Utility of Knowledge Discovered from Sanitized Data

Michal Sramka, Reihaneh Safavi-Naini, Jörg Denzinger, Mina Askari and Jie Gao
Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary AB T2N 1N4, Canada

{msramka,rei,denzinge,maskari,gaoj}@cpsc.ucalgary.ca

Abstract

While much attention has been paid to data sanitization methods with the aim of protecting users' privacy, far less emphasis has been put to the usefulness of the sanitized data from the view point of knowledge discovery systems. We consider this question and ask whether sanitized data can be used to obtain knowledge that is not defined at the time of the sanitization. We propose a utility function for knowledge discovery algorithms, which quantifies the value of the knowledge from a perspective of users of the knowledge. We then use this utility function to evaluate the usefulness of the extracted knowledge when knowledge building is performed over the original data, and compare it to the case when knowledge building is performed over the sanitized data. Our experiments use an existing cooperative learning model of knowledge discovery and medical data, anonymized and perturbed using two widely known sanitization techniques, called ϵ-differential privacy and k-anonymity. Our experimental results show that although the utility of sanitized data can be drastically reduced and in some cases completely lost, there are cases where the utility can be preserved. This confirms our strategy to look at triples consisting of a utility function, a sanitization mechanism, and a knowledge discovery algorithm that are useful in practice. We categorize a few instances of such triples based on usefulness obtained from experiments over a single database of medical records. We discuss our results and show directions for future work.

Keywords: utility of knowledge, privacy-preserving data mining, differential privacy, k-anonymity, cooperative learning

1 Introduction

One of the main reasons why many organizations are interested in getting access to or doing large data collections is that these organisations want to use these collections to create new knowledge useful for them. Long before we had computers, data collections were used to, for example, get an idea what the average travel time between places was, how much food a soldier in an army consumes per day, or when fields should be planted. With computers, it became easier to do the analysis of data collections and the kind of statistical analysis that was done before could now be done with much larger data sets. And nowadays, even more complex analysis for even more complex knowledge can be done using advanced data mining techniques that try to come as close as possible to creating knowledge that is useful for many possible users of these techniques.

From the perspective of an individual, whose personal data is included in large data collections, some of the knowledge that organisations can gain from these collections will result in positive effects also for the individual, like new knowledge detecting diseases, but there can also be negative effects, like increases in car insurance premiums based on new knowledge an insurance company got out of their data collections. Even more, there is also always the possibility of abuse of personal data, like someone with access to the data informing a potential employer of an individual about his/her current financial situation which might give this potential employer advantages in the salary negotiations. So, for an individual, privacy of all or some of his/her individual data is an important concern and owners of data collections need to respect these privacy concerns [3].

At first glance, data mining and privacy seem to represent opposing goals, with mining trying to bring all kinds of knowledge "into the open" and privacy being interested in keeping such knowledge "in the dark". But at a closer look, we can identify some key conditions with regard to both areas that can make the goals of them compatible. The users of data mining want to create knowledge *useful* for them, which means that they will have some kind of idea regarding the *utility* of new knowledge created. On the other side, individuals whose data is included in data collections usually are only concerned with the privacy of some of their data and there is also quite a spectrum of definitions of what people think privacy should mean.

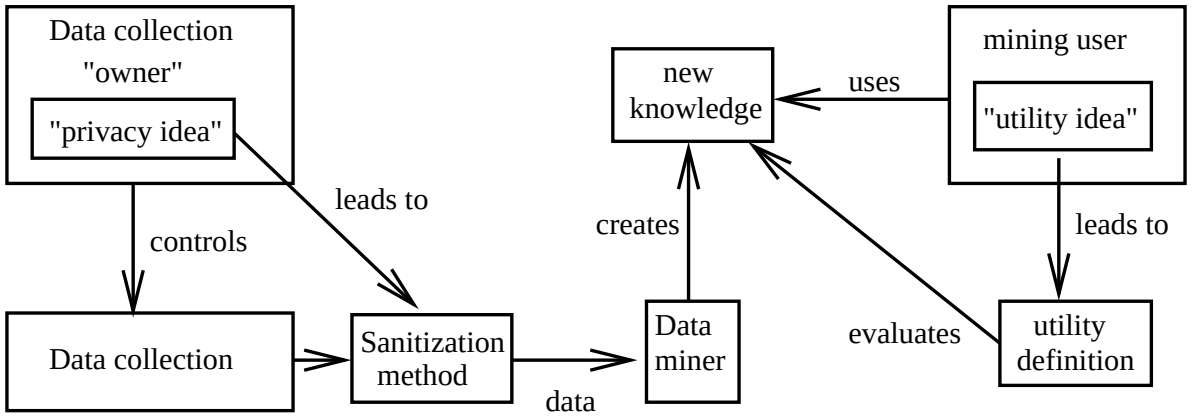


Figure 1: Data mining vs. privacy

So, users of data mining need to refine their utility idea to a precise utility definition and the people responsible for data collections need to come up with some kind of data sanitization that fulfills the privacy wishes that are put on the collection. For a given data mining mechanism, it is then possible to compare the utility that it produces on non-sanitized data with the utility achieved when mining the sanitized data and this way it is possible to identify triples (utility definition, sanitization mechanism, data miner) that are sufficiently compatible to be both useful and privacy-preserving. This general view is also graphically depicted in Figure 1. It should be noted that naturally not all such triples represent compatible instantiations of the 3 components.

In this paper, we present experimental evaluations of such triples that turn out to be not useful, but we are also able to present triples that are very useful for a rather complex mining goal, namely finding rules for the analysis of billing data by doctors and hospitals in order to suggest patients that might have diabetis.

1.1 Sanitization, Knowledge Discovery, and Utility

To mitigate the concern of users over privacy, different approaches have been proposed to transform data so that users' information is protected. Measures such as ϵ-differential privacy [4], ϵ-indistinguishability [5], k-anonymity [13] and its derivatives such as p-sensitive k-anonymity, (α, k)-anonymity, l-diversity, and t-closeness, have been proposed to quantify the level of privacy achievable in the transformed *sanitized* data. Sanitized data may be accessed in one of two models of interaction. In the first model, the sanitized version of the data (database) is published and becomes accessible by the public, while in the second model users can only access the database through a *sanitization filter* that transforms the response to a query before delivering it to the end user. Sanitized data, published or accessible through an interactive sanitization filter, is then used for statical analysis and making inferences leading to new knowledge.

Knowledge discovery (also knowledge building, knowledge learning, data mining) refers to non-trivial extraction of implicit, unknown, and potentially useful information from data [7]. A *knowledge discovery method* extracts trends or patterns from data and carefully and accurately transforms them into useful and understandable information. Such information, henceforth called *knowledge*, is more complex than what is typically retrievable by standard techniques, such as statistics, but is uncovered through the use of artificial intelligence techniques. Usefulness of the knowledge obtained at the end of the knowledge discovery process is measured by *utility* functions that quantify the level of user satisfaction with the knowledge discovered by the method.

An important and less considered aspect of data sanitization algorithms is how they preserve the usefulness of the data for knowledge discovery systems. Ideally, a sanitization algorithm must ensure that no private or sensitive information will leak about individuals, while knowledge discovery algorithms can be used to extract useful knowledge about groups and trends in data – similar to those that could be extracted by using the original data. Much attention has been paid to the methods that ensure user anonymity and protect their private and sensitive information, but far less emphasis has been put into measuring and determining usefulness of the sanitized data for knowledge discovery.

1.2 Our contribution

We propose a definition for the utility of discovered knowledge and use it to compare the utility of knowledge discovered from original and sanitized data, using two widely used methods of data sanitization. Our definition of utility captures the value of the knowledge from the perspective of the end user who will be using the knowledge, and it differs from the classical measures in data mining, such as confidence and support of association rules that are all described later. Although we present a fairly general utility function that captures the core part of the usefulness of the obtained knowledge by the end user, there may still be the need and possibility of its extension.

We use k-anonymity [13] and ϵ-differential privacy [4], to quantify the level of anonymity and protection of the sanitized data. We investigate the *usefulness* of the sanitized data, that is, we ask the question whether the sanitized data can be used to obtain knowledge not defined and not known at the time of the sanitization. We do this by using the sanitized data as input to a knowledge discovery algorithm, evaluating the utility of the discovered knowledge using our proposed notion of utility, and comparing our result with with the utility of the knowledge obtained from the same

knowledge discovery algorithm when used with the original unsanitized data as input.

