Low Impact Hiding

This strategy can be easily applied to the algorithms presented before. For this reason we present in the remainder of this report the algorithms giving, at the end, a more general algorithm.

3.10 Pre-Hiding Phase

INPUT: The *sensitive Rules set Rs*, the list of the database items
OUTPUT: The list of the items supporting the rules sorted by occurrences number

```
Begin
Rsup=Rs;
while ($Rsup \neq \emptyset$) do
{
    select r from Rsup;
    for ($i = 0; i++; i \leq transactionmaxsize$) do
        if ($item[i] \in r$) then
        {
            item[i].count++;
            item[i].list=item[i].list$\cup r$;
        }
    Rsup=$Rsup - r$;
}
sort(item);
Return(item); End
```

Figure 3.12: The Item Occurrences classification algorithm

INPUT: The *sensitive Rules set Rs*, the list of the database items
OUTPUT: The list of the Zero Impact Items

```
Begin
Rsup=Rs;
Isup=list of all the items;
$Zitem = \emptyset$
while ($Rsup \neq \emptyset$) and ($Isup \neq \emptyset$) do
{
    select r from Rsup;
    for each ($item \in r$) do
    {
        if ($item \in Isup$) then
            if ($item \notin Asset$) then $Zitem = Zitem + item$;
        $Isup = Isup - item$
    }
    $Rsup = Rsup - r$
}
return(Zitem)
End
```

Figure 3.13: The Multi Rule Zero Impact Algorithm

This phase executes all the operations not specifically related to the hiding operations. More in details, in this phase, the items are classified according to the number of supported rules. Moreover the Zero_Impact Items are classified according to the same criteria. Figure 3.12 reports the algorithm used to discover the occurrences of the items. This algorithm takes as input the set of sensitive rules and the list of all the different items defined for the target database. For every rule contained in the set, we check if the rule is supported by every item contained in the Item_list. If this is the case, a counter associated with the item is incremented and an identifier associated with the rule is inserted into the list of the rules supported by a certain item. This information is used in one of the subsequent steps in order to optimize the computation. Once a rule has been checked with every item in the list, it is removed from the sensitive set. From a computational point of view, it is relevant to note that the entire process is executed a number of times equal to:

$$|sensitive\_set| * |Item\_List|$$

Figure 3.13 reports the algorithm used to identify the Zero_Impact Items for the *sensitive Rules set*. It is quite similar to the one presented in the previous section, the difference is just that it searches in a burst the Zero-Impact items for all the sensitive rules.

3.11 Multi Rules Zero Impact Hiding

The application of the two algorithms presented before returns two important information:

- The items supporting more than a rule, and more in details, the exact number of rules supported by these Items.

- The items supporting a $rule \in sensitive\_rules\_set$ that have the important property of being without impact on the DQ of the database.

Using these relevant information, we are then able to start the sanitization process for those rules supported by Zero Impact Items. The strategy we adopt is thus as follows:

1. We extract from the list of Items, in an ordered manner, the items already identified as Zero_Impact (the Izm set).

2. For each item contained in the new set, we build the set of the sensitive rules supported (Rss) and the set of transactions supporting one of the rules contained in Rss (Tss).

3. We then select a transaction in Tss and we perform the sanitization on this transaction (by using either blocking or distortion).

4. We recompute the support and the confidence value for the rules contained in the *Rss* set and if one of the rule results hidden, it is removed from the *Rss* set and from the *Rs* set.

5. These operations are repeated for every item contained in the *Izm* set.

The result of this approach can be:

- **A sanitized Data Base with an optimal DQ:** this is the case in which all sensitive rules are supported by Zero_Impact Items.
- **A partially Sanitized Data Base with an optimal DQ:** this is the case in which the *sensitive set* has a subset of rules not supported by Zero_Impact Items.

