Classifying Data from Protected Statistical Datasets

Javier Herranz[a,⋆], Stan Matwin[b,c], Jordi Nin[d] and Vicenç Torra[e]

[a] *Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya, C. Jordi Girona 1-3, Mòdul C3, 08034 Barcelona (Spain)*
jherranz@ma4.upc.edu
⋆ *Corresponding author, Telephone: +34 934016015, Fax: +34 934015981*

[b] *School of Information Technology and Engineering, University of Ottawa, r. 5100, 800 King Edward Ave. P.O. Box 450 Stn A, Ottawa (Canada)*
stan@site.uottawa.ca

[c] *Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland*

[d] *LAAS, Laboratoire d'Analyse et d'Architecture des Systèmes, CNRS, Centre National de la Recherche Scientifique, 7, Avenue du Colonel Roche, 31077 Toulouse (France)*
jnin@laas.fr

[e] *IIIA, Artificial Intelligence Research Institute CSIC, Spanish National Research Council Campus UAB s/n, 08193 Bellaterra (Catalonia, Spain)*
vtorra@iiia.csic.es

Abstract

Statistical Disclosure Control (SDC) is an active research area in the recent years. The goal is to transform an original dataset X into a protected one X', such that X' does not reveal any relation between confidential and (quasi-)identifier attributes and such that X' can be used to compute reliable statistical information about X.

Many specific protection methods have been proposed and analyzed, with respect to the levels of privacy and utility that they offer. However, when measuring utility, only differences between the statistical values of X and X' are considered. This would indicate that datasets protected by SDC methods can be used only for statistical purposes.

We show in this paper that this is not the case, because a protected dataset X' can be used to construct good classifiers for future data. To do so, we describe an extensive set of experiments that we have run with

different SDC protection methods and different (real) datasets. In general, the resulting classifiers are very good, which is good news for both the SDC and the Privacy-preserving Data Mining communities. In particular, our results question the necessity of some specific protection methods that have appeared in the privacy-preserving data mining (PPDM) literature with the clear goal of providing good classification.

Keywords: Statistical Disclosure Control, Classification methods, Disclosure Risk, Information Loss

---

1. Introduction

The amount of digital data that is collected and stored increases every day [21]. This leads to the development of many new applications, with different requirements concerning the use of the data. We consider one such a situation, which is attracting a lot of attention from researchers in different domains. It can be informally described as follows.

An entity has some data which may contain confidential information about individuals. This entity wants to allow other users to use this data to meet their particular needs in data mining, data analysis and / or model building. An important assumption is that the entity does not know how the other users will use the data. They may want to classify or predict future values of some attribute, to obtain association rules, or to compute statistical information about the data. At the same time, the entity wants to protect the privacy of the individuals who contributed the data. There is an obvious trade-off between the achieved level of protection (privacy) and the correctness of the results that the users could obtain (utility).

Depending on the way in which the information transfer from the owner entity to the users is done, we can distinguish two situations.

- In *interactive* protocols, the owner entity does not initially publish any information. An interested user can send some queries to the owner entity, depending on the desired (and possibly private) result he wants to obtain from the data. The owner entity processes these queries by using his secret original data, and sends back some answers. Ideally, the users must obtain no additional information on the original data, apart from the requested result.

- In *non-interactive* protocols, the owner entity transforms the original data X into some protected data X', and then releases X' to the public. In this way, anybody can use X' to try to obtain some (more or less reliable) result on X.

In this work we focus on the non-interactive scenario. Different methods have been proposed to protect original data X in such a way that reliable results on X can be obtained from protected data X', while the privacy of the confidential information is preserved. Two different research groups have addressed this scenario: the Statistical Disclosure Control (SDC) community and the Privacy-preserving Data Mining (PPDM) community. On the one hand, PPDM solutions [39] are usually designed (and analyzed) having in mind that the protected data X' will be used for some specific data mining operation (such as classifier learning [2, 15, 17, 40]). On the other hand, SDC solutions [9] are typically designed (and analyzed) from a statistical perspective: one assumes that the protected dataset X' is going to be used only (or mostly) to compute statistical information about the original data X. Although the two approaches may seem different, the main goal of the two communities is basically the same: to perturb a (numerical) dataset X in some way, to produce X', such that X' is:

- *similar enough* to X so that useful information about X can be obtained from X', and
- *different enough* to X so that confidential information appearing in X cannot be obtained by observing X'.

For this reason, it is not surprising that the particular perturbation methods proposed by the two communities are very similar. For instance, SDC noise addition [4, 35] has been developed independently of PPDM data perturbation [2], and SDC microaggregation [8] has been proposed independently of PPDM k-anonymity [33, 36].

1.1. Our Contribution

The previous paragraphs raise the following natural question: what happens if SDC (respectively, PPDM) perturbation methods are analyzed from a PPDM (respectively, SDC) point of view? For example, could a protected dataset X' that has been obtained from an original dataset X by applying a SDC perturbation process be used not only to compute reliable statistical

information on X, but also to successfully perform some data mining task such as learning a classifier?

In this paper, we try to answer this last question. We could have considered the analogous problem of taking a PPDM perturbation method and analyzing the statistical quality of the results obtained from X'. But we have preferred to study SDC methods for different reasons. First of all, publicly available implementations of SDC perturbation methods are much easier to find than implementations of PPDM methods. Furthermore, the privacy level offered by SDC methods is easier to measure and tune, because the SDC community has developed its own tools, *e.g.* record linkage analysis [43], to analyze the privacy level provided by its perturbation methods.

The experiments that we have run to answer the question raised above can be described as follows. We consider an original dataset X which contains a class attribute. The records in X are partitioned in a training dataset X_1 and a testing dataset X_2, such that $X = X_1 \cup X_2$. A SDC perturbation method is applied to X_1, resulting in X_1' (the class attribute is not altered, however). Then, different classifiers are learned using X_1' as the training set and X_2 as the testing set, with respect to the class attribute. We finally analyze the performance of the whole process, in terms of its classification quality (accuracy and Area Under Curve, to be defined later). At the same time, we can easily analyze the privacy level of the whole process (measured by the disclosure risk), because it is exactly the privacy level offered by the employed SDC protection method, and privacy of SDC methods has been widely studied and tested for the last years.

We have experimented with different datasets and SDC perturbation methods. We have restricted ourselves to numerical datasets, because most of the SDC and PPDM methods work with this type of data. The obtained results depend on the methods but are in general very good, in the sense that the protected datasets lead to robust classifiers, essentially as good as those that could be constructed from the original dataset. These results are good news for practitioners in both SDC and PPDM communities. People using SDC methods are assured not only that their protection methods achieve good results for statistically-oriented operations, but also that they provide a good accuracy level for other tasks, such as classification. People in PPDM learn that there exist protection methods widely analyzed in SDC (in particular with respect to the privacy level they offer) that can be safely used to protect data that will be used to build very good classifiers. In particular this last conclusion can lead someone to debate the contribution of certain

papers in the PPDM literature that propose very specific protection methods that are designed with the sole goal of achieving very good classification results. We exemplify this argument by comparing the results obtained by SDC methods and the results obtained by a specific PPDM protection method [6].

Our results can be interpreted in a different (less positive, more preventive) way. If a confidential attribute in a dataset has few possible values, such attribute could be used as class attribute if it is not protected, and then our results imply that an adversary could guess with high success probability the value of this confidential attribute, breaking in this way the privacy of the whole system. Therefore, our results could potentially lead to a revisit of the classical SDC paradigm in which the protection method is not applied to confidential attributes. In any case, data owners should take this point into account before deciding which parts of their datasets can be released (with and/or without SDC protection).

1.2. Organization of the Paper

The rest of this paper is organized as follows. In Section 2 we describe the basic concepts of Statistical Disclosure Control; this includes the data protection scenario, measures for data protection quality and disclosure risk, and the specific SDC protection methods that we will experiment with. In Section 3 we explain the topic of classifier induction, in particular the measures that are used to decide if a classification method is good. Section 4 contains the description of our experiments, which are the main contribution of this paper, and a discussion of the obtained results. We conclude by summarizing our results and suggesting further work in Section 5.

2. Statistical Disclosure Control

Our description of statistical disclosure control is embedded in a very practical data use scenario. The original dataset X contains many records (rows), each of them consisting of some numerical values for different attributes (columns). There are three different categories of attributes in X: *identifiers* (X_{id}) which unambiguously identify one individual, *quasi-identifiers* (X_{nc}) which can identify an individual when some of them are combined, and *confidential attributes* (X_c) which may potentially contain sensitive information about the individual, and which are typically unknown to external users.

To preserve the privacy of the confidential data stored in X, but keeping at the same time as much statistical utility of the original data as possible, the following protection protocol is usually applied: (i) identifier attributes in X are either removed or encrypted, (ii) confidential attributes X_c are not modified (because they are assumed to contain the most interesting statistical information); (iii) a protection method ρ is applied to non-confidential quasi-identifier attributes (*i.e.*, $X'_{nc} = \rho(X_{nc})$). Therefore, the protected dataset that is released to the public is $X' = X'_{nc}||X_c$.

2.1. Information Loss vs. Disclosure Risk

A good SDC protection method must achieve a good trade-off between utility and privacy. In other words, the protected dataset X' must be:

- Close enough to X such that statistical values computed on X' are very similar to those that would be obtained by computing directly on X. In other words, the (statistical) *information loss* that appears in the transition from X to X' must be small.

- Different enough from X such that an attacker has a (very) small probability to obtain any correct relation between a protected record in X' with the quasi-identifier attributes corresponding to this record. This probability is denoted as the *disclosure risk*.

Information loss (IL) measures the statistical utility of the protected dataset X', comparing its usefulness with respect to the one of the original dataset X. A few different approaches are used in the SDC community to calculate the information loss. In [12] the authors calculate the average divergence of some statistical values when they are computed on both the original and the protected datasets. A probabilistic variation of these measures (PIL) was presented in [26] to ensure that the information loss value is always within the interval [0,1]. The standard PIL takes into account five specific statistical values: mean, variance, covariance, correlation coefficient and quantiles. In this paper we will use this standard PIL for computing the information loss of different SDC perturbation methods.