Ultimately, we are trying to categorize triples (utility definition, sanitization mechanism, data miner) according to their usefulness in practice. A utility definition is often the missing dimension when evaluating sanitization together with knowledge discovery. We argue that a utility definition should be an integral part when sanitization and knowledge discovery are evaluated for privacy-preservation and usefulness in practice. We use existing sanitization mechanisms and existing knowledge discovery methods, then we add the utility definition to them and categorize them. Our categorization is based on experiments performed over a single database consisting of medical records, and one would expect to see similar results for similar databases.

For knowledge discovery, as our data miner component, we use the cooperative learning system introduced in [8]. It is a multi-agent system, which can produce knowledge that no single data miner can produce on its own. This allows us to perform more complex and versatile experiments on the original and sanitized data than just using a single data miner.

One expects that the utility of knowledge discovered from sanitized data to decline compared to the utility of knowledge discovered from unsanitized data. However, one requires that sanitized data to remain highly 'useful' – in other words the utility to remain at an acceptably high level.

k-anonymization releases sensitive information, but ensures that individuals cannot be linked to the information. Our experiments show that the loss of utility of the knowledge discovered by the cooperative learning system from k-anonymized data is no more than 40% when compared to the utility of knowledge discovered from unsanitized data. This result holds for a weak protection of $k = 2$ and strong privacy protection of $k = 100$. However, there are cases where the utility of the knowledge discovered from k-anonymized data is higher than the utility of knowledge discovered from unsanitized data. This, together with our definition of utility and the cooperative learning system, is an example of a triple that is useful in practice.

Data perturbation protects the privacy of data values. We used 0.01-differential privacy to bound the amount of noise added during perturbation. This represents a strong privacy guarantee for data. Our experiments show that the utility of the knowledge discovered by the cooperative learning system from perturbed data is declining as more quality knowledge is discovered, and in some cases, there is no utility at all. In particular, the loss of utility is 100% as more quality knowledge is mined. This matches our expectation: local patterns and trends in the perturbed data, which represent less complex "low quality" knowledge, can be still extracted from the perturbed data and represents a useful and practical triple, but global patterns and more complex relationships are masked in the perturbed data and so high quality knowledge is harder (impossible?) to discover.

These results suggest important directions for future research in data sanitization with respect to utility in knowledge discovery systems.

1.3 Related work

Privacy-preserving data mining is a field of study concerned with getting valid data mining results without learning the underlying private data. This is usually achieved [15] either by private data mining – data mining performed by the data owner; using Secure Multiparty Computation (SMC) – a cryptography-based technique; or by perturbation techniques. We will concentrate on the latter

approach, that is, we will assume that the data owner, who performs the perturbation, and data miner are two separate entities. Most of the existing proposals in this context follow the method of Agrawal and Srikant [2], which allows to approximate the original distribution of the data, hence preserving some statistical properties of the original data.

There are many papers dealing with privacy-preserving data mining [2, 6, 10, 11, 14]. Usually, the papers describe methods that start with a mining goal and propose measures that protect the privacy of the underlying data, while still producing results contributing to the mining goal. It has to be noted that all of the privacy-preserving data mining mechanisms that we know of are targeting a specific mining goal.

Data anonymization is another field that is concerned with the privacy of the underlying data. A typical example is the k-anonymity [13]. The process of anonymization takes the original data and releases them in a privacy-preserving way. The assumption is that the privacy of the original sensitive data is preserved in the anonymized release. This is usually achieved by distortion, generalization, and suppression. Again, some statistical properties of the original data are preserved, but the data anonymization is not concerned with the use of the information for knowledge discovery, only with the release of anonymized information.

1.4 Organization of the paper

We introduce our model for discovering knowledge from sanitized data in Section 2, together with showing some shortcomings of existing sanitization techniques for knowledge discovery. Section 3 presents our core results – the measure of utility for prediction systems and the definition of a useful and privacy-preserving triple (utility definition, sanitization mechanism, data miner). Experimental results are in Section 4.

2 Data Sanitization and Knowledge Discovery

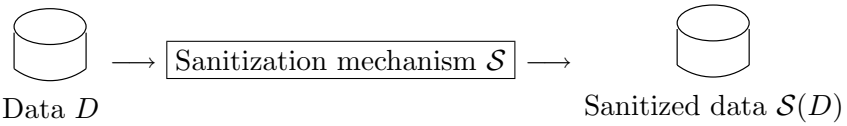
A model for knowledge discovery over sanitized data is introduced. The value of sanitized data for knowledge discovery is often overlooked, and although some statistical properties may be preserved, knowledge discovery goes, in general, beyond statistical characteristics of the data.

2.1 A model for knowledge discovery

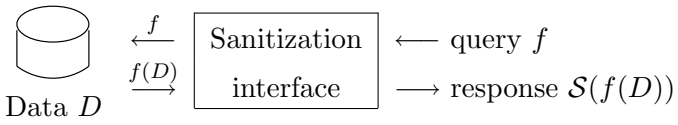
Assume that D is a data set which contains some sensitive data. The sensitive data, denoted D^*, can be elements of D or just parts of some of the elements in D.

A *sanitization* (also called anonymization, pseudonymization, de-identification, obscuration, redaction) is a privacy-preserving process that transforms and releases the data D, while trying to preserve the privacy of the sensitive data D^*. It is achieved by a combination of value removing, swapping, shuffling, substituting, masking, obscuration, perturbation, randomization, sub-sampling, or other techniques [1]. *Anonymization* is a form of sanitization, which protects the identity of individuals in the data, and *perturbation* is another form of sanitization, which protects sensitive and private data, not just the identity of individuals. A *sanitization mechanism* $\mathcal{S}$ is an

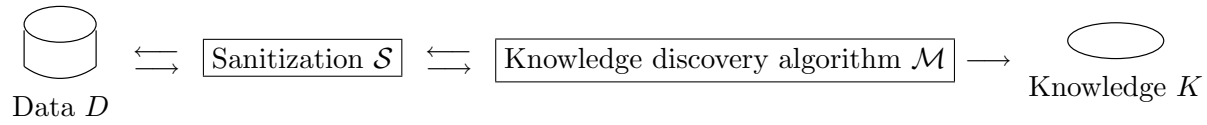
algorithm that on input x, an element in the range of any (query) function defined over D, uses the sanitization process described above to output sanitized x that preserves the privacy of D^*. The process of sanitization can work in either one of the two models of interaction: interactive model and non-interactive model [4]. In the non-interactive setting the original data is sanitized and the output is published. An example of a non-interactive sanitization mechanism is the k-anonymity [13].



In the interactive setting the data is made available to parties through an interface that for incoming queries provides (possibly adaptively chosen) sanitized answers. An example of an interactive sanitization mechanism is the ϵ-differential privacy [4].



The sanitized output coming from an interactive or non-interactive sanitization mechanism is used as an input to knowledge discovery algorithms. A *knowledge discovery algorithm* takes a data set as an input and outputs another data set representing the discovered knowledge in the input. We take an existing knowledge discovery algorithm $\mathcal{M}$ proposed to work over D, and run it unmodified over the sanitized input $\mathcal{S}(D)$.



Our goal is to introduce a measure that quantifies utility of the knowledge discovered by $\mathcal{M}$ when used over D and over $\mathcal{S}(D)$.

2.2 No value of the released sanitized data

The notion of ϵ-differential privacy has been proposed to limit the leakage of private information in the released data. On an example that achieves ϵ-differential privacy, we show that the usefulness of the released (sanitized) data for knowledge discovery can be non-existent.

The definition of the ϵ-differential privacy notion [4] follows: A randomized function $\mathcal{S}$ gives *ϵ-differential privacy* if for all data sets D_1 and D_2 differing on at most one element, and all $R \subseteq Range(\mathcal{S})$, $\Pr[\mathcal{S}(D_1) \in R] \leq \exp(\epsilon) \times \Pr[\mathcal{S}(D_2) \in R]$. $\exp(x)$ is the exponential function e^x.

