δ-privacy: Bounding Privacy Leaks in Privacy Preserving Data Mining

Zhizhou Li[1(✉)] and Ten H. Lai[2]

[1] The Voleon Group, Berkeley, USA
lizhizhou@gmail.com
[2] The Ohio State University, Columbus, USA
lai@cse.ohio-state.edu

Abstract. We propose a new definition for privacy, called δ-privacy, for privacy preserving data mining. The intuition of this work is, after obtaining a result from a data mining method, an adversary has better ability in discovering data providers' privacy; if this improvement is large, the method, which generated the response, is not privacy considerate. δ-privacy requires that no adversary could improve more than δ. This definition can be used to assess the risk of privacy leak in any data mining methods, in particular, we show its relations to differential privacy and data anonymity, the two major evaluation methods. We also provide a quantitative analysis on the tradeoff between privacy and utility, rigorously prove that the information gains of any δ-private methods do not exceed δ. Under the framework of δ-privacy, it is able to design a pricing mechanism for privacy-utility trading system, which is one of our major future works.

1 Introduction

Privacy is a high-profile public issue that attracts attention from the entire society. Information collectors and/or processors, such as Internet business, market consulting companies, and governments, are eager to collect as much information as they could to discover people's behavioral patterns. They acquires information from a large number of people, then extract useful knowledges (e.g., statistics and patterns) from the data set using statistical methods and/or data mining techniques. However, data mining on genuine data would be harmful to data privacy, data providers are not willing to commit their sensitive data to a untrusted data collector.

The goal of privacy preserving data mining is to extract knowledge from a data set, while maintaining data contributor's privacy. However, there is a tradeoff between knowledge discovering and privacy preservation: the more knowledge discovered from the data set, the higher possibility that the privacy is leaked. To preserve privacy, a data mining method should hide a certain amount of information before giving them to public.

This work was done while this author was studying in The Ohio State University.

J. Garcia-Alfaro et al. (Eds.): DPM/CBT 2017, LNCS 10436, pp. 124–142, 2017.
DOI: 10.1007/978-3-319-67816-0_8

There are two major classes of privacy protection methods in knowledge discovering, one is input perturbation, which pre-process the input data to "de-privatize" them, then extract knowledge from the modified data; the other is output perturbation, which generates the true result on the original data set, but modify it before publishing.

Input Perturbation. Input perturbations are also called data anonymity, the idea was first proposed by Sweeney [23]. A database is divided into different equivalence classes with a "identifying tag," records that fit the identifying information will fall into that class. Then those classes, along with the original private data from each record, are published. In that case, even if an adversary knew a target's identifying information and locates it in a equivalence class, he cannot determine which one is it because there are multiple records having the same identifying tag.

Sweeney proposed a rule for grouping similar records, called k-anonymity [23], it requires each class to hold at least k records, such that even if an adversary can identify that a target falls in one class, he only has $1/k$ chance to identify the target record in the class. However, having k entries in a class is not enough, if everyone in a class has the same private data, an adversary can still learn the secret. l-diversity [17] is proposed based on k-anonymity, requiring that each equivalence class should contain k records and l distinct values of private data. t-closeness principle [13] is then developed based on k-anonymity and l-diversity, it further requires the distribution of private data in a equivalence class should be *close* to that in the whole data set. See Definition 10 for more details.

Output Perturbation. Instead of modifying the input data, output perturbation methods modify the data mining results. *Differential privacy* [6,7] is a privacy constraint on how to change the true result to prevent privacy leak. The result should be "de-privatized" such that any adversary is not able to distinguish whether it is generated from a data set that contains the target record, or from a data set that does *not* contain the target. If the target is not in a data set, the data mining result does not contain any information about the target, therefore, adversary is not able to learn any privacy of the target from such a unrelated result; if a data mining result "looks like" such a unrelated result, an adversary is not likely to discover useful fact about the target either.