Regarding privacy, two types of disclosure risk measures can be considered depending on the intention and the resources of the intruder. Firstly, an intruder who observes the protected X' may know the values of some original attributes of X, that he has obtained from an external data source. The goal of this intruder is to link a protected record in X' with its correct original

attributes (*record linkage*). The percentage of correct links established by the intruder between the original and protected datasets is therefore a measure of risk. This *Linkage Disclosure* (LD) risk measure depends on the number of attributes that the intruder is assumed to know. In our experiments, we have assumed that the intruder knowledge varies from one to all attributes. The final LD value is computed as the average percentage of correctly linked records in each case.

Secondly, if the intruder cannot obtain any information from an external data source, he can still try to get an approximation of the original values. *Interval Disclosure* (ID) is one of the approaches to model this scenario. In our experiments, the ID risk is computed as the average percentage of original values falling into an interval defined around the corresponding protected value. The interval is defined as a percentage, between 1% and 10% of the values. This measure is very similar to the measure presented in [2], where authors quantify disclosure risk measuring how closely the original values of a protected attribute can be estimated. The final disclosure risk (DR) measure is computed as DR = 0.5 LD +0.5 ID.

2.2. Some SDC Protection Methods

In the rest of this section we describe the specific SDC protection methods that we are going to consider in this work (to see if they can be successfully combined with classification techniques). We can distinguish three main families of SDC protection procedures [37]: (i) perturbative methods which introduce some kind of error into the original data, (ii) non-perturbative methods which construct the new dataset reducing its quality (*e.g.*, it is less specific) but without introducing new values, and (iii) synthetic data generators, which produce synthetic data that resembles the original one. For an overview of the methods see *e.g.* [1, 13, 42].

In our experiments we will consider some perturbative methods (additive and multiplicative noise, rank swapping, microaggregation) and a synthetic data generator (IPSO). They are described in detail below using the following generic notation: X represents the original dataset, with n instances (or records) and m columns (or attributes). Therefore, x_{ij} represents the value of instance (individual) i for attribute j.

2.2.1. Additive and Multiplicative Noise

This is perhaps the simplest and most intuitive data perturbation method. In additive noise [19], each value x_{ij} of the original dataset X is replaced with

$x'_{ij} = x_{ij} + \epsilon$, where ϵ is the noise.

In the literature, the simplest approach is that ϵ is a normally distributed error drawn from a random variable $\epsilon \sim N(0, \sigma^2_\epsilon)$, and that the variance of ϵ is proportional to those of the original attributes. Thus, if σ^2_j is the variance of an attribute j, then $\sigma^2_\epsilon = \alpha\sigma^2_j$ for some α. When $Cov(\epsilon_{j_1}, \epsilon_{j_2}) = 0$ for all attributes $j_1 \neq j_2$ (this is the case of uncorrelated noise), X and X' have the same means and covariances, but not the same variances ($\sigma^2_j = (\sigma')^2_j/(1+\alpha)$) nor correlation coefficients.

In contrast to that, correlated noise addition preserves correlation coefficients and means. In this case, $\epsilon \sim N(0, k\Sigma)$, where Σ is the covariance matrix of X. A good survey on additive noise is given in [4], where the author presents the basic methods as well as some more elaborated ones.

A very related approach is multiplicative noise [20, 24]: now each original value x_{ij} is replaced with $x'_{ij} = x_{ij} \cdot \epsilon$, where the noise ϵ follows a specific distribution which depends on the original values for attribute j.

2.2.2. Rank Swapping

Rank swapping [7] with parameter p and with respect to an attribute j can be defined as follows. Firstly, the records of X are sorted in increasing order of the values x_{ij} of the attribute j. To simplify notation, let us assume that the records are already sorted, that is $x_{ij} \leq x_{\ell j}$ for all $1 \leq i < \ell \leq n$. Then, each value x_{ij} is swapped with another value $x_{\ell j}$, randomly and uniformly chosen from the limited range $i < \ell \leq i + p$. When rank swapping is applied to a dataset, the algorithm explained above is run for each attribute to be protected, in a sequential way.

The parameter p is used to control the swap range. Normally, p is defined as a percentage of the total number of records in X. Therefore, when p increases, the difference between x_{ij} and $x_{\ell j}$ may increase accordingly. This fact increases privacy, but of course the differences between the original and the protected dataset are higher, thereby decreasing the statistical utility of the data. As noted in [30], the fact that each value is swapped with a value in a fixed, closed (and possibly public) rank makes this basic rank swapping method more prone to re-identification attacks, decreasing privacy protection offered by this method. To mitigate this drawback, two variants of rank swapping are proposed in [30], where some values (with a small but still non-negligible probability) are swapped with values out of the theoretical rank. In the experiments performed in this paper we will use one of these variants of rank swapping, called rank swapping p-distribution. It defines

the swap interval using a normal probability distribution defined by $\mu = \sigma = 0.5 \cdot p$. This modification makes possible that any value in the dataset can be potentially selected for a swap with a given element, with a probability that decreases with the distance to the given element. In this way, the close swap interval is replaced by an open one, solving the privacy issues of classical rank swapping.

The advantages of rank swapping and its variants are their simplicity, their low computation cost and the fact that these methods preserve the univariate attribute distributions and, therefore, the average and variance vectors of the original data.

2.2.3. Microaggregation

Microaggregation is one of the most common methods used to obtain k-anonymity for numerical data: groups of k close records are identified and substituted by their centroid. In this way, an original record is protected against disclosure risk in the sense that k protected records have exactly the same probability to correspond to that original record. To achieve minimum information loss, the goal is to find an optimal microaggregation that minimizes the sum of distances between original records and protected records (centroids). Since the optimal solution to this problem is NP-hard [31] (for the general multivariate case), many effective heuristic algorithms have been proposed to provide good quality results.

Among these methods, we can list the Maximum Distance to Average Vector (MDAV) algorithm [10] or the similar (but more efficient) Centroid-based fixed-size (CBFS) algorithm [22]. We will use CBFS for our experiments. It works as follows. Firstly, the average record $\bar{x}$ of all records in X is computed. The most distant record x_r to the average record $\bar{x}$ is considered, and a cluster around x_r is formed, containing x_r together with the $k-1$ closest records to x_r. All records belonging to this cluster are removed from X. Among the remaining records, the most distant record to $\bar{x}$ is considered, a cluster is formed, etc. The process is repeated until all the records are assigned to one cluster. Finally, the protected dataset X' is built by replacing each original record in X with the centroid of the cluster to which the record belongs.

In the last years, some researchers have pointed out that k-anonymity may not be enough to ensure privacy. For example, consider our scenario for data protection, where confidential attributes X_c are not modified in X', and suppose that all the protected records in a cluster have exactly the same confi-

dential values. Then, even if an intruder cannot know which exact protected record corresponds to a given original record, he will know for sure what are the confidential values for this record. The notions of p-sensitivity [38], l-diversity [25] and t-closeness [23] have been proposed to address this weakness of k-anonymity. The goal is to ensure that the distribution of the confidential values in each of the final clusters satisfies some properties (a minimum number of different values, a minimum entropy value, a distribution very close to the distribution of the confidential values in the entire dataset X, etc.). In the literature we can find some MDAV variants that provide protected datasets X' satisfying these notions (see, for instance [11, 34]). In our experiments, we have tuned CBFS (without worrying about efficiency optimizations) in several ways to provide either p-sensitivity or l-diversity. These modifications, of course, lead to a decrease of the statistical utility, with respect to standard CBFS.

2.2.4. The IPSO Synthetic Data Generators

In this method the original dataset is used to produce from it another dataset with different attribute values, but with certain identical statistical characteristics. The original idea [5] assumes that data can be partitioned into a single non-confidential attribute X_{nc} and a single confidential attribute X_c, dependent on X_{nc}. A linear regression model is fit to this data, and this model is used to generate synthetic values of X'_c from the original values X_{nc}. Random, normally distributed noise is added to X'_c yielding Y'_c. Finally, the values X_{nc} and Y'_c are released. This is easily extended to the case of multiple non-confidential and confidential attributes.

This idea is extended in [28], so that besides preserving the mean vector and the covariance matrix as [5] does, the method also guarantees similarity of the synthetic confidential values to the original confidential values. This variant assumes first that the quasi-identifier attributes X_{nc} follow a multivariate normal distribution with covariance matrix Σ and mean vector $x_i B$, where B is the matrix of regression coefficients.

Let $\hat{B}$ and $\hat{\Sigma}$ be the maximum likelihood estimates of B and Σ derived from the original dataset (X_{nc}, X_c). If a user fits a multiple regression model to (X'_{nc}, X_c), he will get estimates $\hat{B}'$ and $\hat{\Sigma}'$ which, in general, are different from the estimates $\hat{B}$ and $\hat{\Sigma}$ obtained when fitting the model to the original data (X_{nc}, X_c) Therefore, an extension in [28] changes X'_{nc} into X''_{nc} in such a way that the estimate $\hat{B}''$, obtained by multiple linear regression from (X''_{nc}, X_c), satisfies $\hat{B}'' = B$.

Another extension produces a dataset X'''_{nc} such that when a multivariate multiple regression model is fitted to (X'''_{nc}, X_c), both statistics $\hat{B}$ and $\hat{\Sigma}$ obtained on the original data (X_{nc}, X_c) are preserved. This last variant, known as IPSO, will be the one we will use later in our experiments.

For these variants of IPSO based on multiple linear regression, there is a degree of freedom to choice the number g of attributes that are considered to build the regression model. For example, if $g = 2$, then the first two attributes are considered to built the regression model for the third and fourth attributes, which are considered to build the model for the fifth and sixth attributes, and so on.