Obviously, in a real case, the first result rarely happens. For this reason in this we present the last step of our strategy, that allows us to obtain in any case a completely sanitized database

INPUT: The *sensitive Rules set* Rs, the list of the database items (with the Rule support), the list of Zero_Impact Items, the Database D
OUTPUT: A partially Sanitized Database PS

```
Begin
Izm=$\{item \in Itemlist|item \in Zitem\}$;
for $(i = 0, i++, i = |Izm|)$ do
{
    Rss=$\{r \in Rs|Izm[i] \in r\}$
    Tss=$\{t \in D|tfullysupportarule \in Rss\}$
    while$(Tss \neq \emptyset)$and$(Rss \neq \emptyset)$ do
    {
        select a transaction from Tss;
        $Tss = Tss - t$;
        Sanitize(t,Izm[i]);
        Recompute_Sup_Conf(Rss);
        $\forall r \in Rss|(r.sup < minsup)Or(r.conf < minconf)$ do
        {
            Rss=Rss-r;
            Rs=Rs-r;
        }}}
End
```

Figure 3.14: The Multi Rules Zero Impact Hiding Algorithm

3.12 Multi Rule Low Impact algorithm

In this section we consider the case in which we need to sanitize a database having a *sensitive Rules set* not supported by Zero_Impact Items. In this case, we have two possibilities: to use the Algorithm presented in the *data quality section* that has the advantage of being very efficient, or, if we prefer to maintain as much as possible a good DQ, to extend the concept of Multi-Rule approach to this particular case. The strategy we propose here can be summarized as follows:

1. We extract from the ordered list of items, only the items supporting the rules contained in the *sensitive Rules set*. Each item contains the list of the supported rules.
2. For each item and for each rule contained in the list of the supported rules the estimated DQ impact is computed.The Maximum Data Quality Impact estimated is associated to every item.
3. The Items are sorted by Max Impact and number of supported rules.
4. For each item contained in the new ordered set, we build the set of the sensitive rules supported (Rss) and the set of transactions supporting one of the rules contained in Rss (Tss).

5. We then select a transaction in Tss and we perform the sanitization on this transaction (blocking or distortion).
6. We recompute the support and the confidence value for the rules contained in the Rss set and if one of the rule results hidden, it is removed from the Rss set and from the Rs set.
7. These operations are repeated for every item contained in the Izm set until all rules are sanitized.

The resulting database will be **Completely Sanitized** and the damage related to the data quality will be mitigated. Figure 3.15 reports the algorithm. As it is possible to note, in the algorithm we do not specify if the sanitization is based on blocking or distortion. We consider this operation as a *black box*. The proposed strategy is completely independent from the type of sanitization used. In fact it is possible to substitute the code for blocking algorithm with the code for distortion algorithm without compromising the DQ properties we want to preserve with our algorithm. Figure 3.16 reports the global algorithm formalizing our strategy. More in details, taking as input to this algorithm the Asset description, the group of rules to be sanitized, the database and the thresholds under which consider hidden a rule, the algorithm computes the Zero-Impact set, and the Item-Occurrences rank. Then it will try to hide the rules modifying the Zero_Impact Items and finally, if there exist other rules to be hidden, the Low Impact Algorithm is invoked.

```
INPUT: the Asset Schema $A$ associated
to the target database,the sensitive Rules set $Rs$
to be hidden, the Thresholds $Th$, the list of items
OUTPUT: the sanitized database $SDB$

Begin
Iml=$\{item \in itemlist | itemsupportsaruler \in Rs\}$
for each $item \in Iml$ do
{
    for each $rule \in item.list$ do
    {
        N=computestepitem,thresholds;
        For each $AIS \in Asset$ do
        {
            node=recover_item(AIS,item);
            item.list.rule.cost=Item.list.rule.cost
                    +ComputeCost(node,N);
        }
        item.maxcost=maxcost(item);
    }
    Sort Iml by max cost and rule supported.
While($Rs \neq \emptyset$)do
{
    Rss=$\{r \in Rs | Iml[1] \in r\}$
    Tss=$\{t \in D | tfullysupportarule \in Rss\}$
    while($Tss \neq \emptyset$)and($Rss \neq \emptyset$) do
    {
        select a transaction from Tss;
        $Tss = Tss - t$;
        Sanitize(t,Iml[1]);
        Recompute_Sup_Conf(Rss);
        $\forall r \in Rss | (r.sup < minsup) Or$
            $(r.conf < minconf)$ do
        {
            Rss=Rss-r;
            Rs=Rs-r;
        }
    }
    Ilm=Ilm-Ilm[1];
}
} End
```