Our Contribution. In this paper, we propose a metric that measures the largest possible privacy leak in a knowledge discovering process, and provide a new definition of privacy, called δ-privacy, that restricts the privacy leak in a data mining process. Compared with existing privacy definitions, δ-privacy is from an adversary's perspective; by studying adversaries' behavior, we can tell how they are going to harm privacy, that is the most direct way to assess the risk of privacy leak. Under this framework, we are able to tell the data providers what's the risk that their secret is learned by an adversary, and in the meantime, we can tell the data collectors how much information they could acquire. We also show the following:

1. Our framework can be used to evaluate any privacy preserving data mining algorithms; in particular, we show the relation between δ-privacy and existing privacy definitions, namely, if a method satisfy ϵ-differential privacy [6], it will also satisfy δ-privacy with $\delta = \epsilon$ (Lemma 2); if a data anonymity method satisfies t-closeness [13] w.r.t. variational distance, it also satisfies δ-privacy with $\delta = t$ (Lemma 3).
2. Utility and privacy are zero-summed in the sense that the maximum utility gained by the data processor is bounded by the maximum privacy leak allowed (Theorem 1). To obtain more information from the data set, data processors need to ask the data provider to increase the privacy loss limit, i.e., δ. To the best of our knowledge, this is the first work that quantitatively proves this idea.

Informally speaking, if a data mining method is δ-private, then any *partial privacy* of data providers are protected, in the sense that no adversary can tell the secret much better than random guessing. We model a partial privacy of a target record r as a binary-valued function priv on r. An adversary discloses the secret fact by calculating $\mathsf{priv}(r)$. We assume that the adversary is *well-informed*, he is aware of all publicly accessible information of r, he knows the intrinsic knowledge of priv, i.e., the distribution of priv over the records in the database. He makes queries to the database, obtains result m (which is generated by a privacy preserving data mining method) to discover more knowledge, then he outputs one bit as the prediction of $\mathsf{priv}(r)$. We say the data mining result reveals privacy if given m, the adversary can compute $\mathsf{priv}(r)$ with higher success rate versus that before he gets the result. More generally, an adversary $\mathcal{A}$ computes $\mathsf{priv}(r)$; his success rate increases after he gets m. This difference in probability is the improvement of $\mathcal{A}$ after getting m, it reflects how much privacy $\mathcal{A}$ learns from m. We use this difference as the indicator of privacy leak and say a data mining method is δ-private if the privacy leak is always smaller than δ.

Our research focus on large scale databases, which enroll enough samples from the real world. Therefore, the distribution of $\mathsf{priv}(r)$ over the records in the database is close to that over the whole human population. This distribution is an important prior knowledge for adversaries to predict priv on a particular target.

The utility of a data mining method is the quantity of information that one could learn from the data mining result, while the utility gain is the utility minus the intrinsic utility, which can be achieved without querying. This research shows that privacy loss and utility gain are zero-summed, the maximum knowledge that can be extracted is no more than δ. To get more utility, data processor should convince (using money) the data providers to increase the limit of privacy leak. This enables "privacy trading" in the future: data providers can put a price tag on their data based on the risk of privacy leak, and the data user can determine the amount of information to purchase based on how much utility he could get from the data mining process.

2 Preliminaries

Records and Tables. A record is composed of fields, each field contains one attribute of the record. More specifically, a record is a member of a Cartesian product $\mathcal{F} \triangleq \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_k$, where $\mathcal{F}_i$ is a finite set of all possible values for a field. A table $\mathcal{X}$ is a collection of records. There are 2 different types of fields:

1. identifiers, such as name, social security number, or other information that uniquely identifies a record in the database, or quasi-identifiers, such as address, sex, etc. which can help to, albeit not uniquely, identify a record in the database.
2. sensitive data, the information that the data contributors want to keep secret from the public.

Take the farmer's survey database (Table 1) as an example, gender, age, zip code, and owned acres are quasi-identifiers, the combination of those fields can be used to identify a farmer. Note that those fields are accessible to public: gender and age are not considered secret to a farmer; his address and how many acres he owns are available in county auditor. If an attacker knows a farmer in person, he is able to get all those facts. On the other hand, the rented acres and rental rates are sensitive fields in this database, they are not publicly accessible. Also, because they are directly related to farmers' income, farmers would like to keep them secret. In this paper, we consider identifiers and quasi-identifiers as *public fields*, which are accessible to public, and consider sensitive data as *secret fields*, which should be kept secret from the public.