The ϵ-differential privacy notion limits the probability that the randomized function $\mathcal{S}$, a sanitization function, would leak information from a data set that is extended by at most one element. That is, a data leak and therefore a disclosure of a private data through the function $\mathcal{S}$ is possible, but limited by the leakage parameter ϵ.

Consider a randomized function $\mathcal{S}$ that on every input always outputs the same constant. Such a function satisfies the ϵ-differential privacy notion, as well as the similar ϵ-indistinguishability notion [5], for every leakage $\epsilon \geq 0$. Yet the output of this function $\mathcal{S}$, a sanitized release of data, is useless for practical purposes, such as statistical and knowledge discovery.

This discussion does not imply that the ϵ-differential and ϵ-indistinguishability privacy notions are not good. In literature, the ϵ-differential privacy and ϵ-indistinguishability are achieved [5, 4] by using the *value distortion* technique. That is, an original value x is released as $x + r$, where r is a random value drawn from a known distribution. This discussion merely points out that methods that release sanitized data should take into account the usefulness of such data for statistical and/or knowledge discovery purposes.

The "utility of the data at the end of the privacy-preserving process", what we refer to as the usefulness of the released sanitized data, has been identified [15] as one of the quality criteria when evaluating privacy-preserving methods, but too often it is not even mentioned when privacy-preserving notions, algorithms, methods, and techniques are proposed and evaluated. We believe that sanitization processes should take into account the usefulness of the released data, and a guarantee on some degree of usefulness should be an integral part of privacy notions.

2.3 Shortcomings of statistical characteristics for knowledge discovery

Existing sanitization methods, such as anonymization and perturbation, may degrade properties and characteristics of the original data. In the following, we point out some shortcomings of popular perturbation and anonymization techniques when their output is used for knowledge discovery.

Take for example the ϵ-differential privacy [4] or the perturbation method of Agrawal and Srikant [2]. They both propose to use the value distortion technique to achieve perturbation. The additive noise is randomly chosen from a Laplace distribution in the case of ϵ-differential privacy and from a Gaussian distribution in the case of the method of Agrawal and Srikant.

After the perturbed data is released, an approximation of the probability distribution of the original data can be computed using the Expectation Maximization method [2]. In other words, a statistical property of the original data is approximated. Having a probability distribution of the data is sufficient to obtain a decision-tree classifier, but not sufficient for most of the other knowledge building methods. For example, the cooperative learning system [8], that we use for our experiments, looks (among others) for ordered sequences. However, perturbation hides the original values and having statistical properties of the values is not enough to retain the sequence information.

k-anonymity [13] is a data anonymization model for protecting privacy of the individuals. A table consists of identifiable columns (names, social security numbers, etc.), quasi-identifiable columns (combination of which can still identify a person, e.g., date of birth, ZIP code, gender, etc.), and sensitive columns (data that is private to the individual). The k-anonymity model assures that the identifiable columns are removed, sensitive columns are untouched, and any combination of quasi-identifiable values from one row can be found in at least $k-1$ other rows of the table. This means that if a combination of the quasi-identifiable values from one row of the table identifies an individual, then the individual can be confused with at least $k-1$ other individuals in the table.

The k-anonymity model preserves the sensitive information, and hence all its statistical characteristics, but many relations among sensitive values and quasi-identifiable original values are lost. This is a problem for a knowledge discovery method that relies on true original relations.

3 Utility of Prediction Systems

Knowledge is an abstract concept and it can be represented in different ways. In a majority of the cases, the knowledge can be represented or converted to "if ... then ..." rule form. Here, we look at the specific case of rule mining, where knowledge is represented as a *rule* of the form "conditions $\Rightarrow$ conclusion". We assume that our knowledge discovery algorithm $\mathcal{M}$ produces rules of the above form that are used to predict values of specific fields in a data base.

Concentrating on rule mining and prediction systems is not really a restriction, since one can show that rule-based systems are Turing-complete as indicated by programming languages like Ops5 or Prolog. A prediction system can be also used for classification of data into classes, simply by using the prediction system to predict a class.

An association rule is $X \Rightarrow Y$, where X and Y are sets of items. Its meaning is that a transaction – a set of items – that contains the items in X tend to also contain the items in Y. An example is 91% of students who pass Calculus 1 also pass Calculus 2. 91% is called the *confidence* of the rule. The *support* of the rule $X \Rightarrow Y$ is the percentage of transactions in a set of transactions (a database) that contain both X and Y. Confidence and support are the traditional measures of quality of the association rules [6].

We consider a more general form of the rules, not just association rules, in the data. We assume our *rules* are of the form $X \Rightarrow Y$, where X is a boolean expression representing the condition, and Y is a prediction. For example, our rule can be

$$age \geq 50 \ \wedge \ \text{seq}(d136, d451, d94) \ \wedge \ \text{has}(d215) = \text{true} \ \Rightarrow \ \text{diabetes},$$

that is, if a patient is of age 50 or older, has been diagnosed with $d136$, $d451$, and $d94$ (representing diagnostic codes) in this order, and has been diagnosed with $d215$ at any time, than the patient is predicted to have diabetes. In other words, our condition X is a combination of logical expressions that may operate over sequential data. We call such rules *hybrid* rules, because they are usually a combination of different types of expressions [8].

Finally, we want to be able to measure the utility of such rule for the user who will be using this rule. In fact, we want to measure the utility not just of the obtained rules, but of the whole prediction system from the perspective of the end user. Therefore, our utility measure goes beyond the confidence and support measures. For example, a user wants to obtain the information about which patients are susceptible to diabetes. The user does not care whether there will be a large amount of false positives, as long as all the true positives would be detected. This cannot be captured using confidence and support.

3.1 Utility

Let $\mathcal{U} = D_1 \times \cdots \times D_n$ be a universe of tuples, where the D_i's denote domains. $(x_1, \ldots, x_n) \in \mathcal{U}$ is called a *tuple* and each x_i is the value of a *field*. Let DB be a database, a finite subset of $\mathcal{U}$. This corresponds to the relational database concept.

There is a *data mining algorithm* DM that takes a database DB and outputs a set of rules $R = \{X_1 \Rightarrow Y_1, \ldots, X_m \Rightarrow Y_m\}$ that can be used for prediction. Let the set I contain the indices of the fields to be predicted. An *incomplete tuple* is $x = (x_i) \in \mathcal{U}^\perp$, where $\mathcal{U}^\perp = D'_1 \times \cdots \times D'_n$ with $D'_i = \{\perp\}$ for $i \in I$, and $D'_i = D_i$ for other i's. The symbol $\perp$ represents a missing value. The rules in R capture the patterns and trends in the database DB and are used to *predict* the missing field values of incomplete tuples.

A rule $X \Rightarrow Y \in R$ is *applicable* to an incomplete tuple $x \in \mathcal{U}^\perp$, if the boolean expression X evaluates as true when the field values of x are assigned to the corresponding variables of X. A *prediction algorithm* PA takes the set of rules R and an incomplete tuple $x = (x_i) \in \mathcal{U}^\perp$, and for each rule $X_j \Rightarrow Y_j \in R$ determines whether the rule is applicable to x. If it is, then the tuple $(x_i) \in \mathcal{U}$, with the x_i's, for $i \in I$, coming from the prediction Y_j, is added to the output set V. The tuples in the output set V, $|V| \leq m$, may be conflicting on the predicted values, because several rules in R may be applicable to x.

A *conflict resolution algorithm* CR takes the set $V \subseteq \mathcal{U}$, decides on the best estimate for the missing fields indexed by I, and outputs this tuple $v \in V$. The predicted fields of v are $\hat{x} = (\hat{x}_i)_{i \in I}$, and we assume the true values of these fields are $\bar{x} = (\bar{x}_i)_{i \in I}$.

The set of algorithms DM, PA, and CR (together with the index set I that depends on DM) comprise a *prediction system*, a type of *knowledge discovery system*.

We define a *error implication function* $E(\hat{x}, \bar{x})$ that weights the seriousness of any occurring error for the utility, e.g., $E(\hat{x}, \bar{x})$ can be some value based on the predicted $\hat{x}$ and true $\bar{x}$ that accounts for the preferences of a user who measures the utility. Note that we want the function E to have a high value if there is no error, while it should be near 0 for serious errors. The function E can be seen as $E(\hat{x}, \bar{x}) = \sum_{i \in I} E_i(\hat{x}_i, \bar{x}_i)$, a sum of error implication functions for each of the predicted fields.

