Data Mining and Knowledge Discovery, 11, 121–154, 2005
© 2005 Springer Science+Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s10618-005-0006-6

A Framework for Evaluating Privacy Preserving Data Mining Algorithms*

ELISA BERTINO bertino@cerias.purdue.edu
CERIAS and CS Department, Purdue University

IGOR NAI FOVINO nai@dico.unimi.it
LOREDANA PARASILITI PROVENZA parasiliti@dico.unimi.it
Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano

Editor: Geoff Webb

Received October 27, 2004; Accepted March 28, 2005

Published online: 18 August 2005

Abstract. Recently, a new class of data mining methods, known as *privacy preserving data mining* (PPDM) algorithms, has been developed by the research community working on security and knowledge discovery. The aim of these algorithms is the extraction of relevant knowledge from large amount of data, while protecting at the same time sensitive information. Several data mining techniques, incorporating privacy protection mechanisms, have been developed that allow one to hide sensitive itemsets or patterns, before the data mining process is executed. Privacy preserving classification methods, instead, prevent a miner from building a classifier which is able to predict sensitive data. Additionally, privacy preserving clustering techniques have been recently proposed, which distort sensitive numerical attributes, while preserving general features for clustering analysis. A crucial issue is to determine which ones among these privacy-preserving techniques better protect sensitive information. However, this is not the only criteria with respect to which these algorithms can be evaluated. It is also important to assess the quality of the data resulting from the modifications applied by each algorithm, as well as the performance of the algorithms. There is thus the need of identifying a comprehensive set of criteria with respect to which to assess the existing PPDM algorithms and determine which algorithm meets specific requirements.

In this paper, we present a first evaluation framework for estimating and comparing different kinds of PPDM algorithms. Then, we apply our criteria to a specific set of algorithms and discuss the evaluation results we obtain. Finally, some considerations about future work and promising directions in the context of privacy preservation in data mining are discussed.

1. Introduction

Data mining technology has been developed with the goal of providing tools for automatically and intelligently transforming large amount of data in knowledge relevant to users. The extracted knowledge, often expressed in form of association rules, decision trees or clusters, allows one to find interesting patterns and regularities deeply buried in the data, that are meant to facilitate decision making processes. Such a knowledge discovery

*The work reported in this paper has been partially supported by the EU under the IST Project CODMINE and by the Sponsors of CERIAS.

process, however, can also return sensitive information about individuals, compromising the individual's right to privacy. Moreover, data mining techniques can reveal critical information about business transactions, compromising the free competition in a business setting. Thus, there is a strong need to prevent disclosure not only of confidential personal information, but also of knowledge which is considered sensitive in a given context. For this reason, recently much research effort has been devoted to addressing the problem of privacy preserving in data mining. As a result, several data mining techniques, incorporating privacy protection mechanisms, have been developed based on different approaches. For instance, various sanitization techniques have been proposed for hiding sensitive items or patterns that are based on removing reserved information or inserting noise into data. Privacy preserving classification methods, instead, prevent a miner from building a classifier able to predict sensitive data. Additionally, privacy preserving clustering techniques have been recently proposed, which distort sensitive numerical attributes, while preserving general features for clustering analysis. Given the number of different privacy preserving data mining (PPDM) techniques that have been developed over the last years, there is an emerging need of moving toward standardization in this new research area, as discussed in Oliveira and Zaiane (2004). We believe that one step toward this essential process is the definition of a framework identifying the various parameters which characterize a PPDM algorithm, thus making it possible to assess and compare such techniques according to a fixed set of evaluation criteria. Because all the various techniques differ among each other with respect to a number of criteria, like performance, data quality, privacy level, it is important to provide a systematic and comprehensive framework for their evaluation. In many cases, no technique is better than the other ones with respect to all criteria. Thus, one has to select the privacy preserving technique based on the criterion to be optimized. A framework like the one developed here is thus essential in order to select the privacy preserving technique which is more adequate based on the data and the requirements of the application domain.

A major feature of PPDM techniques is that they usually entail modifications to the data in order to sanitize them from sensitive information (both private data items and complex data correlations) or anonymize them with some uncertainty level. Therefore, in evaluating a PPDM algorithm it is important to assess the quality of the transformed data. To do this, we need methodologies for the assessment of the quality of data, intended as the state of the individual items in the database resulting from the application of a privacy preserving technique, as well as the quality of the information that is extracted from the modified data by using a given data mining method. The former notion of data quality is strictly related to the use the data are intended for. Moreover, some of those algorithms can be computationally very expensive and thus cannot be used when very large sets of data need to be frequently released. Therefore, in addition to data quality, performance also needs to be carefully assessed. Other aspects, like scalability, need also to be taken into account since the data collected, stored and managed for the mining process grow enormously. We thus clearly need a comprehensive evaluation framework characterized by several metrics relevant for assessing the various aspects of PPDM algorithms. In this paper, we present such a framework which allows one to compare the various privacy preserving techniques on a common platform. The framework consists of a number of evaluation criteria and a set of tools for data pre-processing and PPDM algorithm evaluation. The framework has been extensively used for assessing a suite of PPDM algorithms developed as part of the CODMINE project (CODMINE, 2002–2003).

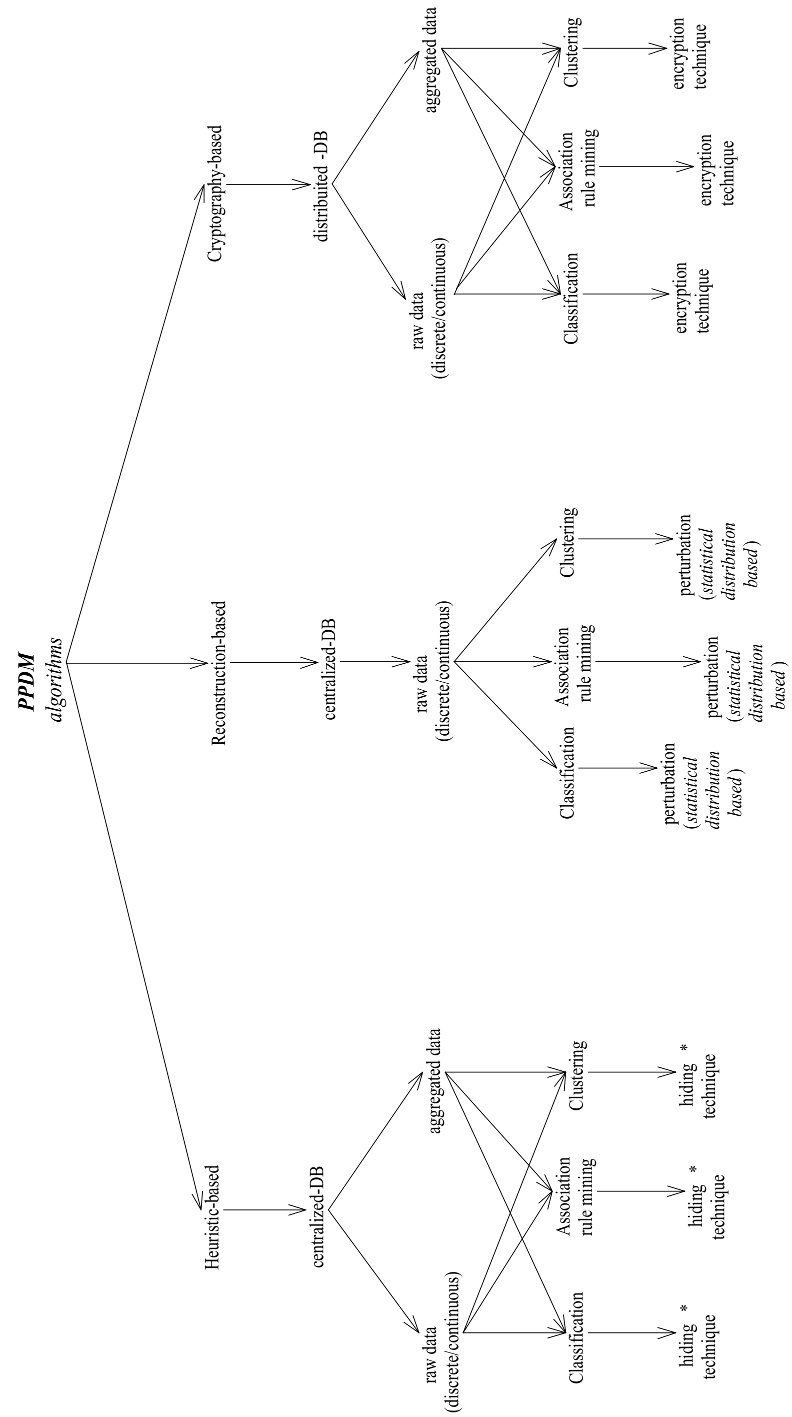
The problem of disclosing private information when partially or totally releasing data storage is also addressed in the area of statistical databases. The statistical disclosure control (SDC) aims at protecting individual data, referred to as *microdata* according to the SDC terminology, when releasing some relevant statistics by means of statistics-based techniques, some of which are also adopted in the area of data mining. Domingo-Ferrer and Torra (2002) have analyzed various SDC methods and compared some of them on the basis of some relevant evaluation measures they have also developed. Given the relations between these two research areas, we will also analyze and compare some of these methods along with the metrics used for evaluating them.

The remainder of this paper is organized as follows. Section 2 reviews different PPDM methods focusing on the evaluation criteria. Section 3 describes the evaluation framework we have developed for assessing various PPDM algorithms. Section 4 shows how such a framework has been used in the evaluation of a particular kind of PPDM algorithms, known as *data fuzzification* techniques. Finally, Section 5 presents considerations about future extensions and promising directions in the context of privacy preserving data mining.

2. Background and related work

Recent research in the area of privacy preserving data mining has devoted much effort to determine a trade-off between the right to privacy and the need of knowledge discovery, which is crucial in order to improve decision-making processes and other human activities. Such research has resulted in several approaches to the evaluation of privacy preserving techniques. In this section, we first present a taxonomy of the PPDM algorithms that have been proposed based on a classification presented in Verykios et al. (2004). We then present a brief review of the major work in this area, focusing on the metrics used in each approach to evaluate the proposed privacy preservation methods.

Verykios et al. (2004) analyze the state-of-the-art in the area of PPDM, classifying the proposed privacy preservation techniques according to five different dimensions: (i) data distribution (centralized or distributed); (ii) the modification applied to the data (encryption, perturbation, generalization, and so on) in order to sanitize them; (iii) the data mining algorithm which the privacy preservation technique is designed for; (iv) the data type (single data items or complex data correlations) that needs to be protected from disclosure; (v) the approach adopted for preserving privacy (heuristic, reconstruction or cryptography-based approaches). Figure 1 shows a taxonomy of the existing PPDM algorithms according to those dimensions. Obviously, it represents a first organization in this new area and does not cover all the possible PPDM algorithms. However, it gives one overview of the algorithms that have been proposed so far, focusing on their main features. While heuristic and reconstruction-based techniques are mainly conceived for centralized datasets, cryptography based algorithms are designed for protecting privacy in a distributed scenario by using encryption techniques. Reconstruction-based algorithms recently proposed aim at hiding sensitive raw data by applying perturbation techniques based on probability distributions. Moreover, several heuristic-based approaches for hiding



*hiding technique = {perturbation, blocking, swapping, aggregation, generalization, sampling}

Figure 1. A taxonomy of the developed PPDM algorithms.

both raw and aggregated data through a hiding technique (perturbation, blocking, data swapping, aggregation, generalization and sampling) have been developed, first, in the context of association rule mining and classification and, more recently, for clustering techniques. Now, we briefly describe some of the algorithms proposed in the PPDM area.