3. Classifier Induction

The goal of this paper is to investigate if one can perform good classifier induction starting from a protected dataset X' that has been output by a SDC perturbation method. The task of classifier induction is to learn from specific examples of instances, each represented by a vector of attribute values and *labeled* by class values, a general mapping from the attribute space to classes that allows to classify or predict future instances. More precisely, we can describe classifier induction as follows: data are given as vectors of attribute values, where the domain of possible values for attribute j is denoted as A_j, for $1 \leq j \leq N$. Moreover, a set $C = \{c_1, ..., c_k\}$ of k classes is given; this can be seen as a special attribute or label for each record. Often $k = 2$, in which case we are learning a binary classifier. Inducing, or learning a classifier, means finding a mapping F: $A_1 \times A_2 \times ... \times A_N \rightarrow C$, given a finite *training* set $X_1 = \{< x_{ij}, c_i >, 1 \leq j \leq N, c_i \in C, 1 \leq i \leq M \}$ of M labeled examples. In general we want F to be expressed in a certain language, *e.g.* F can be a set of *n-1* dimensional hyperplanes partitioning an n-dimensional space into k subspaces, or a decision tree with leaves belonging to C, or a set of rules with consequents in C. We also want F to perform well, in terms of its predictive power, on (future) data not belonging to X_1.

3.1. Measuring the Quality of Classifiers

Predictive power of a classifier is its ability to produce correct labels on unseen data. While we cannot measure the performance of a classifier on future data, we can estimate this performance by evaluating the behavior of the classifier on a testing dataset $X_2 \subseteq A_1 \times A_2 \times ... \times A_N \times C$, on purpose withheld from the learning process (*i.e.* $X_1 \cap X_2 = \emptyset$).

The most simple and common measure of classifier performance is obviously the percentage of records in the testing dataset that are correctly classified. This percentage is called the *accuracy* (ACC). However, the accuracy is inadequate for imbalanced datasets. For instance if the distribution of two classes is 90%-10%, then a trivial classifier always predicting the majority class has accuracy 90%. Therefore, performance evaluation measures that are not sensitive to class distribution are also desirable; the *Area Under Curve* (AUC) is one such measure.

To define AUC, let us first consider the binary classification case, that we extend later to a k-class classification, with $k > 2$. For a binary classifier with classes T and F, the four possible situations when a label is assigned to a new record can be represented in a *confusion matrix*

label	assigned=T	assigned=F
true=T	TP	FN
true=F	FP	TN

TP and TN represent the situations where the correct label is assigned. From the confusion matrix, we can also compute two useful measures of classifier performance, the True Positive Rate $TPR = \frac{TP}{TP+FN}$ and the False Positive Rate $FPR = \frac{FP}{FP+TN}$. We can observe that since these rates are defined using only information about one class (a row in the above table), they are independent of the class distribution. The confusion matrix can be extended from a square table for a binary classification task to a k-dimensional table for the k-class task. Similarly, the ratios FPR and TPR generalize to the k-class setting, by summing up $k-1$-dimensional slices of the table for the denominator of these ratios. Fawcett [16] showed how the AUC can be computed from these TPR and the FPR rates. The AUC measures the quality of separation between classes. Both the ACC and AUC measures are embedded in the WEKA [44] system, that we will use in our experiments.

3.2. Some Well-Known Classifiers

While many classifiers have been defined in the literature, none is universally better than the others in terms of their predictive power. The choice of the classifier depends therefore on the characteristics of the data and on the requirements of the classification task and the model built (computational cost, stability, interpretability etc.) The main classifiers used in data mining

practice are Decision Trees, Naive Bayes, k-Nearest Neighbor (k-NN), and the Support Vector Machine (SVM).

The Decision Tree classifier [32] builds tests of single attribute values that lead to subsets of instances with highly predictable class label. Decision trees are highly popular due to their interpretability: the learned classifier can be logically interpreted and reasoned with by the user of the model.

The SVM classifier [41] lifts the classification task from its original data space to a much more high-dimensional feature space, and then learns a linear classifier in that space using the so called *kernel trick* that performs the computation in the data space. SVM often (but not always) produces higher performance than other classifiers, but suffers from lack of interpretability.

The Naive Bayes classifier [14] predicts a class by combining, in a simple manner, prior probabilities of a class value as determined by values of each individual attribute. Naive Bayes is highly efficient to learn and to apply.

The k-nearest neighbor classifier [3] determines the class of an instance by choosing the most common class of its k closest neighbors. The method is often chosen due to understandability of its underlying principle by the users. Furthermore, different pruning techniques can be applied to dramatically improve the efficiency of this method, which makes it suitable for classifying over very large datasets.

The last two classifiers are *lazy*, *i.e.* no explicit, intensional classifier is built in the learning process, and the class of a test instance is determined by using the training data itself. All the classifiers have certain internal parameters (*e.g.* degree of pruning for Decision Trees, value of k for k-NN, or kernel function for SVM) that need to be decided when a classifier is used.

4. Classifying from SDC Protected Datasets

This section contains the central part of our work. Our goal is to show, through various experiments with different datasets and SDC protection methods, that a dataset that has been protected with a SDC method can be used to construct a reliable classifier for future (original) data.

4.1. The Scenario

Our experiments are designed to model the following scenario. An entity (*e.g.* a statistical agency, or a medical research establishment) has a dataset X which contains information obtained from a number of individuals, and

some of this information is confidential (*e.g.* income levels, medical conditions, etc.). The entity wants to release some protected version X' of the dataset to the public, so that external parties (*e.g.* researchers, media, etc.) can use the data for their purposes. In order to be able to securely release the data, the entity must take steps to protect the privacy of individuals who provided the data.

Suppose that the dataset X contains numerical information, and the entity thinks that most of the external parties that will use X' will be interested in statistical analysis on the data in X. For this reason, the entity decides to protect X with a SDC perturbation method, releasing the resulting protected dataset X' to the public. Now, assume that one of the external parties is not interested in statistical analysis, but he would like to use X to build a classifier, in order to predict the class (a special attribute) of future records, out of X but with the same attributes. The question is: could this party use X' to build a reliable classifier, similar to the one that he would have obtained by using X? We hope the answer to this question to be 'yes'. Otherwise, the entity should help this specific external party by computing a different protected version of X, maybe using a PPDM technique, specifically designed for classification. Furthermore, in this case the external party would be forced to tell to the entity which kind of use he is going to make of the data, which can be by itself a privacy breach for this party.

In order to run experiments that will allow us to empirically answer the above-mentioned question, we first have to decide on the datasets used, the classifiers studied, and the measures applied to evaluate the quality of the whole process (privacy, quality of the resulting classifiers, etc.).

4.2. Selected Classifiers, Datasets and SDC Methods

We have used the implementations available in WEKA [44], for the four classifiers: Decision Trees (DT), Naive Bayes (NB), k-Nearest Neighbors (k-NN), and Support Vector Machines (SVM), which are the most popular and effective classifiers used in everyday data mining practice.

Regarding the datasets X, we have selected two datasets from the UCI repository [29] and one dataset extracted from the U. S. Census [27] using the Data Extraction System (DES). These three data sets have the following properties: (i) the attributes are numerical (because the SDC protection methods that we will apply are designed to work with numerical data); (ii) there is an attribute with a few (and uniformly distributed) values - this attribute can be chosen as the class attribute. The description of these

	Abalone	Vehicle	Census
Records	4177	846	13518
Attributes	9	19	13
Classes	M(36%)	opel(25%)	L (20%)
	F(31%)	saab(25%)	VL (20%)
	I(33%)	bus(25%)	M (20%)
		van(24%)	H (20%)
			VH (20%)

Table 1: Datasets description.

datasets can be found in Table 1. For the class attribute, we have noted the percentage of records in the dataset that belong to each class; in this way, we will be able to compare the performance (accuracy, in particular) of the selected classifiers with a trivial classifier that simply assigns the majority class to each new record (from the testing dataset).

Finally, we have considered the Statistical Disclosure Control protection methods described in Section 2: additive and multiplicative noise addition, rank swapping, microaggregation (with extensions to achieve p-sensitivity or l-diversity) and IPSO. These are quite representative of SDC perturbative protection techniques, because: (i) noise addition is applied individually to each value in the database; (ii) rank swapping is applied at the same time to all the values of an attribute; (iii) microaggregation takes into account all the values of some block of attributes (likely, all the attributes, and therefore the whole dataset is processed at the same time); and (iv) IPSO is a synthetic data generator.

4.3. Experiments and Results

In order to test the four classifiers, we have followed the standard procedure of 10-fold cross validation, adapted to the situation that we are considering. Recall that the owner of a dataset X_1 wants to release a protected version X_1' of X_1 to the public, in order to allow for data analysis. If an external analyst is interested in classification, his goal will be to predict the value of some class attribute for some new records, out of X_1, using a classifier learned from X_1'. Of course, these new records (that form a different dataset X_2) are assumed to be given in their original form, without any pro-

tection. The SDC method will not be applied to the class attribute, and so the original values for this class attribute will be publicly released.

To simulate this situation, we take our selected datasets X (Abalone, Vehicle or Census) and we split them into two parts, the training and the testing datasets, following the standard 10-fold cross-validation protocol. To do so, we select at random 10% of the records in X: this will be the testing dataset X_2. The remaining records (90%) will be the training dataset, X_1, which will be protected with the corresponding SDC method. Note that we have to repeat this process 10 times ensuring that all original records are included once in the testing dataset X_2. Then, each of the four classifiers is run on the protected training dataset X_1', and tested with each record in the (non-protected) testing dataset X_2. We will use two performance measures: the accuracy (ACC) and the Area Under Curve (AUC), which are defined in Section 3. For each combination dataset - SDC method, we run ten independent executions of this 10-fold cross validation routine, and then we average the results (such average is needed because noise addition and rank swapping are non-deterministic protection methods).

We have applied many different parameterizations for each of the tested SDC methods, from weak protection to a strong protection. For additive noise addition, we have used the following values for the variance modification, $\alpha \in \{1, 2, 3, 5, 10, 25, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 750, 1000\}$. For multiplicative noise addition, we have used $\alpha \in \{2.5, 5, 7.5, 10, 12.5, 15, 20, 25, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$. For rank swapping p-distribution, we have considered the values $p \in \{1, 3, 5, 12, 25, 35, 50\}$. Recall that p determines the average of the normal distribution which defines the length of the interval where swapping is done. This swap interval is computed as a percentage of the total number of records in the dataset. For k-microaggregation, we have used CBFS with different values for the parameter $k \in \{5, 10, 15, 20, 25, 50, 75, 100, 125, 150\}$. Regarding our versions of CBFS which ensure p-sensitivity or l-diversity, we have used the same ten values for parameter k, and then values $p \in \{2, 3\}$ and $l \in \{2, 3\}$ for the parameters related to sensitivity and diversity of the chosen class attribute. Note that the values of p, l make sense only when they are smaller than the number of possible values for the class attribute (which is 3,4 or 5 in the considered datasets). Finally, for the IPSO synthetic method, we have considered different values for the number g of attributes that are taken to build the regression models.