In addition, the user is interested in incomplete tuples with a certain interest factor, therefore the user decides on a weight $w(x)$ for each $x \in \mathcal{U}^\perp$, where higher weights for tuples show that the user is more interested in those tuples.

The user computes the utility of the prediction system, resp. the correctness part of the utility, by finding

$$\begin{aligned} \mathtt{Utility}(\mathsf{DM}, \mathsf{PA}, \mathsf{CR}, DB) &= \sum_{x \in \mathcal{U}^\perp} w(x) E(\hat{x}, \bar{x}) \\ &= \sum_{x \in \mathcal{U}^\perp} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(DB), x)), \bar{x}). \end{aligned} \tag{1}$$

We note that the utility as we have just defined it is just some kind of core utility (that deals with prediction errors), but there might be the need for extensions to it in order to capture additional aspects of usefulness of data in practice.

To actually evaluate the utility in practice, we consider a test set $T \subseteq \mathcal{U}^{\perp}$, possibly derived by removing the values to be predicted from the tuples in the original database that were not used for learning. Then we evaluate the finite sum over all $x \in T$.

3.2 Example

Consider the following real-world scenario: A health insurance company wants to predict whether an insured person can have a specific disease in the future. Assuming that prevention and early detection is cheaper then a treatment, such a prediction will benefit everyone – the patient, the insurance company, and the health care provider. Hence the health care provider is willing to cooperate, but is required by law to protect the privacy of the patients and their health information.

The health care provider releases the information in a privacy-preserving way to the insurance company. Although the data mining is performed over the whole database, the insurance company is not interested in predicting results for all the patients, only in those that have insurance.

The mining algorithm outputs rules that predict a disease, say field x_j. That is, the output is either a patient has/will have the disease or not. It is acceptable to have false-positives, but unacceptable to have missed-positives. The error implication function E_j is

$$E_j(\hat{x}, \bar{x}) = 1 \text{ if } \hat{x}_j - \bar{x}_j \geq 0 \quad \text{and} \quad E_j(\hat{x}, \bar{x}) = 0 \text{ otherwise .}$$

This can be visualized as

$\bar{x}_j$ (true)	$\hat{x}_j$ (predicted)	result	description
0	1	acceptable	false-positive
1	1	good	correct prediction
1	0	unacceptable	missed-positive
0	0	good	correct prediction

That is, each correct and acceptable prediction would add to the utility, while unacceptable predictions will not. Or a negative value can be chosen to actually lower the utility. Regarding the interest weights w, we can simply assign $w(x) = 1$ if the patient x is of interest, and 0 otherwise. The utility is then an integer in the range 0 up to the number of people of interest. The higher the number, the greater is the value of the mining process for the user.

3.3 Utility in the privacy-preserving case

Our theoretical utility is $\texttt{Utility}(\mathsf{DM}, \mathsf{PA}, \mathsf{CR}, DB)$ from the equation (1) such that the series converges. This includes the possibility that $w(x)$ is almost always 0 (i.e., $w(x)$ is non-zero in a finite number of cases, and thus the sum is finite). In practice, the utility of the prediction system working over the original non-sanitized data is computed as

$$\texttt{Utility}_{orig} = \sum_{x \in T} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(DB), x)), \bar{x}),$$

where $T \subseteq \mathcal{U}^{\perp}$ is a finite test set and DB represents the learning set.

We have two possibilities to compute the utility of the prediction system working over sanitized data, that is, a prediction system where the input to the data miner DM is a sanitized database $\mathcal{S}(DB)$. These two possibilities depend on whether the obtained rules are robust – resilient to sanitization. If they are, we can apply the discovered rules to the original data coming from the universe $\mathcal{U}^{\perp}$, and hence we evaluate the utility over a non-sanitized original test set $T \subseteq \mathcal{U}^{\perp}$ as

$$\texttt{Utility}_{san \to orig} = \sum_{x \in T} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(\mathcal{S}(DB)), x)), \bar{x}).$$

The other possibility is that the discovered rules are only applicable to sanitized incomplete tuples. Then the utility is computed as

$$\texttt{Utility}_{san \to san} = \sum_{x \in T} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(\mathcal{S}(DB)), \mathcal{S}(x))), \mathcal{S}(\bar{x})).$$

Now, consider a triple

$$(\texttt{Utility},\ \mathcal{S},\ \mathcal{M})$$

consisting of (a utility definition, a sanitization mechanism, a knowledge discovery algorithm). The $\mathcal{M}$ consists of a data mining algorithm DM that implicitly defines the universe $\mathcal{U}$ (a database schema), a prediction algorithm PA, and a conflict resolution algorithm CR. The utility function $\texttt{Utility}$ depends on the knowledge discovery algorithm $\mathcal{M}$, the sanitization mechanism $\mathcal{S}$, a user defined weight function w and an error implication function E_i, a learning set DB, and a testing set T, all described before.

The *privacy-preserving property* of such a triple is solely depending on the privacy-preserving guarantees of the sanitization mechanism $\mathcal{S}$, when applied to a database coming from the universe $\mathcal{U}$.

We say such a triple is *useful* if

$$\texttt{Utility}_{san \to orig} \geq c \cdot \texttt{Utility}_{orig} \quad \text{or} \quad \texttt{Utility}_{san \to san} \geq c \cdot \texttt{Utility}_{orig},$$

where c is a constant representing some (society accepted) decline in the utility due to a privacy protection.

4 Experimental Results

The objective of our experiments is to determine the effect of sanitization on utility of knowledge discovery, in particular on a prediction system. The question we are addressing is whether the sanitized data can be used to discover knowledge not defined at the time of the sanitization. Based on the experimental results, we want to categorize some triples (utility definition, sanitization mechanism, knowledge discovery algorithm) according to their usefulness.

In our experiments, we use the *Cooperative Learning* (*CoLe*) *system* [8] for knowledge discovery. We also use the diabetes medical data from [8]. For sanitization, we use two methods, namely k-anonymity and ϵ-differential privacy. We first use the sanitization techniques on the original medical data, and then evaluate utility of the knowledge obtained by the CoLe system before and after sanitization.

4.1 The CoLe system

CoLe [8] is a knowledge discovery system. The goal of the CoLe system is to mine medical data and discover hybrid rules that predict an unknown diagnosis. The CoLe system specializes in employing multiple cooperative data miners to achieve this goal.

MEDICAL DATA. The diabetes medical data for our experiments comes from the Calgary Health Region. It was collected for billing purposes in the public health care system. Patients' clinic visits are recorded with diagnostic codes, making the data a good source for data mining studies.

This data contains a group of the population born before 1954 that have been living in Calgary continuously since 1994 (till the time of data collection – 2001). In the original study [8], the patients with no diabetes diagnosis in 1995 - 1999 but at least one diabetes diagnosis in 2000 are of interest. The CoLe system analyzes diagnoses between 1995 and 1999 in order to find rules that can potentially reveal diabetes patients earlier than laboratory tests.

In the medical data provided to us, there are two tables. Table 1 shows some sample data in these two tables. One table is the registration table (REG table) containing ID, gender, year of birth, age group, and the numbers of visits for the last five years together with the average number of visits in those 5 years. This information is provided for 9450 patients. The other table (MD table) contains 2,059,929 medical records coming from health care service events (e.g., visits to a clinic). Each record consist of the date of the service, up to three diagnostic codes, and ID that references the ID in the REG table. The diagnostic codes are defined by the International Classification of Diseases, 9^{th} revision (ICD-9) and are not numerical data but strings. We refer to all this data as the *original data*.