Oliveira and Zaiane (2002) propose a heuristic-based framework for preserving privacy in mining frequent itemsets. They focus on hiding a set of frequent patterns, containing highly sensitive knowledge. They propose a set of sanitized algorithms, that only remove information from a transactional database, also known in the SDC area as *non-perturbative* algorithms, unlike those algorithms, that modify the existing information by inserting noise into the data, referred to as *perturbative* algorithms. The algorithms proposed by Oliveira and Zaiane rely on a item-restriction approach, in order to avoid the addition of noise to the data and limit the removal of real data. In the evaluation of the proposed algorithms they introduce some measures quantifying the effectiveness and the efficiency of their algorithms. The first parameter is evaluated in terms of: *Hiding Failure*, that is, the percentage of restrictive patterns that are discovered from the sanitized database; *Misses Cost*, that is, the percentage of non-restrictive patterns that are hidden after the sanitization process; *Artifactual Pattern*, measured in terms of the percentage of discovered patterns that are artifacts. The specific algorithms proposed by Oliveira and Zaiane do not introduce any artifactual patterns, whereas with respect to the hiding failure and misses cost parameters, it turns out that the more restrictive patterns are hidden, the more legitimate patterns are missed. The specification of a *disclosure threshold* ϕ, representing the percentage of sensitive transactions that are not sanitized, allows one to find a balance between the hiding failure and the number of misses. The efficiency of the algorithms is measured in terms of CPU time, by first keeping constant both the size of the database and the set of restrictive patterns, and then by increasing the size of the input data in order to assess the algorithm scalability. Moreover, Oliveira and Zaiane propose three different methods to measure the *dissimilarity* between the original and sanitized databases. The first method is based on the difference between the frequency histograms of the original and the sanitized databases. The second method is based on computing the difference between the sizes of the sanitized database and the original one. The third method is based on a comparison between the contents of two databases.

In Sweeney (2002), instead, Sweeney proposes a heuristic-based approach for protecting raw data through generalization and suppression techniques. The methods she proposes provide *K-Anonymity*. Roughly speaking a database is K-anonymous with respect to some attributes if there exist at least k transactions in the database for each combination of the attribute values. A database A can be converted into a new database A^1 that guarantees the K-Anonymity property for a sensible attribute by performing some generalizations on the values of the target attributes. As result, such attributes are susceptible to *cell distortion* due to the different level of generalization applied in order to achieve K-Anonymity. Sweeney measures the cell distortion as the ratio of the domain of the attribute to the height of the attribute generalization which is a hierarchy. In the same article the concept of *precision* is also introduced. Given a table T, the *precision* represents the information loss incurred by the conversion process from a table T to a K-Anonymous Table T^k. More in detail the *precision* of a table T^k is measured as one minus the sum of all cell distortions, normalized by the total number of cells. In some way, the *precision* is a measure of the data quality or the data utility of the released table, specifically conceived for PPDM algorithms adopting generalization techniques for hiding sensitive information.

A reconstruction-based technique, instead, is proposed in R. Agrawal and Srikant (2000) for estimating the probability distribution of original numeric data values, in order to build a decision tree classifier from perturbed training data. They also introduce a quantitative measure to evaluate the amount of privacy offered by a method and evaluate the proposed method against this measure. The privacy provided by a reconstruction-based technique is measured by evaluating how closely the original values of a modified attribute can be determined. More specifically, if one can estimate with c% confidence that a value x lies in an interval, then the width of such interval defines the amount of privacy with a c% confidence level. They also assess the accuracy of the proposed algorithms for Uniform and Gaussian perturbation and for fixed privacy level. The reconstruction-based approach proposed by Agrawal and Aggarwal (2001) is based on an Expectation Maximization (EM) algorithm for distribution reconstruction, which converges to the maximum likelihood estimate of the original distribution on the perturbed data. The metrics they propose for the estimation of their methods, provide a quantification and a measurement of privacy and information loss. Unlike the approach in Agrawal and Srikant (2000), the privacy metric proposed by Agrawal and Aggarwal takes into account the fact that both the perturbed individual record and the reconstructed distribution are available to the user as well as the perturbing distribution, as it is specified in Evfimievski (2002). This metric is based on the concept of mutual information between the original and perturbed records. The average conditional privacy of an attribute A given some other information, modeled with a random variable B, is defined as $2^{h(A|B)}$, where $h(A/B)$ is the conditional differential entropy of A given B representing a measure of uncertainty inherent in the value of A, given the value of B. The information loss, instead, measures the lack of precision in estimating the original distribution from the perturbed data. It is defined as half the expected value of the L_1-norm between the original distribution and the reconstructed one. The proposed metric for evaluating information loss is related to the amount of mismatch between the original distribution and its estimate in terms of area. Both the proposed metrics are universal in the sense that they can be applied to any reconstruction algorithm, independently from the particular data mining task applied.

Evfimievski et al. (2002) propose a framework for mining association rules from transactions consisting of categorical items, where the data has been randomized to preserve privacy of individual transactions, while ensuring at the same time that only true associations are mined. They also provide a formal definition of privacy breaches and a class of randomization operators that are much more effective in limiting breaches than uniform randomization. According to Definition 4 from Evfimievski et al. (2002), an itemset A results in a privacy breach of level ρ if the probability that an item in A belongs to a non randomized transaction, given that A is included in a randomized transaction, is greater or equal to ρ. In some scenarios, being confident that an item be not present in the original transaction may also be considered a privacy breach. In order to evaluate the privacy breaches, the approach taken by Evfimievski et al. is to count the occurrences of an itemset in a randomized transaction and in its sub-items in the corresponding non randomized transaction. Out of all sub-items of an itemset, the item causing the worst privacy breach is chosen. Then, for each combination of transaction size and itemset size, the worst and the average value of this breach level are computed over all frequent itemsets. Finally, the itemset size giving the worst value for each of these two values is selected.

Another reconstruction-based technique is proposed by Rivzi and Haritsa (2002). They propose a distortion method to pre-process the data before executing the mining process. The privacy measure they propose deals with the probability with which the user's distorted entries can be reconstructed. Their goal is to ensure privacy at the level of individual entries in each customer tuple. In other words, the authors estimate the probability that a given 1 or 0 in the true matrix representing the transactional database can be reconstructed, even if for many applications the 1's and 0's values do not need the same level of privacy.

A cryptography-based technique is proposed by Kantarcioglu and Clifton (2002). They specifically address the problem of secure mining of association rules over horizontally partitioned data, using cryptographic techniques to minimize the information shared. Their solution is based on the assumption that each party first encrypts its own itemsets using commutative encryption, then the already encrypted itemsets of every other party. Later on, an initiating party transmits its frequency count, plus a random value, to its neighbor, which adds its frequency count and passes it on to other parties. Finally, a secure comparison takes place between the final and initiating parties to determine if the final result is greater than the threshold plus the random value. The proposed methods are evaluated in terms of communication and computation costs. The communication cost is expressed in terms of the number of messages exchanged among the sites, that are required by the protocol for securely counting the frequency of each rule. The computation cost, instead, is expressed in terms of the number of encryption and decryption operations required by the specific algorithm. Another cryptography-based approach is described in Vaidya and Clifton (2002). Such approach addresses the problem of association rule mining in vertically partitioned data. In other words, its aim is to determine the item frequency when transactions are split across different sites, without revealing the contents of individual transactions. A security and communication analysis is also presented. In particular, the security of the protocol for computing the scalar product is analyzed. The total communication cost depends on the number of candidate itemsets and can best be expressed as a constant multiple of the *I/O* cost of the apriori algorithm.

In the context of statistical disclosure control a large number of methods, called *masking* methods in the SDC jargon, have been developed to preserve individual privacy when releasing aggregated statistics on data, and more specifically to anonymize the released statistics from those data items that can identify one among the individual entities (person, household, business, etc.) whose features are described by the statistics, also taking into account, additionally, related information publicly available (Willenborg and De Waal 2001). In Domingo-Ferrer and Torra (2002) a description of the most relevant masking methods proposed so far is presented. Among the perturbative methods specifically designed for continuous data, the following masking techniques are described: additive noise, data distortion by probability distribution, resampling, microaggregation, rank swapping, and so on. For categorical data both perturbative and non-perturbative methods are presented. The top-coding and bottom-coding techniques are both applied to ordinal categorical variables; they recode, respectively, the first/last p values of a variable into a new category. The global-recoding technique, instead, recodes the p lowest frequency categories into a single one. All these masking methods are assessed according to the two main parameters: the *information loss* and the *disclosure risk*, that is, the risk that a piece of information be linked to a specific individual. Several metrics are presented in the paper for assessing the *information loss* and the *disclosure risk* given by a SDC method. Additionally, in order to provide a trade-off level between these two metrics, a score is defined that gives the same importance to disclosure risk and information loss.

3. Evaluation framework

In this section, we present the evaluation framework we have identified and the related benchmarks for estimating PPDM algorithms. Before describing the selected evaluation criteria, we briefly describe the goals of a PPDM algorithm and explain what we mean by the general term of "Privacy".

In our society the *Privacy* term is overloaded, and can, in general, assume a wide range of different meanings. For example, in the context of the HIPAA[1] Privacy Rule, *Privacy* means the individual's ability to control who has the access to personal health care information. From the organizations point of view, *Privacy* involves the definition of policies stating which information is collected, how it is used, and how customers are informed and involved in this process. Moreover, there are many other definitions of privacy that are generally related with the particular environment in which the privacy has to be guaranteed. What we need is a more generic definition, that can be instantiated to different environments and situations. From a philosophical point of view, Schoeman (1984) and Walters (2001) identify three possible definitions of privacy:

1. Privacy as the right of a person to determine which personal information about himself/herself may be communicated to others.
2. Privacy as the control over access to information about oneself.
3. Privacy as limited access to a person and to all the features related to the person.

These three definitions are very similar apart from some philosophical differences that are not in the scope of our work. What is interesting from our point of view is the concept of "Controlled Information Release" emerging from the previous definitions. From this idea, we argue that a definition of privacy that is more related with our target could be the following: "*The right of an individual to be secure from unauthorized disclosure of information about oneself that is contained in an electronic repository*". Performing a final tuning of the definition, we consider privacy as "*The right of an entity to be secure from unauthorized disclosure of sensible information that are contained in an electronic repository or that can be derived as aggregate and complex information from data stored in an electronic repository*". The last generalization is due to the fact that the concept of individual privacy does not even exist. As in Oliveira and Zaiane (2004) we consider two main scenarios. The first is the case of a Medical Database where there is the need to provide information about diseases while preserving the patient identity. Another scenario is the classical "Market Basket" database, where the transactions related to different client purchases are stored and from which it is possible to extract some information in form of association rules like "If a client buys a product X, he/she will purchase also Z with y% probability". The first is an example where individual privacy has to be ensured by protecting from unauthorized disclosure sensitive information in form of specific data items related to specific individuals. The second one, instead, emphasizes how not only the raw data contained into a database must be protected, but also, in some cases, the high level information that can be derived from non sensible raw data need to protected. Such a scenario justifies the final generalization of our privacy definition. In the light of these considerations, it is, now, easy to define which are the main goals a PPDM algorithm should enforce:

1. A PPDM algorithm should have to prevent the discovery of sensible information.
2. It should be resistant to the various data mining techniques.
3. It should not compromise the access and the use of non sensitive data.
4. It should be usable on large amounts of data.
5. It should not have an exponential computational complexity.