Table 1. The Original Farmers Database. In this table, gender, age, zip code and owned acres are quasi-identifiers; rented acres and rental rate are considered secret.

Gender	Age	Zip_code	Owned_acres	Rented_acres	Rental_rate
Female	43	43111	100	120	130
Male	37	42102	551	1100	140
Male	35	43110	120	91	125
Male	56	43208	625	110	180
Male	31	43315	220	630	175
Male	51	43111	64	0	NA
Female	45	43102	250	2000	200
Male	37	43215	320	1200	200
Male	41	43215	580	400	170
Male	25	43102	200	200	150

Queries. A query is a question or request for information to the database management system (or database for short). We regard it as step-by-step instructions which retrieve information and/or discover knowledge from the database.

Traditionally, when given a query, the database follows the all instructions in the query, then returns the result as response.

In our research, there is no constraint on the query, an adversary can ask any question. Also, the adversary is allowed to ask multiple questions, and the same question can be asked for multiple times. See Sect. 4 for more discussion.

Privacy-Preserving Data Mining Methods. We regard a privacy preserving data mining method $\mathcal{M}$ as a mechanism to generate responses to queries. $\mathcal{M}(\mathsf{q}, \mathcal{X})$ takes as input one query q and a database $\mathcal{X}$, it produces a result of q, but that result should not reveal privacy of the records in $\mathcal{X}$. Obviously, if it exactly follows the instructions in q as a traditional database does, it is not safe to data providers' privacy because adversaries can design queries to retrieve secret data he wants. To protect privacy, $\mathcal{M}$ would either (1) follow the instructions in q but add noise to the result, or (2) follow the instructions but perform them over the de-privatized table, or (3) takes other possible approaches that would not follow the instructions but still generate a response.

Usually, $\mathcal{M}$ is a randomized algorithm, the message generated by it is a random variable. We denote by

$$m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})$$

the messages/transcripts generated by $\mathcal{M}$ and sent to $\mathcal{A}$.

3 Attacker Model

To define privacy, we first discuss what information an adversary wants to learn, and what prior knowledge he already has. In our discussion, we always assume that an adversary, who wants to disclose *partial privacy* of records in the database, is *well informed*, which means (1) he knows all the *public information,* e.g., the identifying information, of the target, and (2) he holds *intrinsic knowledge,* i.e., he know how normal people behave. To discover more knowledge about the target, he makes query to a database, which runs privacy preserving data mining algorithms to answer queries. Finally, he makes a judgment on the target. In this section, we will explain the above concepts.

What to Learn. In a database, sensitive fields contains private information of a record. If any adversary is able to learn any partial information about a secret field (not necessarily the whole field), it is considered a privacy leak. Take the farmers' survey as example (see Table 1), rented acres and rental rates are secret fields. Adversaries may not know exactly how many acres a farmer rents, but they are more interested in partial information like "does Bob rent over 200 acres" or "is Alice's rental rate in between 90 to 110 dollars" or "is Eve's rental acres below average." Conceptually, a partial privacy is a statement about the records in the database. If an adversary can correctly determine whether the statement is true or false, we say the adversary discloses the partial privacy of the target record.

We model such partial privacy as *privacy predicates*. A privacy predicate $\mathsf{priv} : \mathcal{F} \mapsto \{0,1\}$ is a binary-valued function, its input is a record r (including all fields), its output is either true or false on some statement about r. If the statement is true, then $\mathsf{priv}(r) = 1$, otherwise $\mathsf{priv}(r) = 0$. An adversary is then an algorithm computing $\mathsf{priv}(r)$ based on his knowledge about r.