REG

ID	SEX	BYEAR	AGEGRP	VISITS95	V96	V97	V98	V99	AVGVISITS
2	1	1922	8	7	9	12	12	16	11.2
3	0	1924	7	7	6	34	28	14	17.8
5	0	1947	5	5	1	0	0	0	1.2
6	0	1934	6	7	35	52	17	23	26.8
13	0	1928	6	7	5	1	0	0	2.6
18	0	1948	5	4	12	15	8	13	10.4
19	1	1925	7	8	13	7	16	18	12.4

MD

SERVDATE	DIAG1	DIAG2	DIAG3	ID
1996-03-06				2
1999-03-25	595			2
1997-06-27	594.9	788.0		3
1999-12-14	733	717.8	719.4	5
1995-09-06	626.2	V79.0		13
1999-09-13	V79.0	780.9		18
1998-11-16	V81.2			19

Table 1: Sample data in the REG and MD tables

The original data that was provided to us has been already anonymized by omitting the names of the patients, because of the privacy protection measures in the health care system. Still there is sensitive information in the tables that when joined and matched to publicly available information might reveal the identity and other sensitive information about the patients.

We consider the columns with diagnosis codes to be sensitive information. The codes may represent diseases that individuals would like to keep private. Using the other available information in the original data, an individual may be linked to a sensitive value. For example, if we know that an individual was born in 1948, is male, and in 1999 visited the doctor around 50 times, then we can uniquely determine this individual's ID and therefore his diagnoses and diseases.

CoLe system description. The CoLe system [8, 9] consists of two cooperative data mining agents – a sequence miner and a conjunctive miner – and a combination agent. The concept of cooperative learning allows us to produce rules that no single data mining algorithm can produce on its own. Each data mining agent uses a single data mining algorithm on the given data (or part of it). The results from all the mining agents are combined by the combination agent into hybrid rules. This mining-and-combination work forms a mining *iteration* in the CoLe system. The *quality* of the produced rules are measured by both their accuracy and coverage (proportion of the total patients that a rule applies to) and this measure is combined into a fitness value. The higher the fitness value the better is the quality of the mined hybrid rules. This can be seen as trying to assign an indication of utility to a rule. The CoLe system performs several mining iterations and is parametrized by a *fitness threshold*. The rules exceeding this fitness threshold from any iteration are put into the final rule set and output as the mining result.

Applying CoLe to medical data. When mining on the diabetes medical data, the produced hybrid rules contain conditions whether an individual is susceptible to have diabetes and hence whether the individual should be tested for it. In this experiment, we are interested in the obtained hybrid rules. Each hybrid rule is of the form "conditions $\Rightarrow$ possible diabetes". For example, a hybrid rule is

$$gender = M \ \wedge \ \text{has}(466.1) = \text{true} \ \wedge \ \text{seq}(401, 780.3, 405) \ \Rightarrow \ \text{diabetes},$$

which means "if a patient is male, has a diagnosis 466.1 (diseases of respiratory system) at any time, and has diagnosis 401 (hypertension) followed by diagnosis 780.3 (symptoms) and then followed by diagnosis 405 (hypertension), the patient is probably having diabetes and should be tested for it."

The CoLe system is non-deterministic. The randomness comes from several parts of the miner: using a Genetic Algorithm as sequence miner, random selection of some features during conjunctive mining, and random sampling of part of the data during each iteration. We will therefore run the miner several times and average the results.

We use the fitness thresholds 3.6, 3.7, 3.8, 3.9, and 4.0, respectively, which represent the steps from low quality to high quality of the mined hybrid rules. To obtain a hybrid set, we run the CoLe system in 10 iterations, which is enough to facilitate the cooperation during the mining process. For each of the above fitness thresholds we repeat the run of the CoLe system 5 times, obtain 5 hybrid rule sets, and present the average of the results for these 5 runs. We denote this particular system by $\mathsf{CoLe}(t)$, where t is a fitness threshold.

4.2 Data sanitization

Since the original medical data still contains sensitive information and there is the possibility to link individuals to this sensitive information, this data should not be publicly released without sanitization.

Our scenario assumes that the data will be released to the public after sanitization, in particular, using a non-interactive anonymization achieved through k-anonymity [12] and unlimited interaction with a perturbation mechanism which achieves ϵ-differential privacy [5] through value distortion.

These two different sanitization techniques, anonymization and perturbation, are performed on the original data to get the sanitized data. We split each sanitized data set into a *learning set*, denoted DB, and a *test set*, denoted T. We do this by splitting all 9450 patients into two equal-size sets of 4725 patients each. The same 4725 patients are used for learning, unless a sanitization method suppressed some of the patients – removed patient data. Since the two k-anonymized data sets that we produced have some records suppressed, their learning sets are reduced to sizes 4704 and 4701, respectively, for 2- and 100-anonymized data.

ANONYMIZATION USING k-ANONYMITY. k-anonymity [13] is a model that has recently gained popularity. It is a model that releases individuals' sensitive data without compromising the identity of the individuals, and thus it is suitable for releasing medical data in a privacy-preserving way.

For providing anonymity, data holders release medical data by removing the identifiers, such as name, address, phone number with the incorrect belief that patient confidentiality is maintained. Sweeney [13] showed that the remaining data can be used to re-identify individuals by linking some other fields of the data (known as quasi identifiers or QI) to other databases. To solve this problem, she proposed the notion of k-anonymity as a privacy-protection in the released data. The released data satisfies k-anonymity if a combination of the QI-attributes from one row can be found in at least $k-1$ other rows in the released data. k-anonymity is the common technique to achieve individual anonymity in the non-interactive model. We perform k-anonymization of our original data using the Datafly algorithm [12, Figure 8], which we present here as Algorithm 1.

Algorithm 1 Datafly

Input: A data table DB consisting of tuples $x = (x_1, \ldots, x_n)$; a list of quasi-identifier attributes $QI = (A_1, \ldots, A_m)$; constraint k; Generalization Hierarchy Scheme for all quasi-identifier attributes GHS_{A_i} for $i = 1, \ldots, m$.

Output: A generalization of $DB[QI]$ with respect to k.

1: $freq \leftarrow$ a frequency list containing distinct sequences of values of $DB[QI]$, along with the number of occurrences of each sequence
2: **while** there exists sequences in $freq$ occurring less than k times that account for more than k tuples **do**
3: let A_j be attribute in $freq$ having the most number of distinct values
4: $freq \leftarrow$ generalize the values of A_j in $freq$
5: **end while**
6: $freq \leftarrow$ suppress sequences in $freq$ occurring less than k times
7: $freq \leftarrow$ enforce k requirement on suppressed tuples in $freq$
8: **return** new table $DB' \leftarrow DB$ with $DB'[QI]$ coming from the generalized values in $freq$

We use only a single Generalization Hierarchy Scheme (GHS) for this algorithm that works with numerical values. At the bottom of the hierarchy we have a real number that is generalized

into an integer by rounding. The next generalizations are into integer intervals of size 2, 5, 10, 20, and 50. Our integer "intervals" of size s are actually the following sets: ..., $\{0, 1, 2, \ldots, s-1\}$, $\{s, s+1, \ldots, 2s-1\}$, ..., and we choose the ceiling of the mid value of the corresponding interval as a representative. If the interval is $\{Ms, Ms+1, \ldots, (M+1)s-1\}$, then our representative is $\lceil (2Ms+s-1)/2 \rceil$. The last step of our generalization is to replace the value with the symbol *, which represents the top (root) of the GHS.

$$\text{GHS: real number} \rightarrow \text{integer} \rightarrow \text{int(2)} \rightarrow \text{int(5)} \rightarrow \text{int(10)} \rightarrow \text{int(20)} \rightarrow \text{int(50)} \rightarrow *$$

For example, we want to generalize the number 192.34. In the first step we round it to 192. In the next steps, we obtain 193, as $193 = \lceil (192+193)/2 \rceil$ is the representative of the interval $\{192, 193\}$ of size 2; then 192, as $192 = \lceil (190+194)/2 \rceil$ is the representative of the interval $\{190, 191, 192, 193, 194\}$ of size 5; and then 195, 190, 175, and *.

To avoid extensive generalization in the case of small values of k, we employ a heuristic trade-off between generalization and suppression. Our stop condition for small k in the generalization phase of the Datafly algorithm is to stop when no more than 0.5% records would need to be suppressed. So in the case $k = 2$, the line 2 of the Datafly algorithm (Algorithm 1) is modified to

2: **while** there exists sequences in $freq$ occurring less than k times that account for more than k tuples but no more than 0.5% of all the records of DB **do**

Applying k-anonymity to the medical data. We chose the sensitive attributes to be the columns of the MD table and hence the ID's of the REG table. Our quasi-identifier (QI) attributes are all the records in the REG table except ID. We use the following labeling for the QI-attributes: A_1 is the column for gender, A_2 for year of birth, A_3 for age group, $A_4, \ldots, A_8$ for the number of visits during 1995-1999, and A_9 is the column of the average number of visits during this period. Our input to the Datafly algorithm is the REG table.