The classifier method k-NN depends on a parameter k. For each dataset,

			Accuracy, ACC				Area Under Curve, AUC			
	PIL	DR	DT	NB	k-NN	SVM	DT	NB	k-NN	SVM
Original	0.00%	100.00%	54.22%	54.78%	53.93%	54.56%	71.60%	73.30%	71.60%	70.30%
Noise, $\alpha = 3$	7.90%	74.56%	54.39%	51.81%	53.36%	54.49%	73.09%	73.41%	71.48%	70.50%
Noise, $\alpha = 10$	24.65%	38.95%	53.67%	51.88%	51.62%	54.37%	73.24%	73.42%	70.55%	70.49%
Noise, $\alpha = 100$	73.94%	4.10%	51.04%	52.21%	48.17%	53.20%	72.06%	73.98%	66.47%	69.50%
MultNoise, $\alpha = 5$	13.50%	50.81%	54.44%	51.90%	52.36%	54.39%	73.51%	73.42%	71.22%	70.50%
MultNoise, $\alpha = 10$	24.81%	24.75%	54.20%	51.76%	54.20%	54.32%	73.15%	73.42%	72.67%	70.41%
MultNoise, $\alpha = 100$	74.29%	0.00%	50.73%	52.12%	50.90%	53.27%	71.00%	73.90%	68.10%	69.52%
RS p-dist, $p = 2$	22.12%	51.12%	53.19%	51.23%	53.99%	54.37%	70.95%	73.24%	74.15%	70.57%
RS p-dist, $p = 10$	29.00%	23.49%	53.55%	51.85%	54.35%	54.18%	71.84%	73.52%	73.17%	70.40%
RS p-dist, $p = 50$	39.96%	7.80%	40.63%	50.56%	37.32%	53.20%	59.24%	73.17%	57.75%	69.50%
CBFS, $k = 5$	39.05%	13.73%	54.56%	51.64%	54.01%	54.54%	74.10%	73.29%	73.26%	70.62%
CBFS, $k = 25$	58.08%	6.65%	53.31%	51.95%	53.05%	54.01%	73.48%	73.10%	74.22%	70.23%
CBFS, $k = 100$	63.55%	4.32%	51.30%	51.59%	53.53%	54.10%	71.16%	73.24%	74.56%	70.31%
CBFS 2-sen, $k = 25$	58.08%	0.55%	53.31%	52.00%	53.05%	54.13%	73.44%	73.10%	74.22%	70.30%
CBFS 3-sen, $k = 25$	73.00%	0.00%	45.00%	42.00%	43.00%	41.00%	62.00%	61.00%	63.00%	60.00%
CBFS 2-div, $k = 25$	61.55%	0.40%	52.72%	51.57%	52.84%	54.37%	72.13%	73.24%	73.09%	70.36%
CBFS 3-div, $k = 25$	86.00%	0.00%	38.00%	39.00%	38.00%	40.00%	60.00%	61.00%	62.00%	63.00%
IPSO $g = 2$	65.09%	1.66%	52.81%	51.52%	50.11%	53.39%	72.36%	73.61%	68.06%	69.66%
IPSO $g = 3$	58.93%	4.93%	51.45%	51.09%	49.87%	52.41%	69.58%	73.22%	68.24%	68.81%
IPSO $g = 4$	58.56%	1.81%	52.05%	51.23%	50.68%	52.52%	70.41%	73.22%	68.52%	69.00%

Table 2: Results obtained with the database Abalone and different SDC perturbation methods.

we have done a 10-fold cross validation experiment with k-NN for different values of k, between 2 and 20, without protecting the training dataset this time, and we have selected the value of k which gives the best results.

To illustrate the outcome of our executions, we include in Tables 2, 3 and 4 the results obtained with some of these parameterizations of SDC methods, and datasets Abalone, Census and Vehicle, respectively. We have included in these tables three different but significative parameterizations for each SDC method, reflecting weak protection, medium protection and high protection levels.

The tables must be analyzed as follows. The first row corresponds to the results obtained on the original dataset X, without any protection. Of course, the disclosure risk DR when releasing the original dataset is 100, and the information loss PIL is 0. The rest of columns give the classification results (both ACC and AUC) obtained on the original dataset X when running the five different classifiers. Then, in the remaining rows, the specified parameterization of the specified SDC method is applied to the dataset $X_1 \subset X$, to produce X_1'; the DR and PIL columns give the disclosure risk and information loss produced by this protection, and the rest of columns

			Accuracy, ACC				Area Under Curve, AUC			
	PIL	DR	DT	NB	k-NN	SVM	DT	NB	k-NN	SVM
Original	0.00%	100.00%	92.75%	76.95%	83.79%	87.75%	97.60%	94.40%	95.20%	95.80%
Noise, $\alpha = 3$	10.65%	79.34%	92.02%	76.82%	84.15%	87.87%	97.69%	94.45%	95.61%	95.93%
Noise, $\alpha = 10$	27.47%	41.17%	90.17%	76.28%	82.85%	87.34%	97.46%	94.28%	95.43%	95.77%
Noise, $\alpha = 100$	77.59%	0.12%	59.43%	63.25%	52.29%	73.53%	80.97%	90.07%	81.98%	91.59%
MultNoise, $\alpha = 5$	14.85%	68.59%	91.77%	76.73%	83.37%	87.83%	97.68%	94.42%	95.53%	95.91%
MultNoise, $\alpha = 10$	28.00%	41.30%	89.97%	76.29%	82.95%	87.39%	97.38%	94.29%	95.41%	95.77%
MultNoise, $\alpha = 100$	77.18%	0.10%	60.14%	64.17%	53.71%	73.95%	82.29%	90.15%	82.39%	91.71%
RS p-dist, $p = 2$	27.82%	58.60%	91.85%	76.49%	83.55%	87.45%	97.56%	94.61%	95.55%	95.81%
RS p-dist, $p = 10$	38.68%	2.47%	81.62%	73.98%	76.62%	83.20%	94.87%	93.93%	93.78%	94.71%
RS p-dist, $p = 50$	40.35%	0.00%	42.68%	35.69%	39.30%	50.61%	72.60%	75.23%	70.88%	78.63%
CBFS, $k = 5$	44.01%	8.38%	91.11%	76.53%	82.42%	88.99%	97.55%	94.25%	95.44%	96.16%
CBFS, $k = 25$	59.48%	1.10%	90.01%	74.58%	78.65%	88.12%	97.10%	93.79%	94.72%	95.86%
CBFS, $k = 100$	67.14%	0.00%	86.23%	72.40%	76.62%	84.12%	96.59%	92.87%	94.50%	94.58%
CBFS 2-sen, $k = 25$	77.99%	0.85%	89.69%	73.21%	78.17%	86.41%	96.91%	93.36%	94.54%	95.27%
CBFS 3-sen, $k = 25$	86.58%	0.29%	77.92%	59.68%	72.06%	74.55%	94.47%	88.82%	92.25%	90.98%
CBFS 2-div, $k = 25$	88.00%	0.41%	76.62%	50.16%	69.20%	69.97%	93.33%	85.04%	91.24%	89.65%
CBFS 3-div, $k = 25$	92.73%	0.00%	53.58%	31.95%	52.94%	48.74%	81.45%	70.63%	84.02%	78.41%
IPSO $g = 2$	58.39%	1.19%	83.87%	72.24%	68.36%	80.02%	96.04%	93.23%	91.08%	93.72%
IPSO $g = 3$	58.22%	0.99%	57.53%	59.51%	52.83%	53.38%	82.31%	89.02%	81.76%	81.34%
IPSO $g = 6$	39.60%	6.37%	82.35%	70.22%	67.99%	75.88%	94.60%	92.05%	89.48%	91.62%

Table 3: Results obtained with the database Census and different SDC perturbation methods.

give the results obtained by the different classification methods, executed over the training dataset X_1' and testing dataset $X_2 = X - X_1$.

Recall that the values DR and PIL are independent from the classification tasks. The information loss (PIL) value obviously increases with the protection parameter; on the other hand, the disclosure risk (DR) value always decreases when the protection is higher.