Figure 3.15: Low Data Quality Impact Algorithm

INPUT: The *sensitive Rules set Rs*, the Database D, the *Asset*
the thresholds
OUTPUT: A Sanitized Database PS

```
Begin
Item=Item_Occ_Class(Rs,Itemlist);
Zitem=Multi_Rule_Zero_Imp(Rs,Itemlist);
if ($Zitem \neq \emptyset$) then MR_zero_Impact_H(D,Rs,Item,Zitem);
if ($Rs \neq \emptyset$) then MR_low_Impact_H(D,Rs,Item);
End
```

Figure 3.16: The General Algorithm

Chapter 4

Conclusions

The problem of the sanitization impact on the quality of the database has been, to the best of our knowledge, never addressed by previous approaches to the PPDM. In this work, we have explored the concepts related to DQ. Moreover we have identified the most suitable set of parameters that can be used to represent the DQ in the context of PPDM. Previous approaches have addressed the PPDM problem not considering the intrinsic meaning of the information stored in a database. This lack is the main cause of the DQ problem. In fact the DQ is strongly related to the use of the data and then, indirectly to the meaning of the data. Starting from this consideration, we introduced a formal schema allowing us to represent the relevant information stored into a database. This schema is able to magnify the relevance of each attribute contained into a set of high level information, the relationships among the attributes and the relevance of the DQ parameters associated with the Information Schema. Based on this schema, we proposed a first strategy and two PPDM algorithms (distortion and blocking based) with the aim of obtaining a sanitization which is better with respect to DQ. We have then refined our strategy in order to hide simultaneously a set of sensible rules. On the basis of this new strategy, we have proposed then a suite of algorithms allowing us to build a most sophisticated PPDM Data Quality Based Algorithm.

Bibliography

[1] N. Adam and J. Worthmann, *Security-control methods for statistical databases: a comparative study.* ACM Comput. Surv., Volume 21(4), pp. 515-556, year 1989, ACM Press.

[2] D. Agrawal and C. C. Aggarwal, *On the Design and Quantification of Privacy Preserving Data Mining Algorithms.* In Proceedings of the 20*th* ACM Symposium on Principle of Database System, pp. 247-255, year 2001, ACM Press.

[3] R. Agrawal, T. Imielinski and A. Swami, *Mining Association Rules between Sets of Items in Large Databases.* Proceedings of ACM SIGMOD, pp. 207-216, May 1993, ACM Press.

[4] R. Agrawal and R. Srikant, *Privacy Preserving Data Mining.* In Proceedings of the ACM SIGMOD Conference of Management of Data, pp. 439-450, year 2000, ACM Press.

[5] R. Agrawal and R. Srikant, *Fast algorithms for mining association rules.* In Proceeding of the 20*th* International Conference on Very Large Databases, Santiago, Chile, June 1994, Morgan Kaufmann.

[6] G. M. AMDAHL, *Validity of the Single-Processor Approach to Achieving Large Scale Computing Capabilities.* AFIPS Conference Proceedings(April 1967),pp. 483-485, Morgan Kaufmann Publishers Inc.

[7] M. J. Atallah, E. Bertino, A. K. Elmagarmid, M. Ibrahim and V. S. Verykios, *Disclosure Limitation of Sensitive Rules.* In Proceedings of the IEEE Knolwedge and Data Engineering Workshop, pp. 45-52, year 1999, IEEE Computer Society.