Datafly outputs 2-anonymization of the original data with roughly 0.41% suppressed records. For a higher privacy level, we choose $k = 100$, and the Datafly algorithm suppresses approximately 0.46% of the records. Table 2 summarizes the suppression of the records and shows the generalization of the QI-columns after k-anonymization. Only the values in the REG table were generalized and suppressed after running the Datafly algorithm. The suppression of some records in the REG table makes the corresponding records (linked by ID's) in the MD table obsolete, and so we suppress them, too.

	# of suppressed records in		# of distinct values in QI-column								
data	REC table	MD table	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
original	-	-	2	50	6	6	103	104	101	106	280
$k = 2$	39 (0.41%)	17124 (0.83%)	2	3	3	3	3	4	3	5	3
$k = 100$	43 (0.46%)	21008 (1.02%)	2	2	3	3	3	2	3	3	2

Table 2: k-anonymization of the medical data

We denote these two described sanitization mechanisms by $\mathcal{S}(k = 2)$ and $\mathcal{S}(k = 100)$.

Data perturbation for ϵ-differential privacy. We use interactive perturbation by value distortion that achieves ϵ-differential privacy, because, compared to other perturbation methods

such as the one of Agrawal and Srikant [2], it quantifies and limits the risk of disclosure of sensitive information.

The definition of ϵ-differential privacy is [4]: A randomized function $\mathcal{S}$ gives *ϵ-differential privacy* if for all data sets D_1 and D_2 differing on at most one element, and all $R \subseteq Range(\mathcal{S})$, $\Pr[\mathcal{S}(D_1) \in R] \leq \exp(\epsilon) \times \Pr[\mathcal{S}(D_2) \in R]$.

This notion limits the probability that the randomized function $\mathcal{S}$, a sanitization function, would leak information from a data set that is extended by at most one element. That is, a data leak and therefore a disclosure of private data through the function $\mathcal{S}$ is possible, but limited by the leakage parameter ϵ.

Dwork et al. [5] showed that ϵ-differential privacy can be achieved in the interactive model using value distortion technique, which to a numerical value x adds a random noise r drawn from a known distribution.

The random noise depends on the query function. It is drawn from a Laplace distribution $\text{Lap}(\mu, b)$, with probability density function $h(y) = \frac{1}{2b} \exp\left(- \frac{|y - \mu|}{b} \right)$. Let DB be a data set, and let f be a query function with range $\mathbb{R}^d$, that is, a query outputs a vector of numerical values. The L1-sensitivity Δf of f is defined as [4]: $\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$ for all D_1 and D_2 differing in at most one element. If an answer $x = f(DB)$ is masked by adding a random noise, a vector $r \in \mathbb{R}^d$ to x, whose values are chosen independently and identically from Laplace distribution $\text{Lap}(\mu, b)$, with location (mean, median, mode) $\mu = 0$, and scale $b = \Delta f / \epsilon$ depending on the leakage parameter ϵ and the query function f, then the sanitized answer $x + r$ achieves ϵ-differential privacy. That is, the noise is drawn from $\text{Lap}(0, \Delta f / \epsilon)$ with probability density function $h(y) = \frac{\epsilon}{2\Delta f} \exp\left(- \frac{\epsilon |y|}{\Delta f} \right)$.

Applying perturbation to the medical data. We use leakage $\epsilon = 0.01$ and we set sensitivity Δf to be 1 for all query functions f, which represents the worst-case scenario. The additive noise for each numerical value is then effectively drawn from the Laplace distribution $\text{Lap}(0, 1/\epsilon)$ with probability density function $h(y) = \frac{1}{200} \exp\left(- \frac{|y|}{100} \right)$.

In addition to perturbation of numerical values, we use perturbation of categorical values. In particular, we use the technique called *uniform randomization* [6], which is a generalized idea of Warner's randomized response technique. The perturbation mechanism replaces a categorical value with some other value in the same category with probability p and keeps it with probability $p - 1$. The larger the probability p, the higher is the privacy protection.

We use the uniform randomization technique to perturb diagnostic codes (columns DIAG1, DIAG2, and DIAG3 in the MD table). With probability $p = 0.75$ we replace the original diagnostic code with another value chosen uniformly at random from the set of all (ICD-9) diagnostic codes, and with probability 0.25 we keep the original diagnostic code.

For our experiments we use two different combinations of perturbation: (1) a perturbation of all numerical values using the value distortion technique and noise from the Laplace distribution, as suggested in [5], and (2) the same perturbation of all numerical values using the value distortion technique as in (1) plus a perturbation of diagnostic codes (categorical values) using the uniform

randomization technique, as suggested in [6]. In addition, we use two variants of perturbation. Additive noise was (a) independently chosen from the distribution for each numerical value, and (b) independently chosen from the distribution for each *new* numerical value and fixed for the same values. This avoids partial disclosure of an original value by repeating the query and obtaining an approximation – a bounded estimator for the original value [1, 2]. The original data is perturbed using (1) or (2), and (a) or (b). The four obtained perturbed data sets are labeled 1a, 1b, 2a, and 2b, accordingly. The corresponding sanitization algorithms will be denoted as $\mathcal{S}(\epsilon = 0.01, \text{variable noise})$, $\mathcal{S}(\epsilon = 0.01, \text{fixed noise})$, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{variable noise})$, and $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{fixed noise})$, respectively.

4.3 Calculating utility

After performing the two mentioned sanitization methods on the original medical data and generating six sanitized data sets, we computed the utility of the knowledge discovered from the original data and the sanitized data in order to compare them.

To compute the utility of the knowledge discovered over the original non-sanitized data, we use the utility function that we derived in Section 3.3:

$$\texttt{Utility}_{orig} = \sum_{x \in T} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(DB), x)), \bar{x}).$$

In this formula, the data mining algorithm DM is the CoLe system as described above. $\mathsf{DM}(DB)$ means that we apply the CoLe system to the learning set from our original data. As a result our miner outputs a set R of hybrid rules that predict diabetes, say a boolean field x_0, and the rules only predict whether x_0 is true. The CoLe system outputs rules that predict only one value, namely x_0, and so the index set $I = \{0\}$. Table 3, row "Original data", summarizes our mining efforts over the unsanitized learning set DB.

PA is our prediction algorithm, which takes the set of rules R and an incomplete tuple x, representing information about a patient with the field x_0 missing, and determines if the individual rules of R are applicable to x. Even if one rule from R is applicable, that is, predicts that a patient represented by x has possible diabetes (x_0 is true), then our conflict resolution algorithm CR chooses true for x_0.

For the error implication function E_0 and weights w's we need to know the preferences of the end user. Suppose that the end user of our system is a health care provider for whom it is acceptable to have false-positives, but undesirable to have missed-positives. The CoLe system predicts only whether x_0 is true. If there is no prediction, the health care provider does not know whether the patient has or does not have diabetes. Although in reality we cannot determine x_0 in this case, we set x_0 to 0 if there was no prediction, to capture false-positives and correct negative predictions. The health care provider then uses the following error implication function

$$E_0(\hat{x}, \bar{x}) = 1 \text{ if } \hat{x}_0 - \bar{x}_0 \geq 0 \text{ and } E_0(\hat{x}, \bar{x}) = 0 \text{ otherwise },$$

which represents

$\bar{x}_0$ (true value)	$\hat{x}_0$ (predicted value)	result	description
0	1	acceptable	false-positive
1	1	good	correct prediction
1	N/A $\Rightarrow$ 0	undesirable	missed-positive
0	N/A $\Rightarrow$ 0	good	correct prediction.

Regarding the interest weights w's, the health care provider assigns $w(x) = 1$ for all patients represented by x that are in the test set T.

Now the utility $\texttt{Utility}_{orig}$ can be computed, and it is an integer in the range 0 up to the number of patients in the test set T with diabetes. The average $\texttt{Utility}_{orig}$ computed from the results of 5 mining runs is shown in Table 4, in the row labeled "Original data".