Current PPDM algorithms do not satisfy all these goals at the same time; for instance, only few of them satisfy the point (2). The above list of goals helps us to understand how to evaluate these algorithms in a general way. The framework we have identified is based on the following evaluation dimensions:

- *efficiency*, that is, the ability of a privacy preserving algorithm to execute with good performance in terms of all the resources implied by the algorithm;
- *scalability*, which evaluates the efficiency trend of a PPDM algorithm for increasing sizes of the data from which relevant information is mined while ensuring privacy;
- *data quality* after the application of a privacy preserving technique, considered both as the quality of data themselves and the quality of the data mining results after the hiding strategy is applied;
- *hiding failure*, that is, the portion of sensitive information that is not hidden by the application of a privacy preservation technique;
- *privacy level* offered by a privacy preserving technique, which estimates the degree of uncertainty, according to which sensitive information, that has been hidden, can still be predicted.

An important question is which one among the presented "dimensions" is the most relevant for a given privacy preserving technique. Dwork and Nissim (2004) make some interesting observations about this question. In particular, according to them in the case of statistical databases *privacy* is paramount, whereas in the case of distributed databases for which the privacy is ensured by using a secure multiparty computation technique *functionality* is of primary importance. Since a real database usually contains a large number of records, the performance guaranteed by a PPDM algorithm, in terms of time and communication requirements, is a not negligible factor, as well as its trend when increasing database size. The *quality of data* guaranteed by a PPDM algorithm is, on the other hand, very important when ensuring privacy protection without damaging the data usability from the authorized users. A trade-off metric can help us to state a unique value measuring the effectiveness of a PPDM algorithm. In Domingo-Ferrer and Torra (2002) the score of a masking method provides a measure of the trade-off between disclosure risk and information loss. It is defined as an average between the ranks of disclosure risk and information loss measures, giving the same importance to both metrics. In Duncan, Keller-McNulty and Stokes (2001) a *R*-*U* confidentiality map is described that traces the impact on disclosure risk *R* and data utility *U* of changes in the parameters of a disclosure limitation method which adopts an additive noise technique. We believe that an index assigning the same importance to both the data quality and the degree of privacy ensured by a PPDM algorithm is quite restrictive, because in some contexts one of these parameters can be more relevant than the other. Moreover, in our opinion the other parameters, even less relevant ones, should be also taken into account. The efficiency and scalability measures, for instance, could be discriminating factors in choosing among a set of PPDM algorithms that ensure similar degrees of privacy and data utility. A weighted mean could be, thus, a good measure for evaluating by means of a unique value the quality of a PPDM algorithm. In the current work, however, we mainly focus on the different evaluation criteria characterizing a PPDM algorithm. In the following, we discuss in details each evaluation criteria we have identified.

3.1. Efficiency

The assessment of the resources used by a privacy preserving data mining algorithm is given by its *efficiency*, which represents the ability of the algorithm to execute with good performance in terms of all used resources. Performance is assessed, as usually, in terms of time and space, and, in case of distributed algorithms, in terms of the communication costs incurred during information exchange.

Time requirements can be evaluated in terms of CPU time, or computational cost, or even the average of the number of operations required by the PPDM technique. Clearly, it would be desirable that the algorithms have a polynomial complexity rather than an exponential one. Anyway, it can be useful to compare the performance of the privacy preserving method with the performance of the data mining algorithm for which the privacy preserving method has been developed. Our expectation is that the execution times of the hiding strategies be proportional to the execution times of the mining algorithms that extract the sensitive information.

Space requirements are assessed according to the amount of memory that must be allocated in order to implement the given algorithm.

Finally, communication requirements are evaluated for those data mining algorithms, which require information exchanges during the secure mining process, as the cryptography-based techniques. It is measured in terms of the number of communications among all the sites involved in the distributed data mining task.

3.2. Scalability

Scalability is another aspect that it is important to assess when a PPDM algorithm is analyzed: it describes the efficiency trends for increasing values in data sizes. Therefore, such parameter concerns the increase of both performance and storage requirements together with the costs of the communications required by a distributed technique when data sizes increase. Because of the continuous advances in hardware technology, it is today easy to store large amounts of data. Thus, databases along with data warehouses today store and manage amounts of data which are increasingly large. For this reason, a PPDM algorithm has to be designed and implemented for being scalable with larger and larger datasets. The less rapid is the decrease in the efficiency of a PPDM algorithm for increasing data dimensions, the better is its scalability. Therefore, we have to evaluate the scalability of a PPDM technique as an equally important requirement of such a kind of algorithms.

3.3. Data quality

The main feature of the most PPDM algorithms is that they usually apply perturbation techniques in order to sanitize data from sensitive information (both private data items and complex data correlations). The data quality is, thus, an important parameter to take into account in the evaluation of a PPDM technique. Since the data is often sold for making profit, or shared with others in the hope of leading to innovation, data quality should have an acceptable level according also to the intended data usage. If data quality is too degraded, the released database is useless for the purpose of knowledge extraction. As discussed in the previous section, several data quality metrics have been proposed that are either generic or data-use-specific. However, currently, there is no metric that is widely accepted by the research community. Here we try to identify a set of possible measures that can be used to evaluate different aspects of data quality. We believe that, in evaluating the data quality after the privacy preserving process, it can be useful to assess both the *quality of the data* themselves, considered as a general measure evaluating the state of the individual items contained in the database after the enforcement of a privacy preserving technique, and the *quality of the data mining results* for evaluating the alteration in the information that is extracted from the database after the privacy preservation process, on the basis on the intended data use.

The main problem with data quality is that its evaluation is relative (Tayi and Ballou, 1998), in that it usually depends from the context with respect to which it is analyzed. More in general in the scientific literature data quality is considered a multi-dimensional concept that in certain contexts involves both objective and subjective parameters (Ballou and Pazer, 1985; Wang and Strong, 1996). However, among the various possible parameters, the following ones are usually considered the most relevant:

- Accuracy: it measures the proximity of a sanitized value a' to the original value a.
- Completeness: it evaluates the degree of missed data in the sanitized database.
- Consistency: it is related to the internal constraints, that is, the relationships that must hold among different fields of a data item or among data items in a database.

While the accuracy is a relatively general parameter in that it can be measured without strong assumptions on the dataset analyzed, the completeness is not so general, because, for example, in some PPDM strategies, e.g. blocking, its evaluation is not significant. On the other hand, the last parameter requires to determine all the relationships that are relevant for a given dataset. Data consistency and completenesses are not explored here; we plan to address the development of evaluation methodologies for these parameters as future work.

The accuracy as measure of the quality of data to which a PPDM algorithm is applied is closely related to the *information loss* resulting from the hiding strategy: the lower is the information loss, the greater is the data quality. In our opinion, its measure depends on the specific class of PPDM algorithms. As for heuristic-based techniques, we distinguish the following cases based on the modification technique that is performed for the hiding process. If the algorithm adopts a perturbation or a blocking technique to hide both raw and aggregated data, the information loss can be measured in terms of the dissimilarity between the original dataset D and the sanitized one D'. In the case of transactional dataset perturbation, we can compute such dissimilarity as the difference between the item frequencies of the two datasets before and after the sanitization, as expressed by the following formula:

$$Diss(D, D') = \frac{\sum_{i=1}^{n} |f_D(i) - f_{D'}(i)|}{\sum_{i=1}^{n} f_D(i)} \tag{1}$$

where i is a data item in the original database D and $f_D(i)$ is its frequency within the database, whereas i' is the given data item after the application of a privacy preservation technique and $f_{D'}(i)$ is its new frequency within the transformed database D'. As we can see, the information loss is defined as the ratio between the sum of the absolute errors made in computing the frequencies of the items from a sanitized database and the sum of all the frequencies of items in the original database. If the PPDM algorithm uses, instead, a blocking technique for inserting into the dataset uncertainty about some sensitive data items or their correlations, the dissimilarity can be measured again by the formula 1, where the frequency of the item i belonging to the sanitized dataset D' is given by the mean value between the minimum frequency of the data item i, computed by considering all the blocking values '?' associated with it equal to zero, and the maximum frequency, obtained by considering all the question marks equal to one. In case of data swapping, the information loss caused by an heuristic-based algorithm can be evaluated by a parameter measuring the *data confusion* introduced by the value swappings. If there is no correlation among the different database records, the *data confusion* can be estimated by the percentage of value replacements executed in order to hide specific information. If the data modification consists of aggregating some data values, the information loss is given by the loss of detail in the data. Intuitively, in this case, in order to perform the hiding operation, the PPDM algorithms use some type of "Generalization or Aggregation Scheme" that can be ideally modelled as a tree scheme. Each cell modification applied during the sanitization phase using the Generalization tree introduces a data perturbation that reduces the general accuracy of the database. As in the case of the K Anonymity algorithm presented in Sweeney (2002), we can use the following formula. Given a database DB with N_A fields and N transactions, if we identify as generalization scheme a domain generalization hierarchy *GT* with a depth *h*, it is possible to measure the quality of a sanitized database *SDB* as:

$$Quality(SDB) = 1 - \frac{\sum_{i=1}^{i=N_A} \sum_{j=1}^{i=N} \frac{h}{|GT_{Ai}|}}{|DB| * |N_A|} \tag{2}$$

where $\frac{h}{|GT_{Ai}|}$ represent the detail loss for each cell sanitized. For hiding techniques based on sampling approach, the quality is obviously related to the size of the considered sample and, more generally, on its features. A further analysis is needed to understand how the data quality of this kind of hiding techniques can be better assessed.

In the context of reconstruction-based algorithms aiming to hide the values of a confidential attribute by a statistical-based perturbation technique, the information loss is basically the lack of precision in estimating the original distribution function of the given attribute. As in Agrawal and Aggarwal (2001), we measure the information loss incurred by a reconstruction-based algorithm in estimating the density function $f_X(x)$ of the attribute X, by computing the following value:

$$I(f_X, \widehat{f_X}) = \frac{1}{2} E\left[\int_{\Omega_X} \left|f_X(x) - \widehat{f_X}(x)\right| dx\right] \tag{3}$$

that is, half of the expected value of L_1 norm between $f_X(x)$ and $\widehat{f_X}(x)$, which are the density distributions respectively before and after the application of the privacy preserving technique.

By contrast, we observe that cryptography-based techniques adopting Secure Multiparty Computation protocols, typically used in distributed environments, do not make use of any kind of perturbation technique in order to preserve privacy. Such techniques assure data privacy at each site by limiting the information shared by all the sites through the use of cryptographic techniques. Thus, the quality of data stored et each site is not compromised at all.

As we have stated above, in some cases it can be useful and also more relevant to evaluate the quality of the data mining results after the sanitization process. This kind of metric is strictly related to the use the data are intended for. Data can be analyzed in order to mine information in terms of associations among single data items or to classify existing data with the goal of finding an accurate classification of new data items, and so on. Based on the intended data use, the information loss is measured with a specific metric, depending each time on the particular type of knowledge model one aims to extract.