[8] D. P. Ballou, H. L. Pazer, *Modelling Data and Process Quality in Multi Input, Multi Output Information Systems.* Management science, Vol. 31, Issue 2, pp. 150-162, (1985).

[9] Y. Bar-Hillel, *An examination of information theory.* Philosophy of Science, volume 22, pp.86-105, year 1955.

[10] E. Bertino, I. Nai Fovino and L. Parasiliti Provenza, *A Framework for Evaluating Privacy Preserving Data Mining Algorithms.* Data Mining and Knowledge Discovery Journal, year 2005, Kluwert.

[11] E.Bertino and I.Nai Fovino, *Information Driven Evaluation of Data Hiding Algorithms.* 7*th* International Conference on Data Warehousing and Knowledge Discovery. Copenaghen, August 2005, Springer-Verlag.

[12] N. M. Blachman, *The amount of information that y gives about X.* IEEE Truns. Inform. Theon. vol. IT-14, pp. 27-31. Jan. 1968, IEEE Press.

[13] L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification of Regression Trees.* Wadsworth International Group, year 1984.

[14] S. Brin, R. Motwani, J. D. Ullman and S. Tsur, *Dynamic itemset counting and implication rules for market basket data.* In Proc. of the ACM SIGMOD International Conference on Management of Data, year 1997, ACM Press.

[15] L. Chang and I. S. Moskowitz, *Parsimonious downgrading and decision trees applied to the inference problem.* In Proceedings of the 1998 New Security Paradigms Workshop, pp.82-89, year 1998, ACM Press.

[16] P. Cheeseman and J. Stutz, *Bayesian Classification (AutoClass): Theory and Results.* Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, year 1996.

[17] M. S. Chen, J. Han and P. S. Yu, *Data Mining: An Overview from a Database Perspective.* IEEE Transactions on Knowledge and Data Engineering, vol. 8 (6), pp. 866-883, year 1996, IEEE Educational Activities Department.

[18] F. Y. Chin and G. Ozsoyoglu, *Auditing and inference control in statistical databases.* IEEE Trans. Softw. Eng. SE-8, 6 (Apr.), pp. 574-582, year 1982, IEEE Press.

[19] F. Y. Chin and G. Ozsoyoglu, *Statistical database design.* ACM Trans. Database Syst. 6, 1 (Mar.), pp. 113-139, year 1981, ACM Press.

[20] L. H. Cox, *Suppression methodology and statistical disclosure control.* J. Am. Stat. Assoc. 75, 370 (June), pp. 377-385, year 1980.

[21] E. Dasseni, V. S. Verykios, A. K. Elmagarmid and E. Bertino, *Hiding Association Rules by using Confidence and Support.* in proceedings of the 4*th* Information Hiding Workshop, pp. 369383, year 2001, Springer-Verlag.

[22] D. Defays, *An efficient algorithm for a complete link method.* The Computer Journal, 20, pp. 364-366, 1977.

[23] D. E. Denning and J. Schlorer, *Inference control for statistical databases.* Computer 16 (7), pp. 69-82, year 1983 (July), IEEE Press.

[24] D. Denning, *Secure statistical databases with random sample queries.* ACM TODS, 5, 3, pp. 291-315, year 1980.

[25] D. E. Denning, *Cryptography and Data Security.* Addison-Wesley, Reading, Mass. 1982.

[26] V. Dhar, *Data Mining in finance: using counterfactuals to generate knowledge from organizational information systems.* Information Systems, Volume 23, Number 7, pp. 423-437(15), year 1998.

[27] J. Domingo-Ferrer and V. Torra, *A Quantitative Comparison of Disclosure Control Methods for Microdata.* Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, pp. 113-134, P. Doyle, J. Lane, J. Theeuwes, L. Zayatz ed., North-Holland, year 2002.

[28] P. Domingos and M. Pazzani, *Beyond independence: Conditions for the optimality of the simple Bayesian classifier.* Proceedings of the Thirteenth International Conference on Machine Learning, pp. 105-112, San Francisco, CA, year 1996, Morgan Kaufmann.