Not every predicate is a privacy predicate. Computing priv on r should involve at least one bit in the sensitive fields; predicates that can be computed using only public information are not private. We will precisely define privacy predicates in Definition 1. They should not be computed by any *well-informed* adversary, who holds the public information of the target record, as well as the intrinsic knowledge of the privacy, which we will discuss below.

Public Information. Public information of a record consists of (quasi-) identifying fields of the record. We denote the public information as a function $\mathsf{pub} : \mathcal{F}_1 \times \cdots \times \mathcal{F}_k \mapsto \mathcal{F}_1 \times \cdots \times \mathcal{F}_l$ that maps a record to its public fields, where fields $\mathcal{F}_1, \ldots, \mathcal{F}_l$ are (quasi-)identifiers, and $\mathcal{F}_{l+1}, \ldots, \mathcal{F}_k$ are sensitive fields. Public information is assumed accessible to adversaries. In the farmer survey, the public information of a farmer are her/his gender, age, owned acres and zip code. Those information are considered not private, for example, a curious neighbor of Bob is potentially an attacker of Bob. He knows the Bob's age and location, and he can learn the farmer's ownership information from county auditor. These information are not private to Bob's neighbor. Any information that is derivable from those fields are not considered private.

Once an adversary knows a target's public information, he is able to perform queries to discover more particular information about the target. For example, if Bob's zip code is 43111, adversaries design query like "what is the average rental rate in area of zip code 43111", where *zip code 43111* is used to confined the search range.

Public information function $\mathsf{pub}(r)$ is "one-wayed": it is easy to compute given r, but given $\mathsf{pub}(r)$, it is hard to recover all fields of r. We assume that at least some "hardcore" bits in the sensitive fields $\mathcal{F}_{l+1} \times \cdots \times \mathcal{F}_k$ are not computable given $\mathsf{pub}(r)$, otherwise, if every bit in the sensitive fields is computable given the public information, we say the database contains no secret. See "Comments on Privacy Predicates and Hardcore Predicates" on Page 8 for more discussion on the one-wayness of public informations.

Intrinsic Knowledges. Intrinsic knowledges are also referred to as common knowledges, existing knowledge or prior knowledges. These knowledges are about the privacy itself, they may come from the nature of priv, or from the perception of behaviors of general people, or from social statistics, etc. Intrinsic knowledges are of great importance in predicting priv on a particular target. For example, the rental rates are from 0 to 500 dollar per acre, and this knowledge is a common sense for all people, so the fact "is Bob's rental rate less than 800" becomes trivial because every adversary can answer it with 100% success rate. Another example is, an experienced market analyst is aware that most rental rates are below 400 dollars with only a few exceptions, then he can answer the question "is Bob's rental rate less than 400" with high success rate.

We use the probability distribution of $\mathsf{priv}(r)$ over $r \in \mathcal{X}$ to represent the intrinsic knowledge about priv:

$$p^{\mathsf{priv}} \triangleq \Pr[\mathsf{priv}(r) = 0 | r \leftarrow \mathcal{X}],$$

which we may simplify to p when understood. This knowledge is known to all adversaries. Take the experienced market analyst as example again, he knows that 98% of the rental rates are less than 400 dollar, that is, $\Pr[\mathsf{priv}(r) = 1] = 0.98$ for $r \in \mathcal{X}$. When he tries to predict the rental rate of a particular person in the database (say, Bob), suppose he does not know anything particular about Bob, he will refer to his intrinsic knowledge about priv (i.e., 98% farmers' rental rate is less than 400) and predicts that Bob has rental rate under 400, as most people do. This strategy is obviously better than random guessing. In this example, the probability of $\mathsf{priv}(r) = 1$ over all people plays an important role in predicting priv on a particular person.

A very important assumption in our discussion is that the database is large-scaled and it is equipped with real world data. Generally speaking, if a database enrolls enough samples from the entire population, the distribution of priv over the database would be close enough to that over the entire population, the latter is what we call the "intrinsic knowledge." Intrinsic knowledge is considered public to a well-informed adversary.