If the intended data usage is data clustering, the information loss can be measured by the percentage of legitimate data points that are not well-classified after the sanitization process. Data modification often applied by a privacy preserving technique obviously affects the parameters involved in the clustering analysis. There is, thus, the need to control, as much as possible, the results of such analysis before and after the application of a data hiding technique.

When quantifying information loss in the context of the other data usages, it is useful to distinguish between: *lost information* representing the percentage of non-sensitive patterns (i.e., association, classification rules) which are hidden as side-effect of the hiding process; and the *artifactual information* representing the percentage of artifactual patterns created by the adopted privacy preserving technique.

In case of association rules, the lost information can be modelled as the set of non-sensitive rules that are accidentally hidden, referred to as `lost rules`, by the privacy preservation technique, the artifactual information, instead, represents the set of new rules, also known as `ghost rules`, that can be extracted from the database after the application of a sanitization technique.

Similarly, if the aim of the mining task is data classification, e.g. by means of decision trees inductions, both the lost and artifactual information can be quantified by means of the corresponding lost and ghost association rules derived by the classification tree.

These measures allow one to evaluate the high level information that are extracted from a database in form of the widely-used inference rules before and after the application of a PPDM algorithm.

3.4. Hiding failure

The percentage of sensitive information that is still discovered, after the data has been sanitized, gives an estimate of the *hiding failure* parameter. Most of the developed privacy preserving algorithms are designed with the goal of obtaining zero hiding failure. Thus, they hide all the patterns considered sensitive. However, it is well known that the more sensitive information we hide, the more non-sensitive information we miss. Thus, some privacy preserving data mining algorithms have been recently developed which allow one to choose the amount of sensitive data that should be hidden in order to find a balance between privacy and knowledge discovery.

3.5. Privacy level

In order to evaluate the privacy protection offered by a PPDM algorithm, we need to define a unique parameter quantifying the privacy level ensured by these algorithms. As previously stated, a metric for evaluating the privacy level offered by a PPDM method is proposed in Agrawal and Srikant (2000): if the perturbed value of an attribute can be estimated, with a confidence c, to belong to an interval $[a, b]$, then the privacy is estimated by $(b\text{-}a)$ with confidence c. This metric does not work well because it does not take into account the distribution of the original data along with the perturbed data. We need, therefore, a metric that considers all the informative content of data available to the user. Agrawal and Aggarwal (2001) address this problem by introducing a new privacy metric based on the concept of information entropy.

Shannon in formulating his most well-known theorem (Shannon, 1948) defines the concept of *Information Entropy* as follows: let X be a random variable which takes on a finite set of values according to a probability distribution $p(x)$. Then, the entropy of this probability distribution is defined as follows:

$$h(X) = -\sum p(x)\log_2(p(x)) \tag{4}$$

or, in the continuous case:

$$h(X) = -\int f(x)\log_2(f(x))\,dx \tag{5}$$

where *f(x)* denotes the density function of the continuous random variable *x*.

Information Entropy is a measure of how much "choice" is involved in the selection of an event or how uncertain we are of its outcome. It can be used for quantifying the amount of information associated with a set of data. The concept of "information associated with data" can be useful in the evaluation of the privacy achieved by a PPDM algorithm. Because the entropy represents the information content of a datum, the entropy after data sanitization should be higher than the entropy before the sanitization. Moreover the entropy can be assumed as the evaluation of the uncertain forecast level of an event which in our context is evaluation of the right value of a datum. As in Agrawal and Aggarwal (2001), we measure the level of privacy inherent in an attribute *X*, given some information modeled by *Y*, by the following quantity:

$$\Pi(X \mid Y) = 2^{-\int f_{X,Y}(x,y)\log_2 f_{X|Y=y}(x))dxdy} \tag{6}$$

However, we have to notice that the value of the privacy level depends not only on the PPDM algorithm used, but also on the knowledge that an attacker has about the data before the use of data mining techniques and the relevance of this knowledge in the data reconstruction operation. This problem is underlined, for example, in Trottini (2001, 2003). In our work this aspect is not considered in the testing phase, but it will be investigated as part of future work; for instance by referring to expression (6), it is possible to introduce assumptions on attacker knowledge by properly modeling *Y*.

The measure of the entropy level, and thus of the privacy level, is very general and in order to use it in the different PPDM contexts, it needs to be refined in relation with some characteristics like the type of transactions, the type of aggregation and PPDM methods. Here, for example, we show how the entropy concept can be instantiated in order to evaluate the privacy level in the context of "association rules". Our approach is based on the work of Smyth and Goodman (1992) that use the concept of *Information Entropy* in order to measure the amount of information contained in the association rules extracted from a database, with the aim of ranking and thus characterizing the most important rules in terms of information they contain. They think of a rule $y \Rightarrow x$ as a condition *if* $Y = y$ *then* $X = x$ with a certain probability p and they then define a *J*-measure representing the entropy of a rule as:

$$J(x, Y = y) = p(y)j(x, Y = y) \tag{7}$$

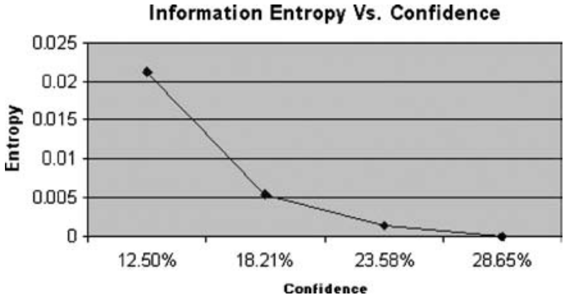


Figure 2. Evolution of information entropy with respect to confidence.

The term *p(y)* is the probability of the rule antecedent and *j(x, Y = y)* is the cross-entropy, defined as:

$$j(x, Y = y) = p(x \mid y) \log \frac{p(x \mid y)}{p(x)} + (1 - p(x \mid y)) \log \frac{1 - p(x \mid y)}{1 - p(x)}. \tag{8}$$

If we consider the association rules model and a specific rule $y \Rightarrow x$, the value *p(y)*, that is, the probability of antecedent, is equal to frequency of *y* and the value *p(x / y)* is the probability that the variable X assumes the value *x*, given that *y* is the value of variable Y. It represents the strength of the rule *if Y = y then X = x* and it is referred to as *confidence* of the rule. Now, we define a parameter entropy privacy (*EP*) as:

$$EP = J(x, Y = y) - J1(x, Y = y) \tag{9}$$

where *J1* is the *J*-measure after the operation of data hiding. Some preliminary tests executed in the context of this work, show that the simple *J*-measure does not provide an intuitive evaluation parameter. In fact, as the Information Theory suggests, we would expect as result that when the confidence decreases, the level of entropy increases (see figure 2). Indeed, in some particular cases, the trend obtained is the one shown in figure 3. This is due to the fact that the *J*-measure represents the average conditional mutual information, or, in other words, the difference between the "a priori" and "a posteriori" probabilistic distributions of the two random variables *X* and *Y*. On the base of this observation, we note that if:

- *P(X ∧ Y) < P(X)× P(Y)* the two variables *X* and *Y* are negatively correlated
- *P(X ∧ Y) > P(X)× P(Y)* the two variables *X* and *Y* are positively correlated
- *P(X ∧ Y) = P(X)× P(Y)* the two variables *X* and *Y* are independent

Remembering that:

$$\frac{P(X \wedge Y)}{P(Y)} = P(X \mid Y) \tag{10}$$

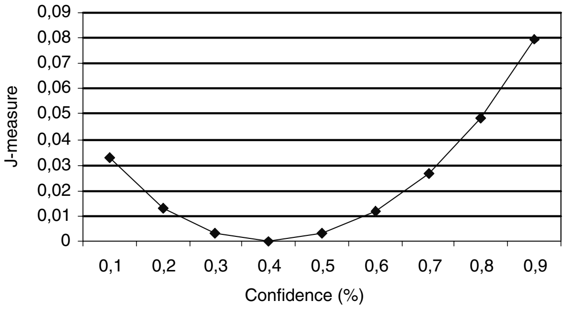


Figure 3. Evolution of information entropy with respect to confidence in some particular cases.

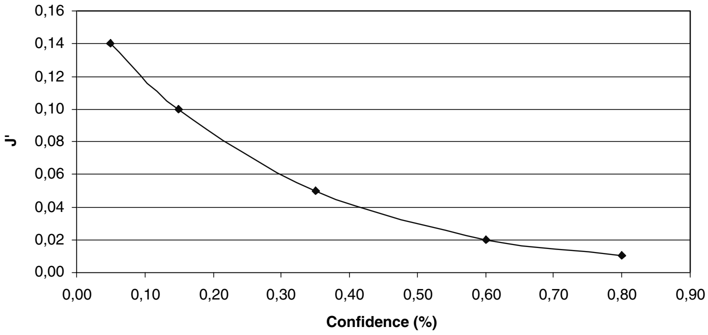


Figure 4. Evolution of J' with respect to Confidence.

we observe than the J-measure does not take into account the type of correlation between the involved random variables. For this reason, we finally adopt as measure the derivative of the J-measure:

$$J'(X; Y = y) = p(y) * \left(\log_2\left(\frac{p(x \mid y)}{p(x)}\right) - \log_2\left(\frac{1 - p(x \mid y)}{1 - p(x)}\right)\right) \tag{11}$$

Figure 4 shows the graph obtained when using J'. Finally, we measure the amount of privacy introduced by the following expression:

$$Level\_of\_privacy = (J'_1 - J') \tag{12}$$

where J'_1 is calculated after the sanitization and J' is measured before sanitization.

By using the expression (12), we can compute the level of privacy of a classification model, where the class is represented in form of a classification rule *if A1 A2...An then Ci*.

We observe that a decrease in the confidence value, and therefore an increase in the level of privacy, results in an increase of entropy, as shown in figure 4. This trend is in accordance with the Information Theory, because if a rule or in general an information is well hidden, the informative value of its discovery is higher than an information that is simple to discover. We plan to define a corresponding privacy level, specifically conceived for PPDM algorithms based on clustering techniques.

4. Experimental evaluations

In order to measure the effective accuracy of the parameters described in the previous section, we have carried out some experimental evaluations on a set of data hiding algorithms. In the following, we first present the algorithms we have assessed according to these evaluation criteria. We then describe the data set we have used along with the methodology we have adopted for the evaluation process. Finally, the results of our evaluations are discussed.

4.1. Algorithms

Here, we describe a specific class of privacy preserving data mining algorithms, developed in the context of CODMINE project (CODMINE, 2002–2003), that we have chosen as testbed for our evaluations. As part of future work, we plan to apply our framework to the other classes of PPDM algorithms in order to assess them according to the several criteria we have identified. The algorithms, we have analyzed as testbed for our evaluation activity, rely on a heuristic-based hiding approach, according to which the original dataset is modified by blocking data values. Their goal is not to protect the individual privacy but the extraction of aggregated data in terms of association rules, that are considered sensitive. First of all, we present a scenario that requires preventing the disclosure of sensitive information, modelled as associations among data items, which could be even unknown to data owners.