[29] P. Drucker, *Beyond the Information Revolution.* The Atlantic Monthly, 1999.

[30] P. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* Wiley, year 1973, New York.

[31] G. T. Duncan, S. A. Keller-McNulty and S. L. Stokes, *Disclosure risks vs. data utility: The R-U confidentiality map.* Tech. Rep. No. 121. National Institute of Statistical Sciences. 2001

[32] C. Dwork and K. Nissim, *Privacy preserving data mining in vertically partitioned database.* In Crypto 2004, Vol. 3152, pp. 528-544.

[33] D. L. EAGER, J. ZAHORJAN and E. D. LAZOWSKA, *Speedup Versus Efficiency in Parallel Systems.* IEEE Trans. on Computers, C-38, 3 (March 1989), pp. 408-423, IEEE Press.

[34] L. Ertoz, M. Steinbach and V. Kumar, *Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data.* In Proceeding to the SIAM International Conference on Data Mining, year 2003.

[35] M. Ester, H. P. Kriegel, J. Sander and X. XU, *A density-based algorithm for discovering clusters in large spatial databases with noise.* In Proceedings of the 2nd ACM SIGKDD, pp. 226-231, Portland, Oregon, year 1996, AAAI Press.

[36] A. Evfimievski, *Randomization in Privacy Preserving Data Mining.* SIGKDD Explor. Newsl., vol. 4, number 2, year 2002, pp. 43-48, ACM Press.

[37] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, *Privacy Preserving Mining of Association Rules.* In Proceedings of the 8*th* ACM SIGKDDD International Conference on Knowledge Discovery and Data Mining, year 2002, Elsevier Ltd.

[38] S. E. Fahlman and C. Lebiere, *The cascade-correlation learning architecture.* Advances in Neural Information Processing Systems 2, pp. 524-532. Morgan Kaufmann, year 1990.

[39] R. P. Feynman, R. B. Leighton and M. Sands, *The Feynman Lectures on Physics, v I.* Reading, Massachusetts: Addison-Wesley Publishing Company, year 1963.

[40] S. Fortune and J. Wyllie, *Parallelism in Random Access Machines.* Proc. Tenth ACM Symposium on Theory of Computing(1978), pp. 114-118, ACM Press.

[41] W. Frawley, G. Piatetsky-Shapiro and C. Matheus, *Knowledge Discovery in Databases: An Overview.* AI Magazine, pp. 213-228, year 1992.

[42] S. P. Ghosh, *An application of statistical databases in manufacturing testing.* IEEE Trans. Software Eng. 1985. SE-11, 7, pp. 591-596, IEEE press.

[43] S.P.Ghosh, *An application of statistical databases in manufacturing testing.* In Proceedings of IEEE COMPDEC Conference, pp. 96-103,year 1984, IEEE Press.

[44] S. Guha, R. Rastogi and K. Shim, *CURE: An efficient clustering algorithm for large databases.* In Proceedings of the ACM SIGMOD Conference, pp. 73-84, Seattle, WA. 1998, ACM Press.

[45] S. Guha, R. Rastogi and K. Shim, *ROCK: A robust clustering algorithm for categorical attributes.* In Proceedings of the 15*th* ICDE, pp. 512-521, Sydney, Australia, year 1999, IEEE Computer Society.

[46] J. Han and M. Kamber, *Data Mining: Concepts and Techniques.* The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, August 2000.

[47] J. Han, J. Pei and Y. Yin, *Mining frequent patterns without candidate generation.* In Proceeding of the 2000 ACM-SIGMOD International Conference on Management of Data, Dallas, Texas, USA, May 2000, ACM Press.

[48] M. A. Hanson and R. L. Brekke, *Workload management expert system - combining neural networks and rule-based programming in an operational application.* In Proceedings Instrument Society of America, pp. 1721-1726, year 1988.