In practice we have two possibilities to compute the utility of discovered knowledge from sanitized data, depending whether the knowledge is robust or not, that is, whether the obtained rules are resilient to sanitization or not. Either the knowledge is robust, and then we can apply the rules obtained from mining over sanitized data to make prediction about non-sanitized data (san $\rightarrow$ orig):

$$\texttt{Utility}_{san \rightarrow orig} = \sum_{x \in T} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(\mathcal{S}(DB)), x)), \bar{x}).$$

Or the knowledge is not robust, and then we should only apply the rules obtained from mining over sanitized data to make predictions about sanitized data (san $\rightarrow$ san):

$$\texttt{Utility}_{san \rightarrow san} = \sum_{x \in T} w(x) \sum_{i \in I} E_i(\mathsf{CR}(\mathsf{PA}(\mathsf{DM}(\mathcal{S}(DB)), \mathcal{S}(x))), \mathcal{S}(\bar{x})).$$

In both cases, we obtain the rules from running the CoLe system DM over the sanitized learning set $\mathcal{S}(DB)$. But then we can apply the rules using our prediction algorithm PA to either an unsanitized incomplete tuple x or a sanitized incomplete tuple $\mathcal{S}(x)$. All the algorithms DB, PA, and CR are the same as before. The averaged $\texttt{Utility}_{san \rightarrow san}$ and $\texttt{Utility}_{san \rightarrow orig}$ from 5 runs for each sanitized data set and our selected fitness thresholds is shown in Table 4.

4.4 Discussion

Table 3, row "Original data", summarizes our mining efforts over the unsanitized learning set DB. The other rows present mining efforts over the specified sanitized learning sets $\mathcal{S}(DB)$'s. The columns show the average number of obtained hybrid rules in R from 5 runs for each of the fitness thresholds 3.6, 3.7, 3.8, 3.9, and 4.0.

We see that the higher is the fitness threshold (the higher quality rules we request), the fewer or equal number of rules is obtained. This is an expected behavior of the CoLe system for the unsanitized learning set DB, and it remains true even when we mine over the sanitized learning sets $\mathcal{S}(DB)$'s – the only exception being when applying the CoLe system with fitness threshold 3.9 to Perturbed data 1b, and this exception is possibly a statistical anomaly due the random effects in the mining and the relatively small number of repetitions.

The CoLe system discovered more hybrid rules when mining over unsanitized data, with some exceptions of mining over k-anonymized data. These exceptions are due to the generalization of

Fitness Threshold → Learning Set ↓	3.6	3.7	3.8	3.9	4.0
Original data	791.2	146.8	32.8	8.0	0.4
Perturbed data 1a	592.2	37.8	1.0	0.2	0.0
Perturbed data 1b	595.2	49.0	2.0	2.4	0.0
Perturbed data 2a	16.0	0.0	0.0	0.0	0.0
Perturbed data 2b	3.6	0.0	0.0	0.0	0.0
2-anonymized data	795.0	75.0	62.4	5.0	2.4
100-anonymized data	568.6	142.6	56.8	13.2	6.2

Table 3: The average number of obtained hybrid rules from 5 runs

the columns of the REG table that resulted in more uniform values in the columns, so the mined rules had higher support, and therefore more rules passed the fitness threshold. In other words, k-anonymity seems to remove outliers.

For Perturbed data 2a and 2b, the CoLe system with fitness threshold 3.6 discovered very few and with fitness threshold 3.7 - 4.0 no hybrid rules. The perturbation of diagnostic codes in these two learning sets disguised the data in a way that the CoLe system was unable to determine meaningful patterns and trends in the data from. Note that the other perturbed data (1a and 1b) as well as k-anonymized data did not change the diagnostic codes, because, respectively, they were categorical data and we were perturbing numerical data, and because the diagnostic codes were the sensitive data for k-anonymization and the anonymization process does not change them. The number of discovered rules, however, is not our criterion to evaluate the prediction, as it does not take into account the preferences and goals of an end user. That is why we have introduced the utility function.

Table 4 presents the average of five computed utility values of knowledge that was discovered in 5 runs. The row "Original data" corresponds to $\texttt{Utility}_{orig}$, that is, to the utility of knowledge discovered over the unsanitized learning set applied to the unsanitized testing set. We call this utility the *original utility*. The remaining rows of the left part of the table correspond to $\texttt{Utility}_{san \to orig}$, and the rows of the right part of the table correspond to $\texttt{Utility}_{san \to san}$.

Fitness Threshold ↘	$\texttt{Utility}_{orig}$ and $\texttt{Utility}_{san \to orig}$					$\texttt{Utility}_{orig}$ and $\texttt{Utility}_{san \to san}$				
Knowledge From ↓	3.6	3.7	3.8	3.9	4.0	3.6	3.7	3.8	3.9	4.0
Original data	1942.6	1532.2	431.0	253.4	86.6	1942.6	1532.2	431.0	253.4	86.6
Perturbed data 1a	1962.6	1224.6	216.2	59.0	0.0	1958.2	1223.0	216.2	45.8	0.0
Perturbed data 1b	1915.0	1227.2	175.8	80.0	0.0	1909.4	1204.6	144.6	52.8	0.0
Perturbed data 2a	9.4	0.0	0.0	0.0	0.0	136.8	0.0	0.0	0.0	0.0
Perturbed data 2b	0.0	0.0	0.0	0.0	0.0	26.8	0.0	0.0	0.0	0.0
2-anonymized data	1907.2	1290.6	518.2	188.4	90.4	1888.4	1253.2	463.4	160.2	72.4
100-anonymized data	1863.0	1402.4	636.8	191.6	141.0	1831.4	1086.6	364.2	156.0	75.6

Table 4: Average utility of the discovered knowledge from 5 runs

In most of the experiments, the number of false positives was well below 2 times the number of correct predictions, except in the case of Perturbed data 2a where it was 7.35. Yet health and

insurance professionals are willing to accept higher ratio of false positives to correct predictions.

The mining over Perturbed data 2a and 2b produced almost no hybrid rules, and there is almost no utility – close to 0 and less than 8% compared to the original utility. The perturbation of diagnostic codes in these two learning sets disguised the data in a way that the miner was unable to determine meaningful patterns and trends in the data. This effectively shows that there is a knowledge discovery method that provides utility over unsanitized data, but *fails* when used over sanitized data. This also justifies the need to look at the triples (utility definition, sanitization mechanism, knowledge discovery method) and categorize them according to their usefulness.

The utility in the cases of Perturbed data 1a and 1b is most of the time lower than the original utility, the only exception arising from computing the utility of knowledge discovered by the CoLe system with fitness threshold 3.6 over Perturbed data 1a. For the remaining fitness thresholds and for both Perturbed data 1a and 1b, the loss of utility is roughly 11%, 66%, 82%, and 100%, respectively, as the fitness threshold goes up. Some decline in the utility is expected and acceptable, since a privacy-preserving mechanism is in place, but a high loss of utility should be considered unacceptable.

For the 2- and 100-anonymized data, the utility is in the range 61% – 163% of the original utility. When this utility is lower than the original utility, it is still acceptable and higher than the utility in the case of Perturbed data 1a and 1b. For the 2- and 100-anonymized data and fitness thresholds 3.8 and 4.0, and for the 2-anonymized data and fitness threshold 3.8, we obtained a utility that is higher than the original utility. This is due to the higher number of discovered hybrid rules in these cases and the fact that k-anonymization generalizes and smooths data.

The utility that is non-trivial is most likely due to the facts that perturbation and anonymization in the cases of Perturbed data 1a and 1b, and 2- and 100-anonymized data left the diagnostic codes of the MD table unchanged, and that the CoLe system puts more weight to diagnostic codes than to the other data. Recall that the fitness threshold represents the quality of the discovered rules. It is rather obvious that many lower quality rules can provide a similar utility than a few high-quality ones. As the fitness threshold goes up, the number of discovered rules goes down, and this together with the variance of the utility of single high-quality rules explains the drastic decline of the minimum utility and the variance of the utility values in Table 4.