4.1.1. Scenario. A large number of examples from different application domains (such as financial, medical, scientific, demographic, military environments) can be identified for which data mining represents a threat to information security. Consider the well-known market basket analysis environment. Suppose that a large supermarket chain gives a supplier of it, e.g. a HIGH-TV Company making and selling televisions, access to its database containing customer purchases in exchange for a reduction in their television prices. Then, the company can use an association rule mining tool and find some interesting correlations between the purchase of televisions and other products. For example, the company could find that people who purchase DVD player also purchase a television made by the STAR Company, its competing firm. Therefore, the HIGH-TV company can exploit the sensitive information to run a promotional campaign based on which who purchases a HIGH-TV television receives a discount coupon for buying a DVD player. This causes a heavy drop in selling STAR televisions, which increases the store prices to the market chain, because of the lower sales. Moreover, during a subsequent negotiation between the HIGH-TV Company and the supermarket chain, given the reduced competition, the company will be unwilling to reduce the store prices of their televisions to the supermarket chain. The association rule mining tools can, thus, represent a threat to the privacy of individuals and, in this case, to the confidentiality of information concerning abstract entities such as companies, organizations, and so on. Such tools allow an authorized miner to learn about unknown data correlations that can be exploited against the released entity, being prejudicial to its own interests or, in some cases, to the common interests. In the considered example, the supermarket chain wants to take advantage of sharing the huge amount of data it possesses, but, at the same time, it does not want to unconsciously release reserved information. Therefore, it needs a data mining tool providing a certain level of privacy, considered as privacy of its own businesses, when the rule mining process is executed by an authorized party. On the other side, who purchases the access to a data repository wants the data to be as much as possible close the reality they refer to. There is, thus, the need of an association mining tool for mainly ensuring, on one hand, privacy to data owners and, on the other hand, data accuracy to the legal miners.

4.1.2. Problem formulation. Now, we give a formal description of the rule hiding problem. Let D be a transactional database and $I = \{i_1, \ldots, i_n\}$ be a set of literals, called items. Each transaction can be considered as an itemset that is included in I. We assume that the items in a transaction or in an itemset are sorted according to a 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, and *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. A detailed description of this notation can be found in Sabanci University (2003). An association rule is an implication of the form $X \Rightarrow Y$ between two disjoint itemsets X and Y in I. Each rule is assigned both a support and a confidence value. The first one is a measure of a rule frequency, more precisely, it is the probability to find in the database transactions containing all the items in $X \cup Y$, whereas the confidence is a measure of the strength of the relation between the antecedent X and the consequent Y of the rule, that is, the probability to find transactions containing all the items in $X \cup Y$, once we know that they contain X. An association rule mining process consists of two steps: (1) the identification of all the frequent itemsets, that is, all the itemsets, whose supports are higher 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 higher than a minimum confidence threshold, *min_conf*. Along with confidence and 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 is 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 those sensitive rules included in 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.

```
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 5. Algorithm GIH for Rule Fuzzification by Hiding their generating Greatest Itemsets.

4.1.3. Rule hiding algorithms based on data fuzzification. According to the data fuzzification approach, a sequence of symbol 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, it is 0 otherwise. The novelty of this approach is the insertion of an uncertainty symbol, e.g. 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 uniquely determined, but they can range between a minimum and a maximum value. The minimum support of an itemset is defined as the percentage 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*.