[49] J. Hartigan and M. Wong, *Algorithm AS136: A k-means clustering algorithm.* Applied Statistics, 28, pp. 100-108, year 1979.

[50] A. Hinneburg and D. Keim, *An efficient approach to clustering large multimedia databases with noise.* In Proceedings of the 4*th* ACM SIGKDD, pp. 58-65, New York, year 1998, AAAI Press.

[51] T. Hsu, C. Liau and D.Wang, *A Logical Model for Privacy Protection.* Lecture Notes in Computer Science, Volume 2200, Jan 2001, pp. 110-124, Springer-Verlag.

[52] IBM Synthetic Data Generator.
http://www.almaden.ibm.com/software/quest/Resources/datasets/syndata.html

[53] M. Kantarcioglu and C. Clifton, *Privacy Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data.* In Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 24-31, year 2002, IEEE Educational Activities Department.

[54] G. Karypis, E. Han and V. Kumar, *CHAMELEON: A hierarchical clustering algorithm using dynamic modeling.* COMPUTER, 32, pp. 68-75, year 1999.

[55] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley and Sons, New York, year 1990.

[56] W. Kent, *Data and reality.* North Holland, New York, year 1978.

[57] S. L. Lauritzen, *The em algorithm for graphical association models with missing data.* Computational Statistics and Data Analysis, 19 (2), pp. 191-201, year 1995, Elsevier Science Publishers B. V.

[58] W. Lee and S. Stolfo, *Data Mining Approaches for Intrusion Detection.* In Proceedings of the Seventh USENIX Security Symposium (SECURITY '98), San Antonio, TX, January 1998.

[59] A. V. Levitin and T. C. Redman, *Data as resource: properties, implications and prescriptions.* Sloan Management review, Cambridge, Vol. 40, Issue 1, pp. 89-101, year 1998.

[60] Y. Lindell and B. Pinkas, *Privacy Preserving Data Mining.* Journal of Cryptology, vol. 15, pp. 177-206, year 2002, Springer Verlag.

[61] R. M. Losee, *A Discipline Independent Definition of Information.* Journal of the American Society for Information Science 48 (3), pp. 254-269, year 1997.

[62] M. Masera, I. Nai Fovino, R. Sgnaolin *A Framework for the Security Assessment of Remote Control Applications of Critical Infrastructure* 29th ESReDA Seminar “Systems Analysis for a More Secure World”, year 2005

[63] G. MClachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker, New York, year 1988.

[64] M. Mehta, J. Rissanen and R. Agrawal, *MDL-based decision tree pruning.* In Proc. of KDD, year 1995, AAAI Press.

[65] G. L. Miller, *Resonance, Information, and the Primacy of Process: Ancient Light on Modern Information and Communication Theory and Technology.* PhD thesis, Library and Information Studies, Rutgers, New Brunswick, N.J., May 1987.

[66] I. S. Moskowitz and L. Chang, *A decision theoretical based system for information downgrading.* In Proceedings of the 5*th* Joint Conference on Information Sciences, year 2000, ACM Press.

[67] S. R. M. Oliveira and O. R. Zaiane, *Toward Standardization in Privacy Preserving Data Mining.* ACM SIGKDD 3rd Workshop on Data Mining Standards, pp. 7-17, year 2004, ACM Press.

[68] S. R. M. Oliveira and O. R. Zaiane, *Privacy Preserving Frequent Itemset Mining.* Proceedings of the IEEE international conference on Privacy, security and data mining, pp. 43-54, year 2002, Australian Computer Society, Inc.

[69] S. R. M. Oliveira and O. R. Zaiane, *Privacy Preserving Clustering by Data Transformation.* In Proceedings of the 18*th* Brazilian Symposium on Databases, Manaus, Amazonas, Brazil, pp. 304-318, year 2003.

[70] K. Orr, *Data Quality and System Theory.* Comm. of the ACM, Vol. 41, Issue 2, pp. 66-71, Feb. 1998, ACM Press.