Regarding the values describing the performance of the classifiers (ACC and AUC), we have a clear expectation that these values should decrease as the level of protection applied to a dataset increases. Let us observe that this trend holds generally in Tables 2 and 3: the AUC and ACC values decrease for each protection technique as its parameters increase, ensuring stronger protection. We can also observe that there are exceptions from this expected behavior for the lower levels of protection, e.g. in Table 3, the ACC values for the datasets protected with CBFS, $k = 5, 25$ are higher than the ACC values for the unperturbed data (the same happens for other values of ACC and AUC, and some lower protection levels for all methods except rank swapping). We want to emphasize that these small differences in favor of the protected data do not necessarily mean that the performance on this

			Accuracy, ACC				Area Under Curve, AUC			
	PIL	DR	DT	NB	k-NN	SVM	DT	NB	k-NN	SVM
Original	0.00%	100.00%	71.98%	44.79%	71.93%	73.35%	87.50%	76.75%	89.75%	87.25%
Noise, $\alpha = 3$	21.16%	84.65%	68.67%	44.43%	70.57%	73.51%	86.53%	77.42%	89.76%	86.42%
Noise, $\alpha = 10$	30.82%	65.02%	68.67%	43.96%	70.34%	73.16%	86.57%	77.27%	90.15%	86.35%
Noise, $\alpha = 100$	80.15%	5.62%	45.28%	41.47%	50.24%	63.82%	72.15%	76.60%	76.95%	80.73%
MultNoise, $\alpha = 5$	23.55%	89.41%	68.15%	44.95%	71.27%	73.74%	84.35%	76.12%	89.25%	86.48%
MultNoise, $\alpha = 10$	30.62%	82.07%	67.73%	45.02%	70.63%	73.12%	83.71%	73.47%	89.08%	86.23%
MultNoise, $\alpha = 100$	80.26%	2.24%	44.88%	39.68%	50.36%	62.47%	71.32%	73.85%	76.21%	86.12%
RS p-dist, $p = 2$	21.14%	69.03%	68.79%	45.25%	70.93%	73.29%	84.95%	75.99%	89.74%	85.95%
RS p-dist, $p = 10$	34.44%	41.18%	66.77%	45.03%	62.89%	66.69%	85.74%	75.64%	85.53%	82.36%
RS p-dist, $p = 50$	47.49%	6.69%	42.42%	41.59%	36.54%	47.52%	65.53%	72.97%	65.04%	72.63%
CBFS, $k = 5$	53.25%	14.56%	62.76%	44.43%	65.94%	71.75%	83.07%	75.80%	87.30%	85.77%
CBFS, $k = 25$	69.82%	5.63%	49.64%	42.30%	54.13%	64.43%	73.02%	72.02%	82.00%	81.36%
CBFS, $k = 100$	76.39%	2.75%	28.60%	32.72%	42.44%	29.66%	55.08%	58.58%	70.48%	59.48%
CBFS 2-sen, $k = 25$	70.02%	1.14%	51.05%	42.06%	54.24%	65.26%	72.60%	71.92%	82.04%	81.63%
CBFS 3-sen, $k = 25$	70.48%	1.13%	51.18%	42.89%	53.42%	64.43%	74.13%	71.77%	81.70%	81.43%
CBFS 2-div, $k = 25$	70.83%	1.08%	48.11%	40.65%	55.07%	63.48%	69.33%	70.22%	81.25%	80.82%
CBFS 3-div, $k = 25$	74.14%	1.01%	42.79%	38.76%	54.74%	58.53%	69.88%	69.90%	79.21%	79.81%
IPSO $g = 2$	62.49%	4.28%	46.34%	36.64%	37.23%	41.72%	66.21%	59.21%	64.63%	65.27%
IPSO $g = 3$	57.69%	4.19%	53.43%	37.93%	44.21%	54.25%	71.30%	64.06%	69.34%	70.70%
IPSO $g = 9$	37.35%	9.77%	58.05%	38.99%	60.65%	64.31%	80.13%	71.03%	84.30%	81.16%

Table 4: Results obtained with the database Vehicle and different SDC perturbation methods.

data is better: they may well be the artefact of cross-validation in the case when the differences in performance are not statistically significant. More precisely, for these lower levels of protection there are no differences between the performance on protected and non-protected data due to the protection applied, but only due to the random partition into the training and testing sets in cross-validation.

Results of experiments in Tables 2, 3 and 4 confirm the expectations: the performance remains essentially unchanged for the lower levels of protection, and it degrades as the level of protection grows. The surprising (and positive) fact is that the degradation measured in the decrease of ACC and AUC is fairly slow.

4.4. Discussion of the Results

There are two different ways of analyzing the results included in the three tables. The first one is by looking at the columns of the tables, to see which classification methods perform better. This analysis does not lead to very revealing results, because the behavior of the different classification methods seems to depend on the specific dataset. For example, for the Abalone dataset, all five classifiers give very similar results; for the Census dataset,

the Decision Tree (DT) classifier seems to give the best accuracy (ACC); finally, for the Vehicle dataset, the best ACC results are mostly obtained by Support Vector Machines (SVM). In all cases, there are not significant differences among the AUC (Area Under Curve) results obtained by the different classification methods.

As we have mentioned at the end of previous section, the most interesting (and maybe surprising) conclusions of our experiments are obtained when looking at the rows of the tables. It is very easy to see that, while the values in the DR and PIL columns change a lot from one row to the following one (corresponding to a higher protection with the same SDC method), the values in the columns corresponding to the classification measures ACC and AUC change very smoothly. This means that the SDC protection methods are more robust with respect to classification purposes than with respect to statistical analysis, which is surprising because these methods have been initially designed with the aim to preserve statistical values of the original data as much as possible.

This fact, *i.e.* the different evolution of PIL on the one hand and ACC,AUC on the other hand, when the level of protection applied to a dataset increases, is illustrated through some additional figures where we have included the results obtained with all the considered parameterizations of the SDC methods. There is one figure for each SDC method: Figure 1 for additive noise addition, Figure 2 for multiplicative noise addition, Figure 3 for rank swapping, Figure 4 for CBFS, Figure 5 for CBFS with 3-sensitivity and Figure 6 for CBFS with 3-diversity.

Each of the figures contains three graphics, one for each of the datasets (Abalone, Census and Vehicle). In each graphic, the x-axis contains the values of the parameter of each method (α, p and k for noise addition, rank swapping and CBFS, respectively). We have then drawn four curves to measure the results of our experiments. The curves measure how the disclosure risk (DR), information loss (PIL), accuracy (ACC) and area under curve (AUC) behave while the level of data protection increases, with respect to the results obtained on the original dataset (without any protection). That is, since the disclosure risk DR(0) of releasing the original dataset is 100, the curve DR for the disclosure risk variation represents the function 100-DR(x). Analogously, the curve PIL for the information loss represents the function PIL(x). Regarding the values related to classification, the curve for accuracy represents the normalized degradation of ACC as $100 \cdot$ (ACC(0)- ACC(x)) / ACC(0), where ACC(x) is the accuracy obtained by the best classification

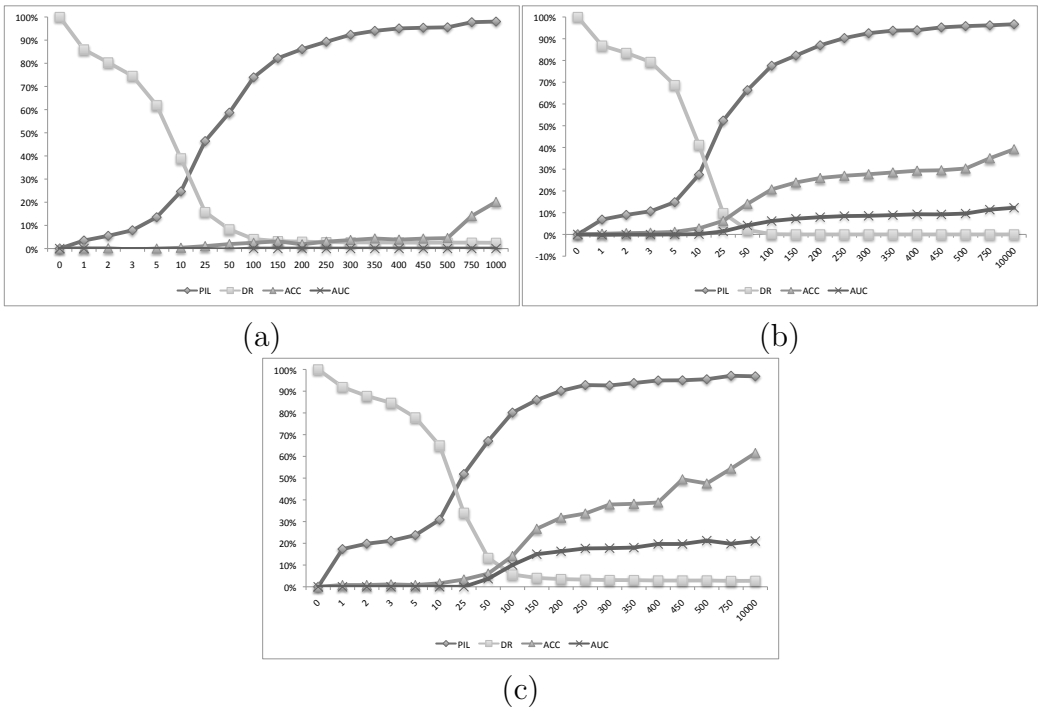

Figure 1: Graphic representation of the results obtained using the additive noise protection method on the (a) Abalone, (b) Census and (c) Vehicle datasets.

method when applied to a dataset protected with parameter x; in particular, ACC(0) is the accuracy obtained by the best classification method when applied to the original (not protected) dataset. The same ideas apply to the curve for area under curve (AUC).

The conclusions that can be drawn from these figures are significant. In particular, for all the SDC protection methods (excluding CBFS with 3-sensitivity or 3-diversity, in Figures 5 and 6), the quality of the classification stays very close to the classification results obtained with the original dataset, even when the protection level is very high. In detail, when the disclosure risk of the SDC methods reaches 10% (which is considered as a very good level of protection by statistical agencies), the relative variation between classifying through original data and classifying through protected data is very small, always less than 10%. In contrast, at these levels of protection, the information loss (PIL) value has typically increased to levels around 40% or 50%.