```
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 $l_r$ in $t$
    3. sort the transactions in $T'_{l_r}$ in descending order of
       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 rules
      9. Remove $t$ from $T'_{l_r}$
    }
    10. Remove $r$ from $R_h$
  }
End
```

Figure 6. Algorithms CR and CR2 for Rule Fuzzification by Confidence Reduction.

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

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 at introducing 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. In order to reduce the support of the large itemset generating a sensitive rule, Algorithm GIH 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 CR and CR2 operate by 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 with 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. All the fuzzification algorithms hide a sensitive rule with an uncertainty level by decreasing the minimum support or confidence values below the resulting thresholds, MST-SM and MCT-SM. More details can be found in Sabanci University (2003).

4.2. *Testing framework*

In this section we present the assessment of the data fuzzification algorithms described in the previous section according to the evaluation framework we have proposed. We describe in details the methodology adopted in the evaluation process, the dataset with respect to which the algorithms have been evaluated, and the tools supporting our evaluation methodology.

4.2.1. Evaluation methodology. The methodology consists of three main steps:

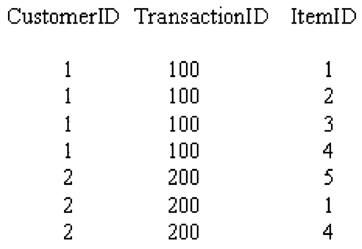
- Database Creation.
- Sanitization of the generated databases.
- Testing of each algorithm according to the specified evaluation criteria.

Each algorithm tries to hide sensitive association rules that can be extracted by the Apriori algorithm from an artificially generated database that summarizes the properties of a supermarket database recording the customer purchase habits. Each row of such database specifies the customer ID, the transaction ID and a set of items corresponding to those products bought by that customer with this ID during the given transaction. While trying to hide some association rules, each algorithm follows a different strategy which leads to questions like: "Which algorithm has the best performance, and the worst data quality? Which algorithm provides the highest privacy level?" In order to answer these questions, we performed the following steps. We first created a synthetic databases by using the IBM Synthetic Data Generator, IBM SDG, (see `http://www.almaden.ibm.com/software/quest/Resources/datasets/syndata.html`). We then analyzed the database to detect all relevant association rules. After that, the sensitive rules to be hidden and the hiding algorithm to use were chosen. Based on these choices, the sanitized database was then generated. All these activities were supported by a prototype system whose architecture will be explained in subsection 4.2.3.

4.2.2. The dataset. In evaluating the data fuzzification algorithms, we have selected a homogeneous data set for all the proposed algorithms with the aim of ensuring a uniform evaluation. Now, we briefly describe the characteristics of the dataset we have selected.

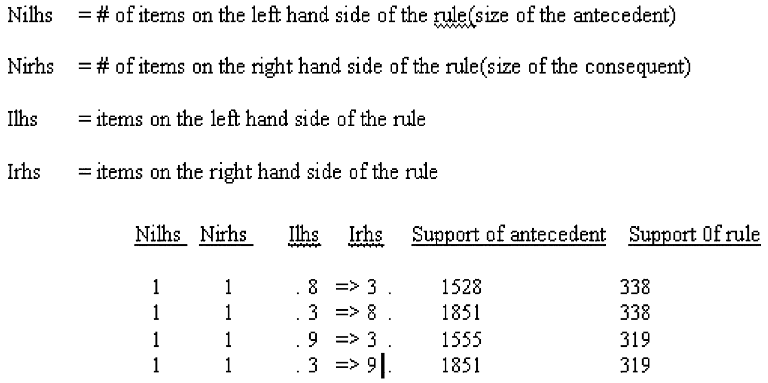
The dimension of the itemset is 50 items, and the average length of the transaction is 5. A different dataset size is selected based on the particular test we have to execute. For tests on data quality, efficiency and hiding failure, we use a database with dimension 20 K; for scalability tests, we used a greater number of databases with dimension of 20 K, 40 K, 60 K, 80 K, 100 K. For the tests on the level of privacy, instead, we have chosen a database of 10K, because this parameter, which depends on the antecedent and consequent supports and on the rule confidence, is not affected by the database dimension.

4.2.3. The prototype. The prototype, that supports the evaluation methodology we have chosen, consists of the following five modules: (i) the file optimizer, (ii) the Apriori Algorithm (association rule generator); (iii) the rule hiding module; (iv) the comparison module; (v) the user interface. The file optimizer takes an input file (in the ".txt" format) containing the transactions generated by the IBM transaction generator. An example of a *.txt file is reported in figure 7. According to such file, the customer whose ID is equal to 1 bought the items identified by 1, 2, 3, and 4 as part of the transaction with ID equal to 100, whereas the customer with ID = 2 bought the items 5, 1 and 4 has part of the transaction with ID equal to 200. The task of the file optimizer module consists of optimizing the database



CustomerID	TransactionID	ItemID
1	100	1
1	100	2
1	100	3
1	100	4
2	200	5
2	200	1
2	200	4

Figure 7. An example of a database transaction file(*.txt).



Nilhs = # of items on the left hand side of the rule(size of the antecedent)

Nirhs = # of items on the right hand side of the rule(size of the consequent)

Ilhs = items on the left hand side of the rule

Irhs = items on the right hand side of the rule

Nilhs	Nirhs	Ilhs	Irhs	Support of antecedent	Support Of rule
1	1	. 8	=> 3 .	1528	338
1	1	. 3	=> 8 .	1851	338
1	1	. 9	=> 3 .	1555	319
1	1	. 3	=> 9 .	1851	319

Figure 8. An example of a rule file(*.out).

transaction file by converting the input file from the text (".txt") to a binary format (file extension suffix ".opt"). The use of binary data reduces the database size if the database contains a large number of transactions. For instance, a *.txt database file of size 1.67 MB can be reduced to the size of 144 KB after the optimization process.

Then, the output of the file optimizer module is given to the Apriori Algorithm which uses the optimized transaction file to generate association rules with the given minimum support and confidence values. The association rule generator returns a file with ".out" extension which contains the association rules of the optimized transactional database. An example of a *.out file is given in figure 8.

Each row in the file illustrated in Figure 8 provides the size of both the antecedent and consequent of the considered rule along with the antecedent and rule supports. Then the rule hiding module is invoked: it takes the output of the association rule generator module, that is, a file of rules with extension ".out" and a ".txt" file containing a parameter denoting the set of rules to be hidden along with its cardinality. The output of the rule hiding module is a ".txt" file containing the transactions generated. If we process the output of the rule hiding module through the file optimizer and the association rule generator in this order, we obtain the association rules resulting from the rule hiding process. By comparing the rules before and after the rule hiding process through the comparison module, we can determine the differences between the two rule files, and thus we are able to determine how many

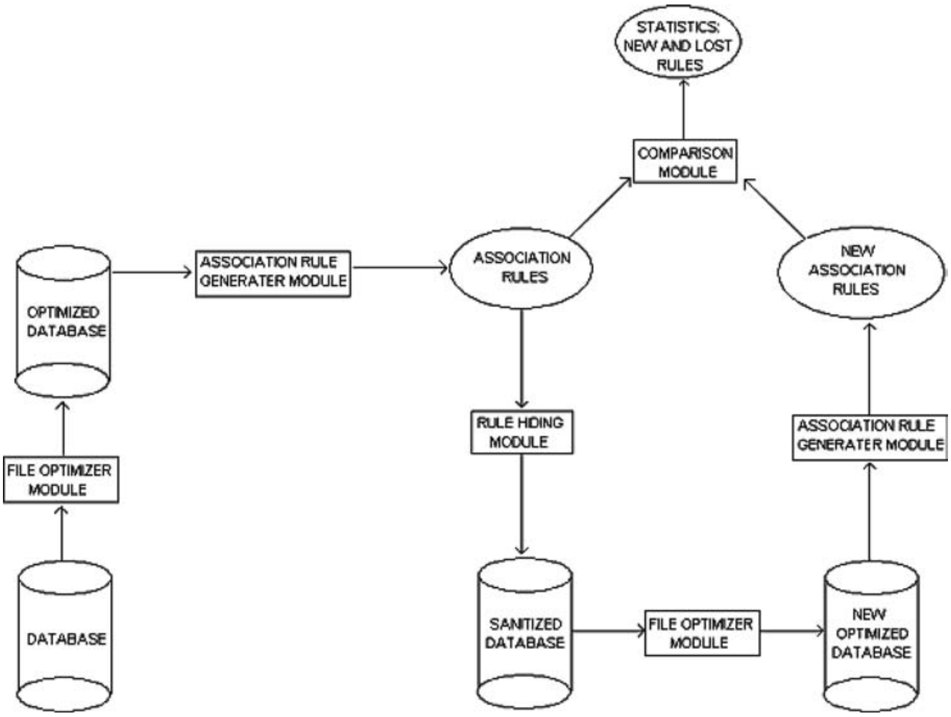


Figure 9. Architecture of the prototype supporting the evaluation methodology.

rules are "lost", and how many rules are "new" (that is, how many new rules have been introduced).

The prototype has been developed in Delphi environment by using Object Pascal language. Figure 9 illustrates the prototype architecture. It takes the input parameters from the user and sends them to the executables that are compiled in MS Visual Studio C++. These executable modules are the file optimizer, the apriori, the rule hider and the comparison module that have been previously described. Each executable module performs synchronously.

4.2.4. General automatic data hiding algorithms evaluation tool (GADHAET). The prototype we have described is the core of a more general tool, called *GADHAET*, that we have designed with the aim to realize a general and automatic evaluator of data hiding algorithms. From the logical point of view, this tool is composed by four main units: data mining, data hiding, statistics, loader.

The *Data mining* unit receives in input a database and applies the appropriate data mining algorithms in order to determine all possible information that can be retrieved from the database. The results of this phase are transmitted to the *Data Hiding* and *Statistics* unit. The first one applies the algorithms that we want to test to the database, generating an *Internal Sanitized Database* over which the data mining algorithms will be re-applied. The results of these operation are collected and analyzed by the *Statistic* unit in order to assess the different properties of the considered algorithm according to the evaluation framework. In order to make the tool usable for a large variety of data mining techniques, the different algorithms and the different strategies used for evaluating the criteria over different types of algorithm and with different types of database are not part of the tool, but they are included in *dynamic modules* that can be automatically loaded by using the *Loader* unit. It is thus possible to test new algorithms and strategies by simply encoding them into new modules that can be transparently used.

4.3. Evaluation of experimental results

Here we present some results of the experiments we have performed in order to evaluate the set of hiding algorithms based on data fuzzification according to the evaluation criteria described in Section 3. The experiments have been performed on a Pentium III with 256 MB of RAM under Windows OS. In the testing phase, the same working load of the system was ensured. For each set of experiments, five trials have been executed and the average value has been computed.