[71] M. A. Palley and J. S. Simonoff, *The use of regression methodology for compromise of confidential information in statistical databases.* ACM Trans. Database Syst. 12,4 (Dec.), pp. 593-608, year 1987.

[72] J. S. Park, M. S. Chen and P. S. Yu, *An Effective Hash Based Algorithm for Mining Association Rules.* Proceedings of ACM SIGMOD, pp. 175-186, May, 1995, ACM Press.

[73] Z. Pawlak, *Rough SetsTheoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, 1991.

[74] G. Piatetsky-Shapiro, *Discovery, analysis, and presentation of strong rules.* Knowledge Discovery in Databases, pp. 229-238, AAAI/MIT Press, year 1991.

[75] A. D. Pratt, *The Information of the Image.* Ablex, Norwood, NJ, 1982.

[76] J. R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, year 1993.

[77] J. R. Quinlan, *Induction of decision trees.* Machine Learning, vol. 1, pp. 81-106, year 1986, Kluwer Academic Publishers.

[78] R. Rastogi and S. Kyuseok, *PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning.* Data Mining and Knowledge Discovery, vol. 4, n.4, pp. 315-344, year 2000.

[79] R. Rastogi and K. Shim, *Mining Optimized Association Rules with Categorical and Numeric Attributes.* Proc. of International Conference on Data Engineering, pp. 503-512, year 1998.

[80] H. L. Resnikoff, *The Illusion of Reality.* Springer-Verlag, New York, 1989.

[81] S. J. Rizvi and J. R. Haritsa, *Maintaining Data Privacy in Association Rule Mining.* In Proceedings of the 28*th* International Conference on Very Large Databases, year 2003, Morgan Kaufmann.

[82] S. J. Russell, J. Binder, D. Koller and K. Kanazawa, *Local learning in probabilistic networks with hidden variables.* In International Joint Conference on Artificial Intelligence, pp. 1146-1152, year 1995, Morgan Kaufmann.

[83] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Learning internal representations by error propagation.* Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations pp. 318–362, Cambridge, MA: MIT Press, year 1986.

[84] G. Sande, *Automated cell suppression to reserve confidentiality of business statistics.* In Proceedings of the 2nd International Workshop on Statistical Database Management, pp. 346-353, year 1983.

[85] A. Savasere, E. Omiecinski and S. Navathe, *An efficient algorithm for mining association rules in large databases.* In Proceeding of the Conference on Very Large Databases, Zurich, Switzerland, September 1995, Morgan Kaufmann.

[86] J. Schlorer, *Information loss in partitioned statistical databases.* Comput. J. 26, 3, pp. 218-223, year 1983, British Computer Society.

[87] C. E. Shannon, *A Mathematical Theory of Communication.* Bell System Technical Journal, vol. 27,(July and October),1948, pp.379423, pp. 623-656.

[88] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, Ill. 1949.

[89] A. Shoshani, *Statistical databases: characteristics,problems, and some solutions.* Proceedings of the Conference on Very Large Databases (VLDB), pp.208-222, year 1982, Morgan Kaufmann Publishers Inc.

[90] R. SIBSON, *SLINK: An optimally efficient algorithm for the single link cluster method.* Computer Journal, 16, pp. 30-34, year 1973.

[91] P. Smyth and R. M. Goodman, *An information theoretic Approach to Rule Induction from databases.* IEEE Transaction On Knowledge And Data Engineering, vol. 3, n.4, August,1992, pp. 301-316, IEEE Press.

[92] L. Sweeney, *Achieving* k-*Anonymity Privacy Protection using Generalization and Suppression.* International Jurnal on Uncertainty, Fuzzyness and Knowledge-based System, pp. 571-588, year 2002, World Scientific Publishing Co., Inc.

[93] R. Srikant and R. Agrawal, *Mining Generalized Association Rules.* Proceedings of the 21*th* International Conference on Very Large Data Bases, pp. 407-419, September 1995, Morgan Kaufmann.