Later, for even higher levels of protection, the results of ACC and AUC

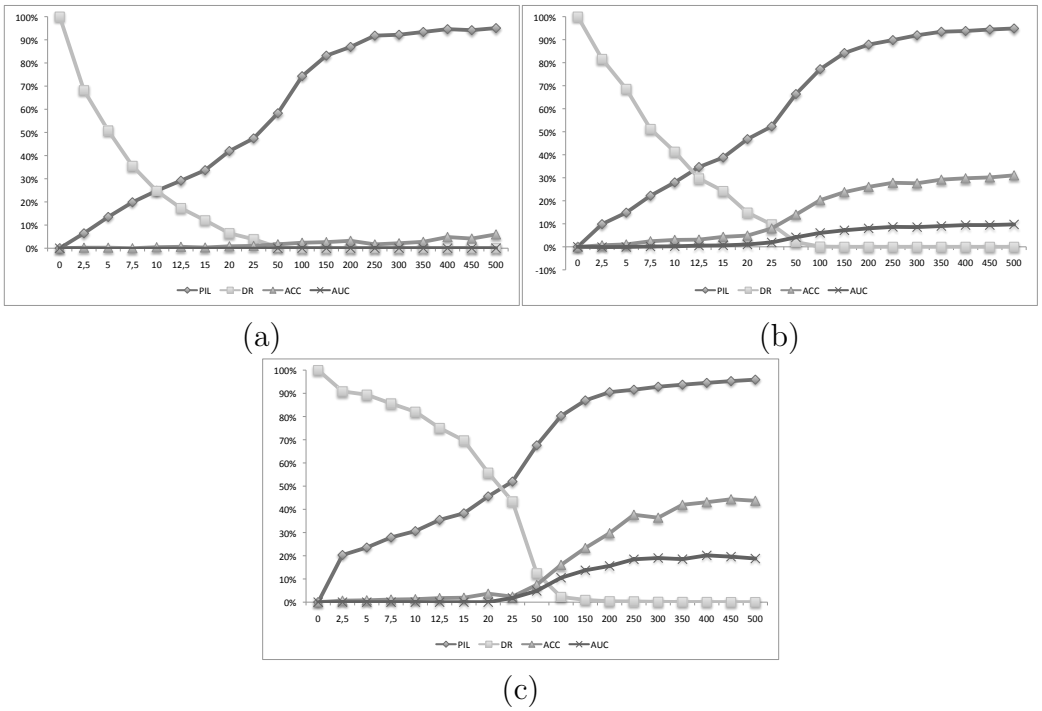

Figure 2: Graphic representation of the results obtained using the multiplicative noise protection method on the (a) Abalone, (b) Census and (c) Vehicle datasets.

start to get worse. But these extreme levels of protection are rarely used by statistical agencies, because the information loss that they produce is too high, as well.

Let us illustrate this with a specific example. Consider the rank swapping p-distribution protection method, when applied to the Abalone dataset (Figure 3 (a)). For $p = 12$, the disclosure risk (DR) is already under 10%, which means that releasing the protected dataset X_1' is quite safe. While this protected dataset leads to a (statistical) information loss (PIL) around 40%, with respect to the original data, the results obtained on classification are almost the same as when classifying using the original data. The classification results start to be bad only when the level of protection reaches $p = 40$ or $p = 50$, which are very extreme cases that will never be implemented in practice.

The results for other combinations of dataset and SDC method are very similar, although there are of course some differences. Regarding the datasets, the Abalone dataset seems to be the one which gives the best results, followed

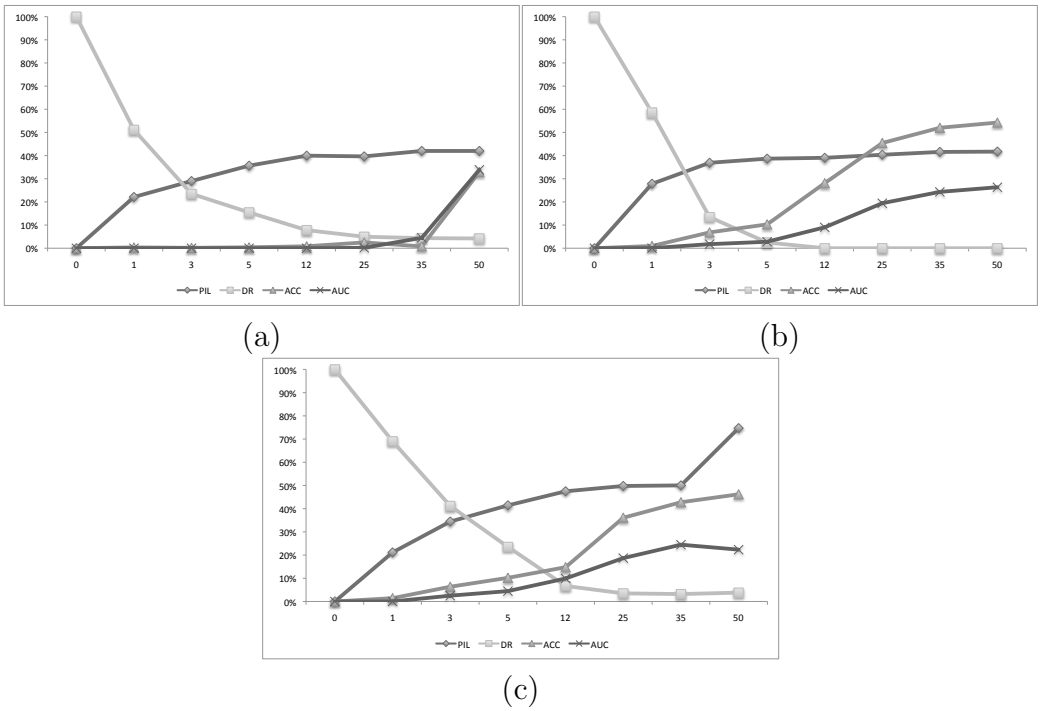

Figure 3: Graphic representation of the results obtained using the rank swapping p-distribution protection method on the (a) Abalone, (b) Census and (c) Vehicle datasets.

by Census and then Vehicle. This can be due to the fact that classification results for the original Abalone dataset are not very good (ACC around 53% and AUC around 72%), and so it is "not difficult" to maintain these results when protection is introduced to the data. Therefore, even if the figures corresponding to Census and Vehicle may seem to give a bit worse results (in terms of the slope of the ACC and AUC curves), one has to notice that the values ACC(0) and AUC(0) are higher in these two cases, and so the slow degradation of ACC(x) and AUC(x) is maybe more noticeable therein.

Regarding the different SDC methods, some differences can be found among the obtained results. The two methods with noise addition (either additive or multiplicative) seem to lead to the best classification results in general, followed by microaggregation (CBFS), rank swapping and IPSO. Some instances of the IPSO protection method lead to very good classification results, as well. Note that the values for the quality of classification are not always correlated with the values of PIL. Therefore, depending on the desired level of protection, an entity releasing a protected dataset may decide

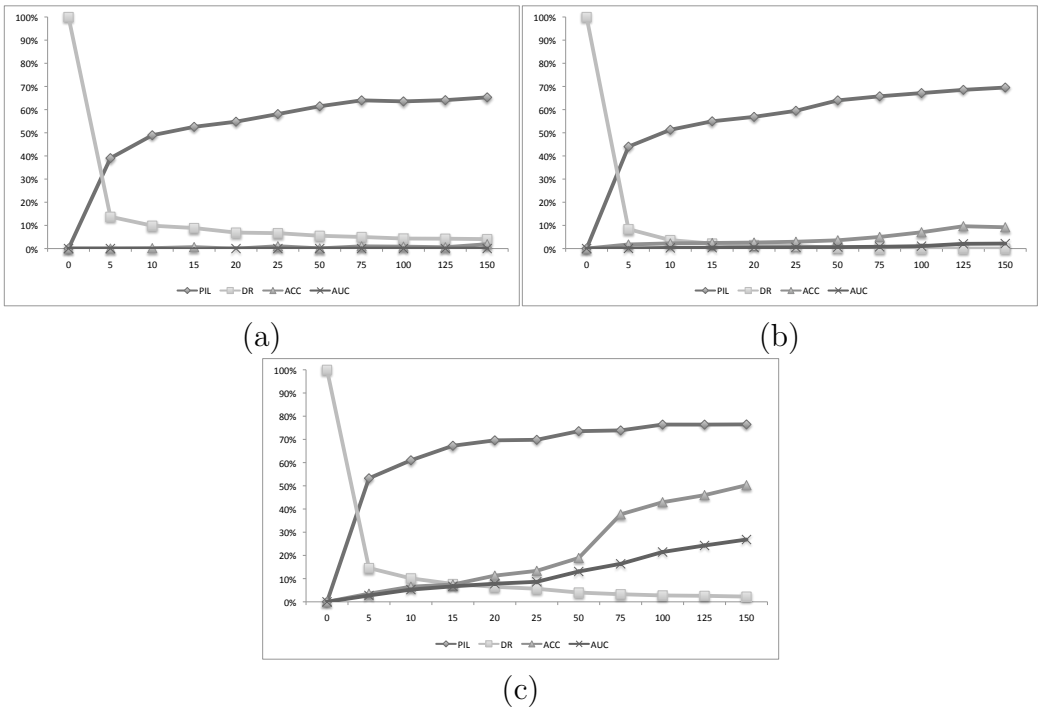

Figure 4: Graphic representation of the results obtained using the CBFS microaggregation protection method on the (a) Abalone, (b) Census and (c) Vehicle datasets.

which SDC method to apply by taking into account the most plausible use of the data that analysts will do. If the entity suspects that analysts will compute statistics on the existing data, then some methods (with an appropriate parameterization) that lead to lower values of PIL will be fine. If the entity suspects that analysts will be interested mostly in predicting values for future instances, then other methods or parameterizations that achieve very good classification results can be a better choice. Anyway, we want to stress that all the SDC methods considered here are surprisingly robust with respect to classification.