Information that is not yet known. Although a well-informed adversary has a good sense of common knowledge, he does not know anything about any specific record(s). As a counter example, suppose $\mathcal{A}$ knew his friend Bob was enrolled in the database $\mathcal{X}$, and he also knew Bob's rental rate is around 140 to 180 dollars (this fact is considered a personal secret, it is not *common* for every farmer); when given question "is Bob's rental rate greater than 140?" he was quite confident to answer *yes*, regardless of the rental rate distribution in $\mathcal{X}$. The second example is, the fact "area of zip code 43113 has rental rate from 200 to 350 dollar per acre" is not considered a common knowledge either, because this fact does not apply to all farmers. Actually, such knowledges are what $\mathcal{A}$ wants to learn in a data mining process. After he knows this kind of knowledge, he is able to make better prediction on $\mathsf{priv}(r)$. We required that a well-informed adversary only knows the general information over *all* records; he does not hold any specific information on any record(s), before he makes queries to the database. See Sect. 4 for more discussion on what information $\mathcal{A}$ has before and after the knowledge discovering process.

Defining Privacy Predicate. More formally, a well informed adversary $\mathcal{A}$ knows the public information $\mathsf{pub}(r)$ on a target record r, as well as the overall distribution of $\mathsf{priv}(r)$ over $r \in \mathcal{X}$. He wants to predict $\mathsf{priv}(r)$ on target r. We denote as $r \leftarrow \mathcal{X}$ that r is chosen from $\mathcal{X}$ at uniform random. Let $\Omega(\cdot)$ be the big Ω notation, denote $|\mathcal{X}|$ the number of records in the table. Let $p = \Pr[\mathsf{priv}(r) = 0 | r \leftarrow \mathcal{X}]$ (therefore, $1-p = \Pr[\mathsf{priv}(r) = 1 | r \leftarrow \mathcal{X}]$); p represents the distribution of priv over $r \in \mathcal{X}$. We define privacy predicate as one that no adversary can compute it better than random guessing, even they know $\mathsf{pub}(r)$ and p.

Definition 1 (Privacy Predicate). *A predicate* priv *is a privacy predicate if (1) it is efficiently computable from* r*, and (2) there exists a function in* $\Omega(|\mathcal{X}|)$ *such that for any well informed, probabilistic polynomial time adversary* $\mathcal{A}$*,*

$$\Pr_{r \leftarrow \mathcal{X}}[\mathcal{A}(\mathsf{pub}(r), p) = \mathsf{priv}(r)] \leq \max\{p, 1-p\} + \frac{1}{\Omega(|\mathcal{X}|)}. \tag{1}$$

Remark. If priv is a privacy predicate, no adversary can predict it better than random guessing, even if they know the public information of the target. That is, the success rate of any adversary given the public information $\mathsf{pub}(r)$ is less than $\max_{\mathcal{A}}\{\Pr_{r \leftarrow \mathcal{X}}[\mathcal{A}(p) = \mathsf{priv}(r)]\} = \max\{p, 1-p\}$, the best success rate in guessing $\mathsf{priv}(r)$, without $\mathsf{pub}(r)$. See "maximum success rate in guessing" for more discussion. It is required that $\mathsf{priv}(r)$ involves at least some bits in the sensitive fields, those bits are not predicable given $\mathsf{pub}(r)$.

Definition 1 only considers adversaries $\mathcal{A}(\mathsf{pub}(r), p)$ who never make query to the database before. After getting data mining result $m \leftarrow \mathcal{M}$ from method $\mathcal{M}$, an adversary $\mathcal{A}(\mathsf{pub}(r), p, m)$ gains new knowledge, then he makes a better judgment on $\mathsf{priv}(r)$. See Sect. 4 for more details.