[94] G. K. Tayi, D. P. Ballou, *Examining Data Quality.* Comm. of the ACM, Vol. 41, Issue 2, pp. 54-58, year 1998, ACM Press.

[95] M. Trottini, *A Decision-Theoretic Approach to data Disclosure Problems.* Research in Official Statistics, vol. 4, pp. 722, year 2001.

[96] M. Trottini, *Decision models for data disclosure limitation.* Carnegie Mellon University, Available at `http://www.niss.org/dgii/TR/ThesisTrottini -final.pdf`, year 2003.

[97] University of Milan - Computer Technology Institute - Sabanci University *Codmine* IST project. 2002-2003.

[98] J. Vaidya and C. Clifton, *Privacy Preserving Association Rule Mining in Vertically Partitioned Data.* In Proceedings of the 8*th* ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 639-644, year 2002, ACM Press.

[99] V. S. Verykios, E. Bertino, I. Nai Fovino, L. Parasiliti, Y. Saygin, Y. Theodoridis, *State-of-the-art in Privacy Preserving Data Mining.* SIGMOD Record, 33(1) pp. 50-57, year 2004, ACM Press.

[100] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin and E. Dasseni, *Association Rule Hiding.* IEEE Transactions on Knowledge and Data Engineering, year 2003, IEEE Educational Activities Department.

[101] C. Wallace and D. Dowe, *Intrinsic classification by MML. The Snob program.* In the Proceedings of the 7*th* Australian Joint Conference on Artificial Intelligence, pp. 37- 44, UNE, World Scientific Publishing Co., Armidale, Australia, 1994.

[102] G. J. Walters, *Philosophical Dimensions of Privacy: An Anthology.* Cambridge University Press, year 1984.

[103] G. J. Walters, *Human Rights in an Information Age: A Philosophical Analysis.* chapter 5, University of Toronto Press, year 2001.

[104] Y. Wand and R. Y. Wang, *Anchoring Data Quality Dimensions in Ontological Foundations.* Comm. of the ACM, Vol. 39, Issue 11, pp. 86-95, Nov. 1996, ACM Press.

[105] R. Y. Wang and D. M. Strong, *Beyond Accuracy: what Data Quality Means to Data Consumers.* Journal of Management Information Systems Vol. 12, Issue 4, pp. 5-34, year 1996.

[106] L. Willenborg and T. De Waal, *Elements of statistical disclosure control.* Lecture Notes in Statistics Vol.155, Springer Verlag, New York.

[107] N. Ye and X. Li, *A Scalable Clustering Technique for Intrusion Signature Recognition.* 2001 IEEE Man Systems and Cybernetics Information Assurance Workshop, West Point, NY, June 5-6, year 2001, IEEE Press.

[108] M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li, *New algorithms for fast discovery of association rules* In Proceeding of the 8rd International Conference on KDD and Data Mining, Newport Beach, California, August 1997, AAAI Press.

European Commission

EUR 23070 EN – Joint Research Centre – Institute for the Protection and Security of the Citizen
Title: Privacy Preserving Data Mining, a Data Quality Approach
Authors: Igor Nai Fovino and Marcelo Masera
Luxembourg: Office for Official Publications of the European Communities
2008 – 56 pp. – 21 x 29.7 cm
EUR – Scientific and Technical Research series – ISSN 1018-5593

Abstract

Privacy is one of the most important properties an information system must satisfy.
A relatively new trend shows that classical access control techniques are not sufficient to guarantee privacy when datamining techniques are used. Privacy Preserving Data Mining (PPDM) algorithms have been recently introduced with the aim of sanitizing the database in such a way to prevent the discovery of sensible information (e.g. association rules). A drawback of such algorithms is that the introduced sanitization may disrupt the quality of data itself. In this report we introduce a new methodology and algorithms for performing useful PPDM operations, while preserving the data quality of the underlying database.

The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.

JRC
EUROPEAN COMMISSION

Publications Office
Publications.eu.int