Finally, the results for microaggregation (CBFS) achieving 3-sensitivity or 3-diversity deserve some comments. These are the only considered SDC methods that lead to quite bad classification results, already at weak levels of protection. This is not surprising at all, because the p-sensitivity and l-diversity requirements are in general so strong that final clusters resulting from microaggregation contain very distant records. Of course, this effect is more evident for small values of k, that is, for small clusters: the initial clus-

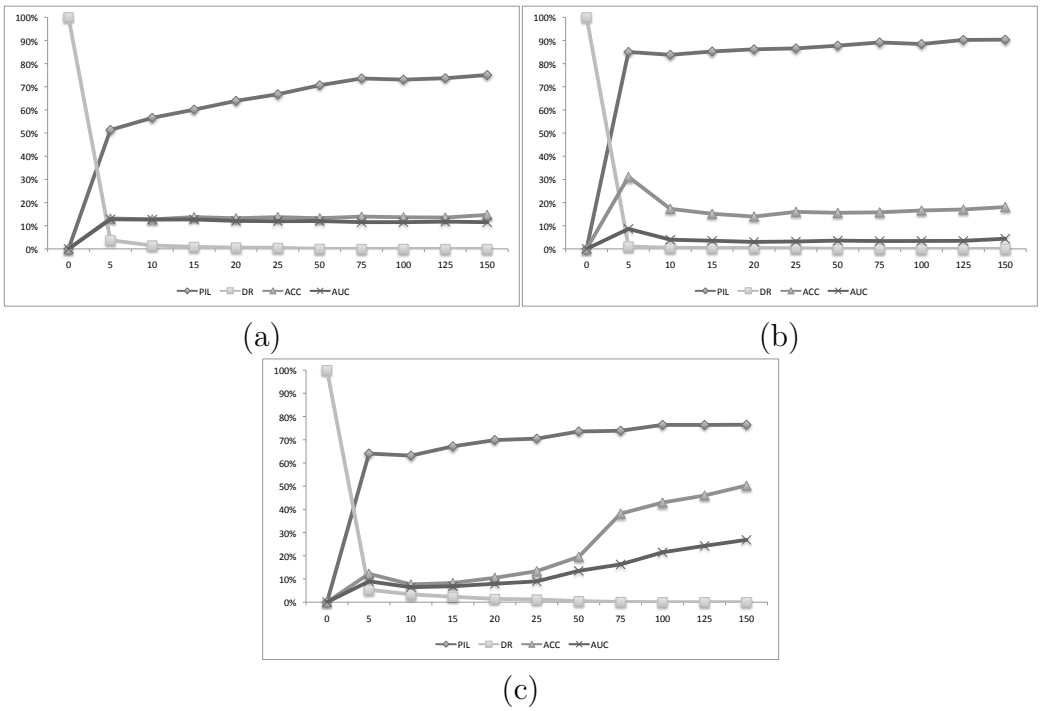

Figure 5: Graphic representation of the results obtained using the CBFS 3-sensitive protection method on the (a) Abalone, (b) Census and (c) Vehicle datasets.

ters resulting from CBFS for $k = 2, 3, 4$ are not likely to achieve 3-sensitivity or 3-diversity, and so they have to be modified. This means that the information loss (PIL) increases a lot; it is easy to check the difference PIL values between standard CBFS and CBFS achieving some of these additional properties. As our experiments show, the poor quality of the resulting clusters also affect the quality of the classification tasks that can be performed on the protected datasets. Still, the degradation of the classification results in these cases is smaller than the degradation of PIL.

4.4.1. Positive News

The surprising results that we have obtained with our experiments are good news, for both the statistical and the classification communities:

- From the statistical point of view, statistical agencies usually choose the protection parameters for the SDC methods in a specific area: where both the information loss (PIL) and the disclosure risk (DR) are under

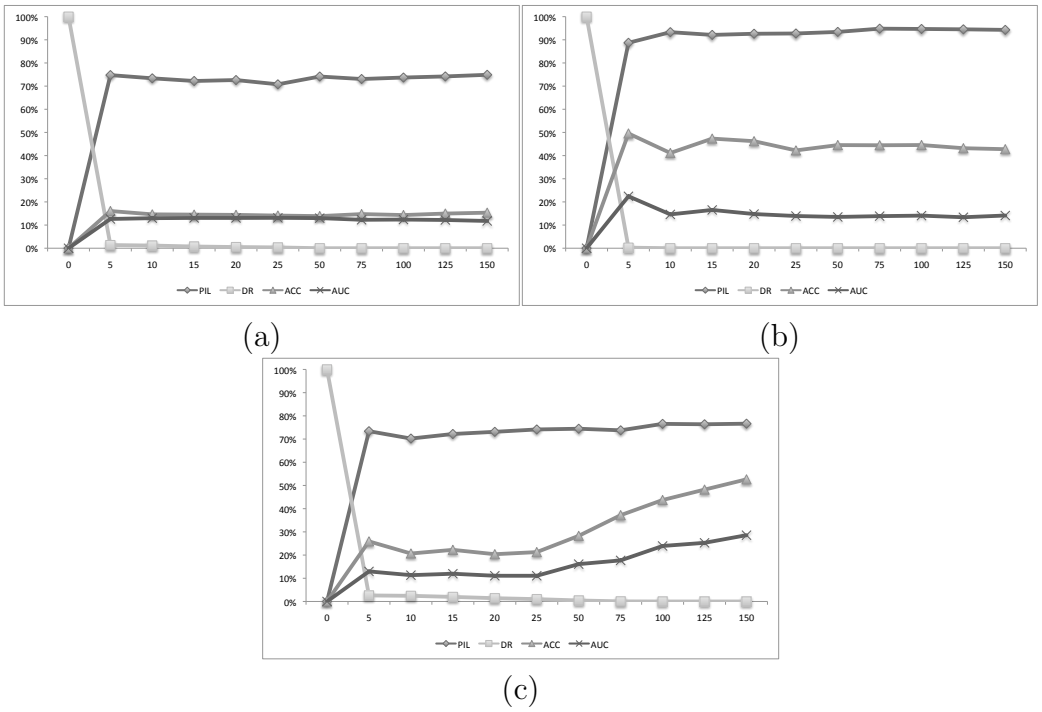

Figure 6: Graphic representation of the results obtained using the CBFS 3-diversity protection method on the (a) Abalone, (b) Census and (c) Vehicle datasets.

30%, or where the disclosure risk is under 10% (in case privacy is a more critical concern). Our experiments show that in this area of protection the quality of the classifiers is very good, almost as good as classifying with non-protected data. Therefore, when a statistical agency releases the protected dataset to the public, it can announce that this (secured) data is useful not only for statistical analysis, but also for classification purposes.

- Imagine a longitudinal study in which a data owner periodically releases the data in order to allow other users to do classification/prediction tasks on it. At some point, this owner realizes that his data contains sensitive information, and starts to worry about privacy. Supported by the results of our experiments, at some point in time he could take a SDC protection method (which is easy to use, to implement, to understand, and it is freely available), with a very high level of protection (*e.g.* noise addition with $\alpha \approx 50$ or rank swapping p-distribution with

$p \approx 10$), and apply it to his data before making it public. In this way, he will be sure that the future classification tasks will be almost as successful as the previous ones (from non-protected data), whereas now the privacy of the original data is preserved.

4.4.2. Preventive News

Although the results of our experiments are mainly good news, they can be interpreted in a different way. Remember that in the typical SDC scenario, the protection method is applied to some attributes only, whereas other (non-identifier) attributes are published in their original form, without being altered. Now imagine that one of these non-altered attributes contain sensitive information, and furthermore it allows only a few values, with a skewed distribution. For example, we can think of a binary attribute telling if a patient (record) suffers from a specific illness or not. This attribute can be thought as the class attribute of our experiments.

If there is a strong correlation between the protected attributes and this attribute, then a classifier built from the publicly released protected dataset X' will be very successful in guessing the value of the class attribute not only for future records (out of X) but also for records that are inside X. This means that an attacker who knows some quasi-identifier attributes of some patient (his age, height, weight, data of entrance to the hospital...) can use the public dataset X' to guess with high probability if this patient suffers from the illness. This would be a serious privacy breach of the whole system.

Note that this problem is related to the well-known trade-off between data mining and privacy (see [18], for example). The solution is to consider each situation separately, because the solution will depend on the attributes that are publicly released, on the attributes that are known only by some agents, on the attributes that are considered to be sensitive, etc. In any case, data owners in general (and statistical agencies in particular) should be very careful when releasing some datasets to the public, specially if they contain sensitive attributes with a skewed distribution and only a few possible values. Before doing so, they should analyze the behavior of a classifier that an attacker could build from the published data, in order to know what is the risk that such attackers successfully damage the privacy of the system. Put in different words, the quality of these classifiers could be considered as another way to measure the risk of a SDC protection method, as it is usually done with linkage disclosure and interval disclosure risks (see Section 2.1).

Of course, this potential privacy problem does not avoid the application

of the positive conclusions of our work to other scenarios. For example, a hospital A that trusts a different hospital B can privately send (*e.g.* using cryptography) a protected dataset X' containing sensitive information about patients in A, in such a way that analysts of hospital B can do data mining, for example to predict the probability that some patients in hospital B suffer from some illness. In this example, if access to this information is securely restricted to the trusted agents (scientists of hospitals A and B), then intruders will never be able to build any classifier to obtain confidential information about patients in hospitals A or B.

4.5. Comparison with a Specific PPDM Protection Method

Our experiments show that the considered SDC protection methods give quite good classification results. However, in the privacy-preserving data mining (PPDM) literature, one can easily find papers proposing specific protection methods which are designed with the clear (and unique) goal of obtaining good data mining (in particular, classification) results. We want to compare the results obtained by generic SDC methods with the results obtained by such a specific PPDM method. We have chosen the rotation perturbation method [6], that was designed with the goal of achieving very good results for k-NN and SVM classifiers.

This method works as follows. The dataset X to be protected must be thought as a matrix with n rows (records) and d columns (attributes). Firstly, X is normalized into the $[0, 1]$ interval. A random orthonormal $d \times d$ matrix $R = \{r_{ij}\}_{1 \leq i,j \leq d}$ is generated; since it is orthonormal, it satisfies $R^T R = Id$. Then, the protected dataset (or matrix) X' is obtained as $X' = X \cdot R$. Note that different orderings of the attributes lead to different results and privacy levels. A local optimization through attribute-swapping is applied to find the best attribute ordering. The whole process is repeated a number v of times, selecting at the end the best rotation matrix R and attribute ordering. Here the quality is measured with respect to some predefined privacy metric or parameter. In the simpler case (that we have considered in our experiments) where all the attributes are assumed to be equally important from the privacy point of view, the privacy parameter is limited to a real value p. Experimental results in [6] show that $v = 50$ repetitions of the process are enough to obtain the best R and attribute ordering; we have thus used $v = 50$ in our experiments.