4.3.1. Efficiency evaluation. In analyzing the efficiency of the considered PPDM algorithms, we focused our attention on assessing the time requirements given the irrelevant costs for data storage. The time requirements of the algorithms presented in Section 4.1 have been evaluated in terms of both computational costs and CPU times. We first present the results of the computational cost analysis of each algorithm. Then, we describe the CPU times we obtained from the performed experiments.

In what follows, we denote with *ATL* the average number of items per transaction, while $A_D = |D| * ATL$ denotes the average number of items in the database. The following theorems state the theoretical complexity of the three fuzzification algorithms. We refer the reader to Appendix A for the proofs.

Theorem 4.1. *Algorithm GIH performs in* $O(|L_h| * |D|)$.

Theorem 4.2. *Algorithm CR performs in* $O(|R_h| * A_D)$.

Theorem 4.3. *Algorithm CR2 performs in* $O(|R_h| * A_D)$.

Besides the theoretical analysis on the complexity of the considered algorithms, we have evaluated the efficiency of these algorithms by means of experimental measurements of their execution times. They are measured in terms of CPU time (in milliseconds) by increasing, first, the values of the rule support or confidence, and then, the number of sensitive rules to be hidden. We applied the data fuzzification algorithms to a database containing 20 K transactions, 50 different items (A_D) and characterized by an average number of items per transaction (*ATL*) equal to 10. For analyzing the CPU time for increasing values of the rule support or confidence, we chose four different association rules with increasing support and confidence values. Then the support and confidence thresholds (MST and MCT) have

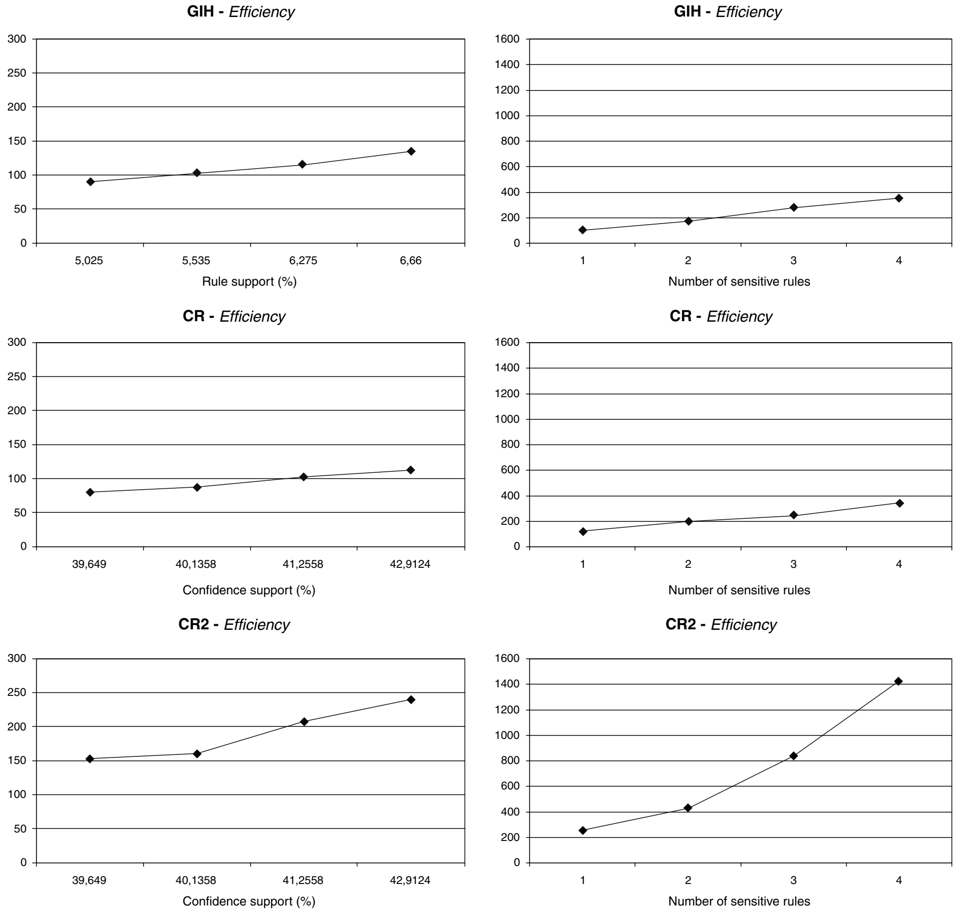


Figure 10. CPU times of Algorithms GIH, CR respectively, when varying support/confidence values and number of sensitive rules to be hidden.

been fixed equal to the minimum value of support and confidence; the safety margin (SM), instead, has been set equal to 10% of the fixed thresholds. As for Algorithm GIH, the four rules have been sorted in ascending order of their support, whereas for testing the other algorithms, the rules have been sorted in ascending order of their confidence.

For all three algorithms, as we can see from figure 10, the CPU time increases for increasing values of the support in the case of Algorithm GIH, and of the confidence for the other algorithms and, similarly, when varying the number of rules to be hidden. However, we have to notice that the order of magnitude is different. Algorithm CR2 has the worst performance with respect to both the rule confidence and the number of rules to be hidden. Algorithms GIH and CR, instead, show a similar trend, but Algorithms CR proves to be the most efficient according to the two considered dimensions.

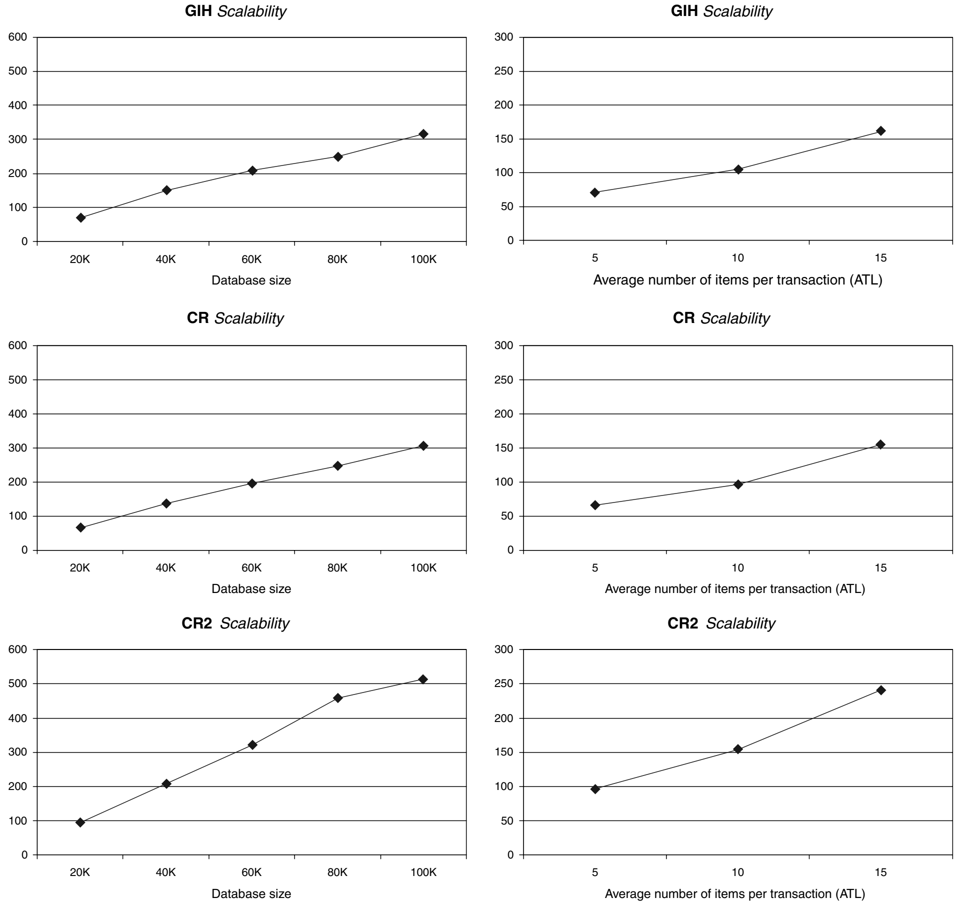


Figure 11. CPU times of Algorithms GIH, CR, CR2, respectively, when increasing data sizes (both ATL and DB size).

4.3.2. Evaluation of scalability. Now, we present the results of the experiments we have executed for evaluating the *scalability* of the three algorithms. This parameter measures the efficiency for increasing values of the data size, that is, the size of the dataset in terms of both transaction number and transaction length (average number of items per transaction, briefly denoted as ATL). Therefore, we have performed two series of tests on each fuzzification algorithm that hides four association rules, characterized by the same features, first, by increasing the database size from 20 k to 100 k, and then by increasing the average transaction length from 5 to 15. The scalability trend of the algorithms is similar to the efficiency trend. As we can see from figure 11, algorithm CR2 proves to be the least scalable algorithm with respect to both the database size and the transaction length. As for

Algorithms GIH and CR, their CPU times grow slower than the CPU time of Algorithm CR2. Again Algorithms CR proves to be the most scalable.

4.3.3. Data quality evaluation. The data quality parameter is a measure of the amount of perturbation applied to the database to be protected from the mining process. In our evaluation, we analyzed both the data dissimilarity measuring the amount of modification applied to the data, and the perturbation in the data mining results after the sanitization process, in terms of the number of *ghost* and *lost rules* due to the hiding operation. Both these metrics, which describe two different, even if correlated, aspects of data quality, have been evaluated, first, when the support or the confidence value of the rule to be hidden increases, and then, when the number of the rules to hide grows. We observe that for each considered parameter, the number of *ghost* rules, due to the application of a fuzzification algorithm, is always equal to zero. Algorithms GIH and CR, in fact, work by decreasing the minimum support and confidence values, respectively, while keeping fixed the maximum value of both support and confidence and, similarly, for all the rules generated by the itemsets containing the blocked values. Algorithm CR2, instead, performs in such a way so to increase the maximum confidence value of the antecedent of the considered rule, while keeping fixed the minimum values for both support and confidence of the rule as well as its antecedent. Therefore, it does not create new rules. Moreover, it less affected by the loss of relevant rules than the other algorithms, even if it has to execute a higher number of operations to reduce the minimum confidence value under the considered threshold, MCT-SM, than Algorithms GIH and CR. This is why Algorithm CR2 decreases the minimum confidence by increasing the maximum support of the rule antecedent, that is, the denominator of the fraction providing the minimum confidence; this also explains the reason why Algorithm CR2 is the less efficient and the less scalable. Figures 12 and 13 show both the two measures for data quality when varying the two considered dimensions. We can see that Algorithm CR has the least dissimilarity since for the same rule to be hidden it performs a lower number of blocking operations by decreasing its minimum confidence. Algorithms CR2, instead, is characterized by the worst dissimilarity (for the reason explained above), even though it ensures the best quality in term of lost rules. The reason is that this algorithm increases the relevance of the rules that are already visible before the privacy preservation process, while limiting the loss of relevant rules. Here, we can see that the combination of these two metrics allows us to better control the state of data after the application of a PPDM algorithm.

To conclude, Algorithm CR is the one that introduces less uncertainty in the database, even if it results in a high loss of relevant association rules. Algorithm CR2, instead, inserts the highest level of uncertainty into the database, but it little affects the results of the rule mining process.

4.3.4. Evaluation of hiding failure. As we have previously discussed, the *hiding failure* parameter provides a measure of the amount of sensitive information that can be still mined after the privacy preservation process.

All algorithms we have analyzed aim at hiding all sensitive rules identified in the transactional database. However, the hiding strategy adopted by Algorithm CR2 applies only if the database to be sanitized contains for each sensitive rule to be hidden some transactions that

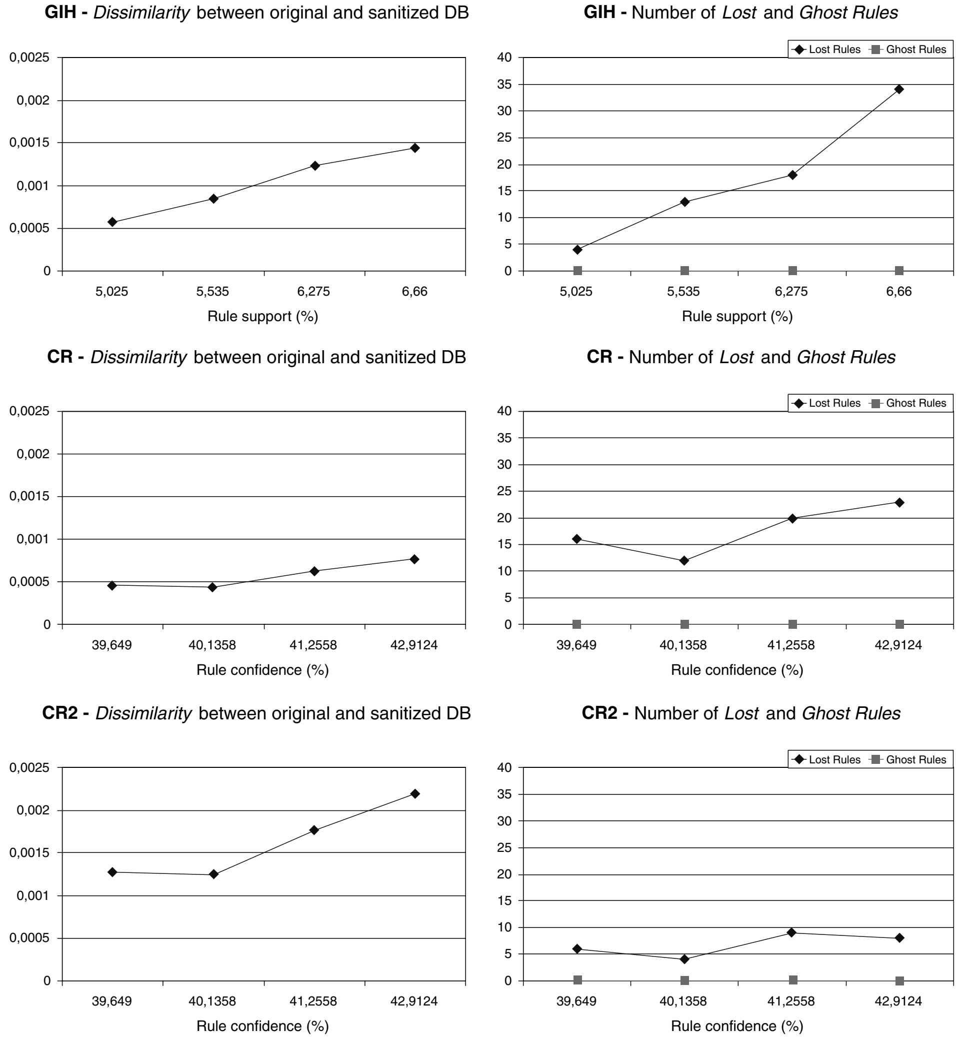


Figure 12. Data quality of Algorithms GIH, CR and CR2 when increasing rule support/confidence values.

partially support the rule antecedent and that do not fully support the rule consequent, that is, they cannot support the rule consequent or can partially support it. If such transactions do not exist, for instance when the rule antecedent consists of only one item, Algorithm CR2 is not able to hide the corresponding rule and, thus, its *hiding failure* parameter is greater than zero. The other data fuzzification algorithms, instead, are characterized by a value of a zero for the hiding failure parameter. Both of them hide a sensitive itemset or an association rule by blocking some items within transactions that fully support the given itemset or the itemset generating the rule. The database obviously contains such transactions given the

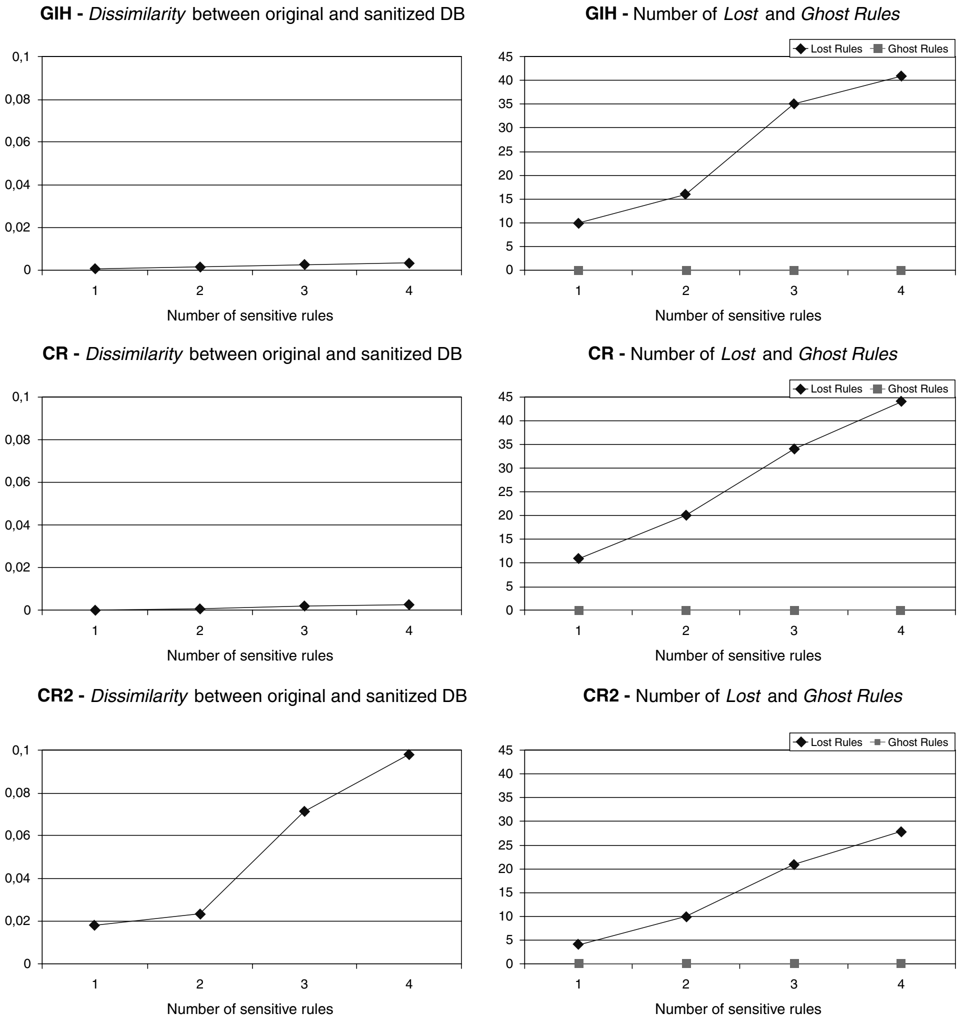


Figure 13. Data quality of Algorithms GIH, CR and CR2 when increasing the number of sensitive rules.

relevance of the considered itemsets. Thus, they provide a complete protection of all the sensitive information contained in any database. Obviously, the more sensitive information we hide, the more non-sensitive information we miss and the more artifactual information we create, in terms of lost and ghost rules, respectively, as we have observed from the data quality graphs. For this reason, in some cases it could be better to keep a low non null hiding failure in order to obtain a sanitized database more similar to the original one.

4.3.5. Evaluation of privacy level. Here, we present some results of the experiments we have executed for assessing the privacy level ensured by the fuzzification algorithms. In

Table 1 Rules to be hidden

Selected rules for Privacy level measure			
Rule_ID	***Rule***	***Supp. (%)***	***Conf. (%)***
R1	9 28 ⇒ 6 13 22	10,23	28,70
R2	9 13 ⇒ 2	25,41	52,39
R3	6 9 22 26 ⇒ 13	10,60	80,71