Maximum Success Rate in Guessing. Given the distribution of a predicate priv over $\mathcal{X}$, the best strategy of guessing $\mathsf{priv}(r)$ (without any information of r, merely guessing) for a randomly chosen input is simply returning the majority bit b in $\{\mathsf{priv}(r) | r \in \mathcal{X}\}$, which yields a success rate equal to the ratio of b in all $\mathsf{priv}(r)$. It is not hard to see that other guessing strategies have lower success rate, that is,

$$\begin{aligned} &\max_{\mathcal{A}} \left\{ \Pr_{r \leftarrow \mathcal{X}}[\mathcal{A}(p) = \mathsf{priv}(r)] \right\} \\ =\ &\max_{b \in \{0,1\}} \left\{ \Pr_{r' \leftarrow \mathcal{X}}[\mathsf{priv}(r') = b] \right\} \\ =\ &\max\{p, 1-p\} \end{aligned} \tag{2}$$

that's because, if $\mathcal{A}$ does not know any information about r, he can only output a random bit. Suppose $\mathcal{A}$ outputs 0 with probability q, and 1 with probability $1-q$. By calculating the success rate of $\mathcal{A}$:

$$\begin{aligned} &\Pr[\mathcal{A} = \mathsf{priv}(r)] \\ =\ &\Pr[\mathcal{A} = 0, \mathsf{priv}(r) = 0] + \Pr[\mathcal{A} = 1, \mathsf{priv}(r) = 1] \\ =\ &q \cdot p + (1-q) \cdot (1-p) \\ =\ &1 - q - p + 2qp \\ =\ &p - (2p-1)(1-q) = (1-p) - q(1-2p) \end{aligned} \tag{3}$$

we have if $p > 0.5$ then he gets the maximum rate p with $q = 1$, otherwise if $p < 0.5$ then he gets the maximum $1-p$ with $q = 0$. That means, $\mathcal{A}$ will get the best success rate when it always returns the majority bit in $\{\mathsf{priv}(r) | r \in \mathcal{X}\}$.

Comments on Privacy Predicates and Hardcore Predicates. Readers with knowledge on cryptography may have been aware that the definition of a

privacy predicate is superficially similar to that of a hardcore predicate. Informally, public information function $\mathsf{pub}(r)$ is "one-wayed" in the sense that it is hard to compute r given $\mathsf{pub}(r)$[1], privacy predicate can be viewed as its "hardcore predicate." Indeed, hardcore predicate is a special form of privacy predicate. Assume that there exists a one-way function $f : \{0,1\}^n \mapsto \{0,1\}^*$ with a hardcore predicate $h : \{0,1\}^n \mapsto \{0,1\}$, then for any PPT adversary $\mathcal{A}$, and x sampled from the universe $\{0,1\}^n$ at uniform random, there exists a negligible function negl such that

$$\Pr_{x \leftarrow \{0,1\}^n} [\mathcal{A}(f(x)) = h(x)] < \frac{1}{2} + \mathsf{negl}(|x|). \tag{4}$$

Due to its definition, distribution of $\{h(x)\}$ is $50:50$ for $x \leftarrow \{0,1\}^n$, where each possible x is equally chosen. Compare Eq. (4) to Eq. (1), a hardcore predicate is indeed a privacy predicate with $p = 0.5$ and $\mathcal{X} = \mathcal{F} = \{0,1\}^n$. Notice that if $\mathcal{X} = \{0,1\}^n$, then $|\mathcal{X}| = 2^{|x|}$. If $\mathsf{negl}(|x|) = 2^{-|x|}$ (a typical choice of negligible function), then $\mathsf{negl}(|x|) = \frac{1}{|\mathcal{X}|} \in \frac{1}{\Omega(|\mathcal{X}|)}$, still satisfies Eq. (1).

However, secret predicate priv is not a hardcore predicate of pub. priv and pub's input are records sampled from $\mathcal{X}$, but the records in $\mathcal{X}$ is not necessarily evenly distributed over the universe $\mathcal{F}$: some values in $\mathcal{F}$ are never chosen to $\mathcal{X}$ and some may be chosen multiple times. As a result, the distribution of $\{\mathsf{priv}(r) | r \in \mathcal{X}\}$ is not $50:50$ for most priv.