Finally, we note that it is very interesting that for Perturbed data 1a and 1b, and 2- and 100-anonymized data, the utility of the knowledge discovered over sanitized data when applied to unsanitized data is at least as much as when applied to sanitized data ($\texttt{Utility}_{san \to orig} \geq \texttt{Utility}_{san \to san}$). However, we do not have a readily available explanation for this phenomenon, as one would expect that knowledge discovered from sanitized data would be "more applicable" to sanitized data than to some other data, namely unsanitized data.

Classification of triples. Recall that by $\mathsf{CoLe}(t)$ we denote the cooperative learning system that works in 10 iterations over the medical data and uses fitness threshold t. $\mathcal{S}(\cdots)$ denotes the sanitization mechanism, either using k-anonymity, ϵ-differential privacy, or p-uniform randomization, with either fixed or variable noise for repeated queries. Finally, `Utility` will denote our utility function that can be used over original as well as sanitized data. We use $w(x) = 1$ as the weight function and $E_0(\hat{x}, \bar{x}) = 1$ if $\hat{x}_0 - \bar{x}_0 \geq 0$ and $E_0(\hat{x}, \bar{x}) = 0$ otherwise as the error implication function.

In Table 5 we present a list of triples (utility definition, sanitization mechanism, knowledge discovery method) and categorize them by their usefulness in practice. We call a triple "Great" if the knowledge discovery over sanitized data provides better utility than over unsanitized data (i.e., achievement of at least 100% in either of the two cases of applying the discovered knowledge from sanitized data to unsanitized or sanitized data), "good" if 75% – 100% of the original utility is achieved, "acceptable" if 50% – 75% of the original utility is achieved, and "unacceptable" otherwise.

triple (`Utility`, sanitization $\mathcal{S}$, miner $\mathcal{M}$)	**category**
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{variable noise})$, CoLe(3.6))	good
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{variable noise})$, CoLe(3.7))	good
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{variable noise})$, CoLe(3.8))	acceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{variable noise})$, CoLe(3.9))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{variable noise})$, CoLe(4.0))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{fixed noise})$, CoLe(3.6))	good
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{fixed noise})$, CoLe(3.7))	good
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{fixed noise})$, CoLe(3.8))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{fixed noise})$, CoLe(3.9))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, \text{fixed noise})$, CoLe(4.0))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{variable noise})$, CoLe(3.6))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{variable noise})$, CoLe(3.7))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{variable noise})$, CoLe(3.8))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{variable noise})$, CoLe(3.9))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{variable noise})$, CoLe(4.0))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{fixed noise})$, CoLe(3.6))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{fixed noise})$, CoLe(3.7))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{fixed noise})$, CoLe(3.8))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{fixed noise})$, CoLe(3.9))	unacceptable
(`Utility`, $\mathcal{S}(\epsilon = 0.01, p = 0.75, \text{fixed noise})$, CoLe(4.0))	unacceptable
(`Utility`, $\mathcal{S}(k = 2)$, CoLe(3.6))	good
(`Utility`, $\mathcal{S}(k = 2)$, CoLe(3.7))	good
(`Utility`, $\mathcal{S}(k = 2)$, CoLe(3.8))	Great
(`Utility`, $\mathcal{S}(k = 2)$, CoLe(3.9))	acceptable
(`Utility`, $\mathcal{S}(k = 2)$, CoLe(4.0))	Great
(`Utility`, $\mathcal{S}(k = 100)$, CoLe(3.6))	good
(`Utility`, $\mathcal{S}(k = 100)$, CoLe(3.7))	good
(`Utility`, $\mathcal{S}(k = 100)$, CoLe(3.8))	Great
(`Utility`, $\mathcal{S}(k = 100)$, CoLe(3.9))	good
(`Utility`, $\mathcal{S}(k = 100)$, CoLe(4.0))	Great

Table 5: Categorization of triples based on our experiments.

5 Conclusions

The loss of the utility of the knowledge discovered by the cooperative learning system from perturbed data is often much more than what should be acceptable, and in some cases, there is no utility at all. On the other hand, k-anonymization of data is an acceptable sanitization mechanism that provides a privacy protection and preserves the utility (and even might provide better utility) of the knowledge discovered by the cooperative learning system.

Since we have shown it is impossible to obtain knowledge from every knowledge discovery method working over sanitized data, this leads to the following idea: a sanitization mechanism must be tailored to guarantee utility to certain knowledge discovery methods to the exclusion of others. We believe that such a guarantee should be an integral part of privacy notions. For example, the k-anonymity notion should, in addition to privacy guarantees, also specify what kind of knowledge discovery is possible over k-anonymized data that would provide reasonable utility. We demonstrated this concept by looking at triples (utility definition, sanitization mechanism, knowledge discovery method) and categorizing them according to their usefulness in practice.

We plan to classify some existing knowledge discovery methods that give reasonable utility of knowledge discovered from data that is sanitized using some widely used sanitization methods.

Future investigation and development of sanitization methods should concentrate on those methods that would guarantee some reasonable degree of utility to several knowledge discovery methods. Ideally, there would be one (probably interactive) sanitization method that would protect the privacy of sensitive information and be useful for all knowledge discovery methods.

Another direction for future research is the following problem: There is a data set D, a sanitization method $\mathcal{S}$, and several knowledge discovery methods that can discover knowledge from D, but which fail to discover knowledge from $\mathcal{S}(D)$. Are there modified knowledge discovery methods, with the same set of objectives as the original knowledge discovery methods, that can succeed to discover meaningful, reasonable, and useful knowledge from $\mathcal{S}(D)$? That is, do we need to modify existing knowledge discovery methods to be able to obtain knowledge from sanitized data?

References

[1] Adam, N. A., and Wortman, J. C. Security-control methods for statistical databases. *ACM Computing Surveys 21*, 4 (1989), 515–556.

[2] Agrawal, R., and Srikant, R. Privacy-Preserving Data Mining. In *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data* (2000), ACM, pp. 439–450.

[3] Clifton, C., Kantarcioglu, M., and Vaidya, J. S. Defining Privacy For Data Mining. In *Proc. of the National Science Foundation Workshop on Next Generation Data Mining* (Baltimore, MD, USA, November 2006), pp. 126–133.

[4] Dwork, C. Differential Privacy. In *Proc. of the 33rd International Colloquium on Automata, Languages and Programming, Part II* (2006), LNCS 4052, Springer-Verlag, pp. 1–12.

[5] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proc. of the 3rd Theory of Cryptography Conference* (2006), LNCS 3876, Springer-Verlag, pp. 265–284.

[6] Evfimievski, A., Srikant, R., Agarwal, R., and Gehrke, J. Privacy preserving mining of association rules. *Information Systems 29*, 4 (2004), 343–364.

[7] Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. From Data Mining To Knowledge Discovery: An Overview. In *Advances In Knowledge Discovery And Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI Press/The MIT Press, Menlo Park, CA, USA, 1996, pp. 1–34.

[8] Gao, J., Denzinger, J., and James, R. C. A Cooperative Multi-agent Data Mining Model and Its Application to Medical Data on Diabetes. In *Proc. Autonomous Intelligent Systems: Agents and Data Mining* (2005), LNAI 3505, Springer-Verlag, pp. 93–107.

[9] Gao, J., Denzinger, J., and James, R. C. CoLe: A Cooperative Data Mining Approach and Its Application to Early Diabetes Detection. In *Proc. of the 5th IEEE International Conference on Data Mining (ICDM 2005)* (Houston, Texas, USA, 2005), IEEE Computer Society, pp. 617–620.

[10] Kantarcioglu, M., and Clifton, C. Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. *IEEE Transactions on Knowledge and Data Engineering 16*, 9 (2004), 1026–1037.

[11] Rizvi, S. J., and Haritsa, J. R. Maintaining data privacy in association rule mining. In *Proc. of the the 28th International Conference on Very Large Data Bases* (Hong Kong, China, August 2002).

[12] Sweeney, L. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5 (2002), 571–588.

[13] Sweeney, L. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5 (2002), 557–570.

[14] Vaidya, J. S., and Clifton, C. Privacy preserving association rule mining in vertically partitioned data. In *Proc. of the eighth ACM SIGKDD International Conference on Knowledge discovery and data mining* (2002), ACM Press, pp. 639–644.

[15] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., and Theodoridis, Y. State-of-the-art in privacy preserving data mining. *SIGMOD Record 33*, 1 (2004), 50–57.