our evaluation, we have considered a database containing 10 K transactions, 30 items and 10 items per transaction on average. Three rules, R1, R2 and R3, have been considered within the selected database having different support and confidence values. Their features are described in Table 1. The minimum support and confidence thresholds, MST and MCT, have been set equal to the support and confidence values, respectively, of each rule to hide. Then, five different safety margins, SM, have been considered. They are equal to 10%, 20%, 30%, 40%, 50% of the fixed MST and MCT values. Then, each rule has been hidden and the average value of the privacy levels has been computed. We can observe from figure 14 that an increase in the safety margin, resulting in a reduction in the applied threshold, MST-SM or MCT-SM, (for both the support and confidence values) leads to an increase in the level of privacy. Additionally, we notice that Algorithm CR2 is the one ensuring the highest privacy level, whereas the other two algorithms, characterized by a similar trend, hide the fixed rules, R1 and R2, with a lower degree of uncertainty.

To conclude, none of the analyzed algorithm is better the other ones wrt all the parameters we have considered. Then, for choosing which one to use in a real case, all these parameters have to be taken into account, giving each one the relevance required by the specific case and the particular application domain.

5. Conclusions and future work

In this paper, we have proposed a framework for evaluating privacy preserving data mining algorithms. Such framework allows one to assess the different features of a privacy preserving algorithm according to a variety of evaluation criteria. Parameters like level of privacy, data quality and hiding failure have been defined and the evaluations of such parameters over a set of association rule hiding algorithms have been presented. As part of future work, we plan to apply the proposed evaluation framework to other classes of privacy preservation algorithms, like *cryptography-based* and *reconstruction-based* methods. This is an important step in order to obtain a complete evaluation of privacy preserving techniques. Another interesting topic that we plan to explore is the data quality parameter and a characterization of dataset features relating to the hiding and quality of data. Finally, the use of entropy and information theory for discriminating relevant information within very large databases needs further investigation.

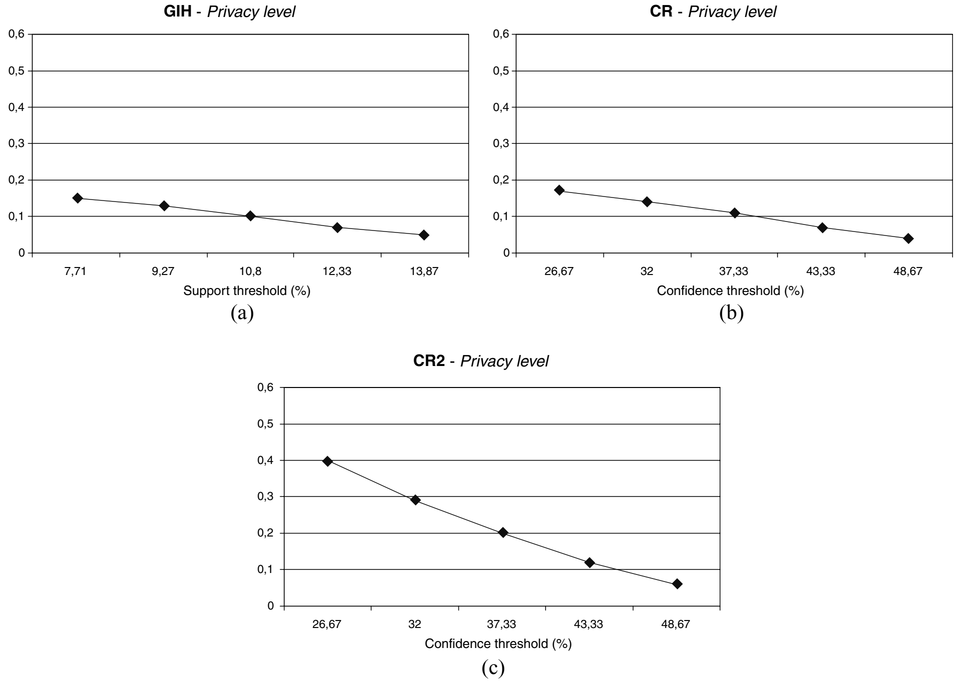


Figure 14. Entropy level ensured by Algorithms GIH, CR and CR2 when hiding the association a set of rules for increasing support/confidence values.

Appendix A: Proofs

Proof-of Theorem 4.1: First, Algorithm GIH sorts the itemset L_h wrt the items it contains and the particular items it supports. In general, sorting N number has a complexity in $O(N\log(N))$. However, in our case the length of transactions and thus of the considered itemsets has an upper bound that is very small compared to the size of the database. Then, Algorithm GIH takes $O(|L_h|)$ to sort the set L_h of large itemsets to hide according to a descending order based on their size and minimum support. For each itemset Z in L_h, it takes $O(|D|)$ in order to generate the set T_Z of the transactions in D that support Z, assuming that the transaction length is bounded by a constant. In addition to that, the algorithm sorts T_Z in ascending order of transaction size. Since the transaction length is bound, the sorting algorithm is of order $O(|T_Z|)$. Then, the item $i \in Z$ with highest minimum support is selected and a question mark is placed for that item in the transaction with minimum size, and this is repeated until the minimum support of the itemset Z goes below the MST by SM. After k iterations, the minimum support of Z will be $minsup(Z)^{(k)} = \frac{|T_Z|-k}{|D|}$; thus the number of steps required to hide Z is $|T_Z| - (MST - SM) * |D|$. Since $|T_Z| \leq |D|$ and $[(MST - SM) * |D|] \leq |D|$ we can state that Algorithm GIH requires $O(|L_h| * |D|)$ time to hide all the itemsets belonging to L_h and consequently all the sensitive rules, whose generating itemsets are stored in L_h. □

Proof-of Theorem 4.2: For each sensitive rule r to hide, Algorithm CR performs the following tasks: it generates the set T_r of transactions of D supporting r, taking $O(|D| * ATL)$, then for each transaction t in T_r it counts the number of items in t, that is of order $O(|T_r| * ATL)$. To sort the transactions in T_r in ascending order of the number of items supported the Algorithm CR takes $O(|T'_{l_r}|)$ for the reason explained above. To hide the selected rule r, the algorithm executes the inner loop until the minimum support becomes lower than *MST* by *SM* or minimum confidence becomes lower than *MCT* by *SM*. The *minsup(r)* and *minconf(r)* are initially equal to $\frac{|T_r|}{|D|}$ and $\frac{|T_r|}{|T_{l_r}|}$ respectively, and after k iterations the fractions become $\frac{|T_r|-k}{|D|}$ for the *minsup(r)* and $\frac{|T_r|-k}{|T_{l_r}|}$ for the *minconf(r)*. This implies that the inner loop executes until $\frac{|T_r|-k}{|D|} < MST - SM$ or $\frac{|T_r|-k}{|T_{l_r}|} < MCT - SM$, that is, $k = min(|D|*(minsup(r)-(MST-SM))), |D|*(minsup(r)-(MCT-SM)*minsup(l_r)))$, which can be assessed in both cases as being $O(|D|)$. The cost of each iteration is given by the *choose_item* operation, which takes $O(1)$ time. Therefore, Algorithm CR takes $O(|R_h| * A_D)$ to hide all the sensitive rules in R_h. □

Proof-of Theorem 4.3 For each rule r, Algorithm CR2 performs the following tasks: it generates the set T'_{l_r} of transactions of D that partially support l_r and do not fully support r_r, taking $O(|D| * ATL)$, then for each transaction t in T'_{l_r} it counts the number of items of l_r in t, that is of order $O(|T'_{l_r}| * ATL)$. To sort the transactions in T'_{l_r} in descending order of the calculated counts, the algorithm takes $O(|T'_{l_r}|)$ in this particular case. To hide the selected rule r, the algorithm executes the inner loop until the minimum confidence becomes lower than *MCT* by *SM*. The *minconf(r)* is initially equal to $\frac{minsup(r)}{maxsup(l_r)} = \frac{|T_r|}{|T'_{l_r}|}$, and after k iterations the fraction becomes $\frac{|T_r|}{|T'_{l_r}|+k}$. This implies that the inner loop executes $\lceil |D|(\frac{minsup(r)}{MCT-SM} - minsup(l_r))\rceil$ steps. The cost of each iteration takes $O(1)$ time, thus Algorithm CR2 performs in $O(|R_h| * A_D)$. □

Note

1. Health Insurance Portability and Accountability Act

References

Agrawal, D. and Aggarwal, C.C. 2001. On the design and quantification of privacy preserving data mining algorithms. In Proceedings of the 20th ACM SIGACT-SIGMOD-SIGART symposium on principle of database system, ACM, pp. 247–255.

Agrawal, R. and Srikant, R. 2000. Privacy preserving data mining. In Proceeedings of the ACM SIGMOD conference of management of data, ACM, pp. 439–450.

Ballou, D. and Pazer, H. 1985. Modelling data and process quality in multi input, multi output information systems. Management science, 31(2):150–162.

Domingo-Ferrer, J. and Torra, V. 2002. A quantitative comparison of disclosure control methods for microdata. In L. Zayatz, P. Doyle, J. Theeuwes and J. Lane (Eds.), Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, North-Holland, pp. 113–134.

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

Dwork, C. and Nissim, K. 2004. Privacy preserving data mining in vertically partitioned database. In Crypto 2004, Vol. 3152, pp. 528–544.

Evfimievski, A. 2002. Randomization in privacy preserving data mining. SIGKDD Explor. Newsl., 4(2):43–48.

Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J. 2002. Privacy preserving mining of association rules. In 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM-Press, pp. 217–228.

Kantarcioglu, M. and Clifton, C. 2002. Privacy preserving distributed mining of association rules on horizontally partitioned data. In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 24–31.

Kumar Tayi, G. and Ballou, D.P. 1998. Examining data quality. Communications of the ACM, 41(2):54–57.

Oliveira, S.R.M. and Zaiane, O.R. 2002. Privacy preserving frequent itemset mining. In IEEE icdm Workshop on Privacy, Security and Data Mining, Vol. 14, pp. 43–54.

Oliveira, S.R.M. and Zaiane, O.R. 2004. Toward standardization in privacy preserving data mining. In ACM SIGKDD 3rd Workshop on Data Mining Standards, pp. 7–17.

Rizvi, S. and Haritsa, R. 2002. Maintaining data privacy in association rule mining. In 28th International Conference on Very Large Databases, pp. 682–693.

Sabanci University. 2003. Models and algorithms for privacy preserving data mining (Tech. Rep.). CODMINE Project.

Schoeman, F.D. 1984. Philosophical Dimensions of Privacy: An Anthology. Cambridge University Press.

Shannon, C.E. 1948. A mathematical theory of communication. Bell System Technical Journal, 27:379–423, 623–656.

Smyth, P. and Goodman, R.M. 1992. An information theoretic approach to rule induction from databases. IEEE Transaction On Knowledge And Data Engineering, 3(4):301–316.

Sweeney, L. 2002. Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, 10(5):571–588.

Trottini, M. 2001. A decision-theoretic approach to data disclosure problems. Research in Official Statistics, 4:7–22.

Trottini, M. 2003. Decision models for data disclosure limitation. Unpublished doctoral dissertation, Carnegie Mellon University. (Available at `http://www.niss.org/dgii/TR/ThesisTrottini-final.pdf`)

University of Milan, Computer Technology Institute and Sabanci University. 2002-2003. CODMINE - IST project. (Available at `http://dke.cti.gr/CODMINE/`)

Vaidya, J. and Clifton, C. 2002. Privacy preserving association rule mining in vertically partitioned data. In 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pp. 639–644.

Verykios, V.S., Bertino, E., Nai Fovino, I., Parasiliti, L., Saygin, Y., and Theodoridis, Y. 2004. State-of-the-art in privacy preserving data mining. SIGMOD Record, 33(1):50–57.

Walters, G.J. 2001. Human Rights in an Information Age: A Philosophical Analysis. In (chap. 5). University of Toronto Press.

Wang, R.Y. and Strong, D.M. 1996. Beyond accuracy: what data quality means to data consumers. Journal of Management Information Systems, 12(4):5–34.

Willenborg, L. and De Waal, T. 2001. Elements of Statistical Disclosure Control, Vol. 155. Springer.