In next section, we will define δ-privacy. Our intuition is, if $\mathcal{A}$'s success rate increases a lot after he gets a response from $\mathcal{M}$, we say $\mathcal{M}$ reveals too much information. δ-privacy is defined as even after the adversary gains new knowledge from data mining, his success rate in predicting $\mathsf{priv}(r)$ is not much higher (limited by δ) than before.

4 δ-privacy

We provide a new privacy definition for privacy preserving data mining, called δ-privacy. δ-privacy protects all partial secrets in the following sense: a well-informed adversary $\mathcal{A}$ makes a guess on $\mathsf{priv}(r)$; then he makes a query to the database, gets a response from a privacy preserving data mining method $\mathcal{M}$, from which he learns new knowledge about r; then he makes a new judgment on $\mathsf{priv}(r)$. The second prediction is supposed to have higher success rate compared with that before he makes a query; δ-privacy requires that the extra success rate is limited by δ, for any privacy predicate priv.

To better illustrate the above idea, we design a game between adversary $\mathcal{A}$ and a privacy preserving data mining method $\mathcal{M}$ as follow.

[1] If f is a one-way function, then given $f(x)$, it is hard to compute an x' such that $f(x') = f(x)$. But by definition of pub, given $\mathsf{pub}(r)$, it is not hard to find a $r' \in \mathcal{F}$ such that $\mathsf{pub}(r') = \mathsf{pub}(r)$, therefore, pub is not a one-way function. See Chap. 6 of [5] or Chap. 6 of [12] for rigorous definition of one-way functions and hardcore predicates.

Game 1. Game of Privacy Preserving Data Mining.

1: Target record $r \leftarrow \mathcal{X}$ is chosen at uniform random.
2: $\mathcal{A}$ is given the public information $\mathsf{pub}(r)$.
3: $\mathcal{A}$ chooses a privacy predicate priv.
4: $\mathcal{A}$ is given the probabilities $\{p, 1-p\}$ of distributions of $\mathsf{priv}(r')$ over $r' \in \mathcal{X}$.
5: $\mathcal{A}(\mathsf{pub}(r), p)$ predicts $\mathsf{priv}(r)$.
6: $\mathcal{A}$ makes query $\mathsf{q} = \mathsf{q}^{\mathsf{priv},\mathsf{pub}(r)}$.
7: $\mathcal{M}$ is given the query q and the database $\mathcal{X}$. It returns a result m to $\mathcal{A}$.
8: $\mathcal{A}(\mathsf{pub}(r), p, m)$ makes a new judgement on $\mathsf{priv}(r)$.

In the above game, the adversary gets the public information of r and the intrinsic knowledge of priv, we say he becomes a well informed adversary at step 4; he makes the first judgment at step 5. After that, he starts a knowledge discovering process, retrieves more information from the database. Finally, based on all information he receives in the game, he computes $\mathsf{priv}(r)$.

The success rate of prediction made in step 8 should not be much higher than that of the guess made in step 5; if the gap is large, which means $\mathcal{A}$ performs much better in computing $\mathsf{priv}(r)$ after he gets m, we say m discloses excessive information/privacy. The difference

$$\begin{aligned} &\Pr_{r\in\mathcal{X}}[\mathcal{A}(\mathsf{pub}(r), p, m) = \mathsf{priv}(r) | m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})] \\ -\ &\Pr_{r\in\mathcal{X}}[\mathcal{A}(\mathsf{pub}(r), p) = \mathsf{priv}(r)] \end{aligned}$$

is the extra success rate after $\mathcal{A}$ gets a response from $\mathcal{M}$. We also notice that before $\mathcal{A}$ makes any query to the database, the best possibility $\mathcal{A}(\mathsf{pub}(r), p)$ can achieve is $\max\{p, 1-p\}$, as shown in Eq. (1). We use the following difference to capture the probability gain (in successfully computing $\mathsf{priv}(r)$) after $\mathcal{A}$ discovers new knowledge from the database:

$$\begin{aligned} &\Pr_{r\in\mathcal{X}}[\mathcal{A}(\mathsf{pub}(r), p, m) = \mathsf{priv}(r) | m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})] \\ -\ &\max\{p, 1-p\}. \end{aligned} \tag{5}$$

Intuitively, this difference is the amount of privacy leaked to the adversary. For the purpose of protecting privacy, this difference should be small. δ-privacy is then defined as no adversary can gain extra success rate more than δ.