Table 5 contains the results concerning this rotation protection method. Among the classifiers used in the previous experiments, k-NN and SVM are

				Accuracy, ACC				Area Under Curve, AUC			
		PIL	DR	DT	NB	k-NN	SVM	DT	NB	k-NN	SVM
Abalone	Rotation, $p = 0.05$	86.12%	9.15%	54.95%	52.94%	52.94%	54.05%	63.56%	64.47%	63.18%	63.05%
	Rotation, $p = 0.10$	87.46%	8.59%	54.70%	52.17%	52.10%	54.06%	64.49%	64.08%	62.55%	62.77%
	Rotation, $p = 0.15$	87.23%	8.15%	53.83%	52.52%	52.23%	54.04%	63.17%	64.56%	62.71%	63.03%
	Rotation, $p = 0.20$	89.43%	7.93%	52.77%	52.80%	52.02%	54.07%	63.21%	64.02%	62.16%	62.37%
Census	Rotation, $p = 0.05$	92.79%	4.32%	84.60%	63.42%	83.76%	84.83%	92.91%	94.45%	95.00%	95.99%
	Rotation, $p = 0.10$	92.15%	4.27%	81.23%	65.14%	83.82%	85.28%	92.01%	94.00%	94.93%	95.05%
	Rotation, $p = 0.15$	95.63%	3.15%	81.66%	63.90%	82.23%	84.95%	91.21%	93.79%	94.53%	95.83%
	Rotation, $p = 0.20$	96.38%	2.21%	81.08%	67.79%	82.27%	81.08%	92.12%	93.47%	94.33%	94.17%
Vehicle	Rotation, $p = 0.05$	91.88%	5.67%	62.67%	42.94%	72.02%	73.75%	72.14%	71.98%	83.09%	78.24%
	Rotation, $p = 0.10$	91.63%	5.90%	63.33%	42.39%	70.04%	73.23%	72.28%	71.36%	82.01%	77.66%
	Rotation, $p = 0.15$	92.15%	6.33%	66.18%	41.36%	70.53%	72.88%	72.25%	71.24%	80.96%	77.09%
	Rotation, $p = 0.20$	92.37%	6.55%	62.86%	41.37%	70.12%	72.39%	69.01%	71.31%	80.42%	76.31%

Table 5: Results obtained by the rotation PPDM method.

rotation-invariant, while NB and DT are not. Thus, it is not surprising that the best classification results obtained by rotation are those concerning these two classifiers k-NN and SVM. However, these results are not significantly better (maybe excepting the results for the dataset Vehicle) than the results obtained by some parameterizations of generic SDC methods. The rotation protection technique achieves quite good privacy results. However, the main drawback of this method is that it leads to very poor PIL results. In other words, releasing a dataset X' that has been produced from X through rotation is useful only for clients who are interested in performing classification tasks, and in particular k-NN or SVM. If a client is interested in computing other kind of (statistical) information on X, then X' will not help him at all; a different protected version of X should be released, in this case.

In contrast, using a generic SDC protection method leads to a perturbed dataset X' which can be used for both tasks: inducing a classifier from X and obtaining statistical information on X. Therefore, we believe that the approach considered in this work is more useful and replicable than the use of very specific PPDM protection methods.

5. Conclusions and Future Work

In this paper we have experimentally shown that well-known SDC protection methods behave very well when used for classification purposes. In other words, the classifiers that are built from a protected dataset are essentially as good as those built from the original dataset, even if the protection level is reasonably high.

In some way, our results may lead people to question the necessity of certain papers that have appeared in the privacy-preserving data mining (PPDM) literature, proposing specific protection methods with the clear goal of providing good classification. If general-purpose SDC methods already provide a good level of privacy and very good classification, results, then what else is needed?

However, our work is not closed at all. For example, our experiments consider numerical databases only; there are SDC methods which work with categorical attributes, so it will be interesting to see if similar results can be obtained in that case. As well, we have concentrated in this work on the possible use of generically protected data for learning a classifier. Other specific data mining tasks (regression, association rule learning, etc.) could be considered as well, in this scenario, to see if the good results that we have got here for classification are also obtained for other tasks.

In the SDC community, when different protection methods are compared or ranked, one usually considers the *score* measure [12], which is simply the average between PIL and DR. Since the conclusion of our work is that SDC protected datasets can be used also for other (non-statistical) tasks such as classification, we believe that other measures such as ACC, AUC or measures related to other mining tasks should be considered as well when analyzing the quality of SDC protection methods.

Acknowledgements

Partial support by the Spanish MEC (projects ARES – CONSOLIDER INGENIO 2010 CSD2007-00004 – and eAEGIS – TSI2007-65406-C03-02) is acknowledged. Javier Herranz enjoys a *Ramón y Cajal* grant, partially funded by the European Social Fund (ESF), from Spanish MICINN Ministry. Jordi Nin is supported by the European Community through the 7th Framework Programme Marie Curie Intra-European fellowship, contract No 235226.

Bibliography

[1] C. C. Aggarwal and P. S. Yu, editors. *Privacy-Preserving Data Mining: Models and Algorithms*, volume 34 of *Advances in Database Systems*. Springer, 2008.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 439–450, 2000.

[3] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[4] R. Brand. Microdataprotection through noise addition. In *Inference Control in Statistical Databases*, volume 2316 of *Lecture Notes in Computer Science*, pages 97–116, 2002.

[5] J. Burridge. Information preserving statistical obfuscation. *Statistics and Computing*, 13:321–327, 2003.

[6] K. Chen and L. Liu. A random rotation perturbation approach to privacy preserving data classification. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, pages 589–592, 2005.

[7] T. Dalenius and S. Reiss. Data-swapping: a technique for disclosure control. *Journal of Statistical Planning and Inference*, 6:73–85, 1982.

[8] D. Defays and M. Anwar. Micro-aggregation: a generic method. In *Proceedings of the 2nd International Seminar on Statistical Confidentiality*, pages 69–78, 1995.

[9] J. Domingo-Ferrer. *Inference Control in Statistical Databases, From Theory to Practice*. Springer, 2002.

[10] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Trans. on Knowledge and Data Engineering*, 14(1):189–201, 2002.

[11] J. Domingo-Ferrer, F. Sebé, and A. Solanas. An anonymity model achievable via microaggregation. In *VLDB Workshop: Secure Data Management*, volume 5159 of *Lecture Notes in Computer Science*, pages 209–218, 2008.

[12] J. Domingo-Ferrer and V. Torra. *Disclosure control methods and information loss for microdata*, pages 91–110. Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, 2001.

[13] J. Domingo-Ferrer and V. Torra. *A quantitative comparison of disclosure control methods for microdata*, pages 111–133. Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, 2001.

[14] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2):103–130, 1997.

[15] W. Du and Z. Zhan. Building decision tree classifier on private data. In *Proc. of the IEEE Int. Conf. on Privacy, security and data mining*, pages 1–8, 2002.

[16] T. Fawcett. Roc graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs, 2003.

[17] A. Friedman, A. Schuster, and R. Wolff. Anonymous decision tree induction. In *Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 151–162, 2006.

[18] M. Kantarciog, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 599–604, 2004.

[19] J. Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *Proceedings of the ASA Section on Survey Research Methodology*, pages 303–308, 1986.

[20] J. Kim and W. E. Winkler. Multiplicative noise for masking continuous data. Research report series (statistics 2003-01), U. S. Bureau of the Census, 2003.

[21] J. Lane, P. Heus, and T. Mulcahy. Data access in a cyber world: making use of cyberinfrastructure. *Transactions on Data Privacy*, 1(1):2–16, 2008.

[22] M. Laszlo and S. Mukherjee. Minimum spanning tree partitioning algorithm for microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, 17(7):902–911, 2005.

[23] N. Li and T. Li. t-closeness: Privacy beyond k-anonymity and -diversity. In *Proc. of IEEE Int. Conf. on Data Engineering*, 2007.

[24] K. Liu, H. Kargupta, and J. Ryan. Random projection based multiplicative data perturbation for privacy preserving data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):92–106, 2006.

[25] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *IEEE Int. Conf. on Data Engineering*, 2006.

[26] J. M. Mateo-Sanz, J. Domingo-Ferrer, and F. Sebé. Probabilistic information loss measures in confidentiality protection of continuous microdata. *Data Mining and Knowledge Discovery*, 11(2):181–193, 2005.

[27] U.S. Census Bureau. Data extraction system. http://www.census.gov/, 2009.

[28] K. Muralidhar and R. Sarathy. Generating sufficiency-based non-synthetic perturbed data. *Transactions on Data Privacy*, 1(1):17–33, 2008.

[29] P. Murphy and D. Aha. UCI Repository machine learning databases. *Irvine, CA: University of California, Department of Information and Computer Science*, 1994.

[30] J. Nin, J. Herranz, and V. Torra. Rethinking rank swapping to decrease disclosure risk. *Data and Knowledge Engineering*, 64(1):346–364, 2008.

[31] A. Oganian and J. Domingo-Ferrer. On the complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal United Nations Economic Commission for Europe*, 18(4):345–354, 2000.

[32] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[33] P. Samatari and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI Intl. Tech. Rep., 1998.

[34] A. Solanas, F. Sebé, and J. Domingo-Ferrer. Micro-aggregation-based heuristics for p-sensitive k-anonymity: one step beyond. In *Int. Workshop on Privacy and Anonymity in Information Society*, pages 61–69, 2008.

[35] N. Spruill. The confidentiality and analytical usefulness of masked business microdata. In *Proceedings of the ASA Section on Survey Research Methodology*, pages 602–610, 1983.

[36] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[37] V. Torra. *Handbook of Data Mining*, chapter Privacy in Data Mining. Human Factor and Ergonomics, 2009.

[38] T. M. Truta and B. Vinay. Privacy protection: p-sensitive k-anonymity property. In *IEEE Int. Conf. on Data Engineering Workshops*, 2006.

[39] J. Vaidya, C. Clifton, and M. Zhu. *Privacy Preserving Data Mining*. Springer, 2006.

[40] J. Vaidya, M. Kantarcioglu, and C. Clifton. Privacy-preserving naïve bayes classification. *The Very Large Database Journal*, 17(4):879–898, 2008.

[41] V. Vapnik. The support vector method. In *Int. Conference on Artificial Neural Networks*, pages 263–271, 1997.

[42] L. Willenborg and T. de Waal. *Elements of Statistical Diclosure Control*. Lecture Notes in Statistics. Springer, 2001.

[43] W. Winkler. Re-identification methods for masked microdata. In *Proceedings of Privacy in Statistical Databases*, volume 3050 of *Lecture Notes in Computer Science*, pages 216–230. Springer, 2004.

[44] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.