Definition 2 (δ-privacy). *Let* priv *be a privacy predicate. Suppose a computationally bounded adversary* $\mathcal{A}$ *is given* $\mathsf{pub}(r)$ *of some* $r \in \mathcal{X}$*, and let* $p = \Pr[\mathsf{priv}(r) = 0 | r \leftarrow \mathcal{X}]$ *denote the distribution of* $\mathsf{priv}(r)$ *over* $r \in \mathcal{X}$*.* $\mathcal{A}$ *makes query* q *and obtains response* m *from a privacy preserving data mining method* $\mathcal{M}$*.* $\mathcal{A}$ *outputs a bit to predict* $\mathsf{priv}(r)$*.* $\mathcal{M}$ *is said to preserving* δ*-privacy if there exists a positive real number* δ*, for any privacy predicate* priv*,*

$$\begin{aligned} &\Pr_{r\leftarrow\mathcal{X}}[\mathcal{A}(\mathsf{pub}(r), p, m) = \mathsf{priv}(r) | m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})] \\ \le\ &\max\{p, 1-p\} + \frac{1}{\Omega(|\mathcal{X}|)} + \delta, \end{aligned} \tag{6}$$

where, on the left hand side, the probability of $\mathcal{A}$ successfully computing $\mathsf{priv}(r)$ *depends on the random coin used by* $\mathcal{M}$.

Remark. For the ease of presentation, we sometimes may omit the term $1/\Omega(|\mathcal{X}|)$ in our discussion. Also, we only consider the case that $\delta \in (0, \frac{1}{2}]$, because if $\delta > \frac{1}{2}$, $\max\{p, 1-p\} + \delta$ exceeds 1, the formula will be always true.

Variants of δ-Privacy. The privacy Definition 2 introduces a absolute bound on the privacy leaks. This constraint can also be in other mathematical forms, the following variant introduces a relative bound, which depends on the intrinsic knowledge of the secret.

Definition 3 ***(δ-privacy-variant-I).*** *Let the notations be the same with those in Definition 2. The variant I of δ-privacy requires that for all computationally bounded adversaries in Game 1,*

$$\begin{aligned} &\Pr_{r \leftarrow \mathcal{X}}[\mathcal{A}(\mathsf{pub}(r), p, m) = \mathsf{priv}(r) | m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})] \\ \leq &\left(\max\{p, 1-p\} + \frac{1}{\Omega(|\mathcal{X}|)}\right) \cdot \exp(\delta). \end{aligned} \tag{7}$$

Dealing with Multiple Queries. Note that in Game 1, step 6 and 7 can be repeated multiple times, that is, we allow the adversary to make multiple queries, and to adaptively choose queries. If the queries are different, we can consider them as one big query, and that will be just the same with the single query case.

Another case is $\mathcal{A}$ makes the same queries for multiple times. As mentioned above, $\mathcal{M}(\mathsf{q}, \mathcal{X})$ is a random variable (r.v.). To learn the distribution of this r.v., adversaries may make the same query q for multiple times to get more samples. The more samples he gets, the more detailed he learns about the distribution, then he is able to use the distribution of $\mathcal{M}$ to predict $\mathsf{q}(\mathcal{X})$ and hence $\mathsf{priv}(r)$. There are three strategies to deal with multiple queries:

1. $\mathcal{M}$ will always return the same answer for the same query. An example for this is the anonymized table (Definition 10): once the anonymized table is published, adversaries run the query on the (deterministic) anonymized table, that is equivalent to a method $\mathcal{M}$ that always output the same answer to the same query.
2. $\mathcal{M}$ returns the same answer by maintaining a hash table, with the key being the input pair $\langle \mathsf{q}, \mathcal{X} \rangle$ and the value being the output m; when a request is made to $\mathcal{M}$, it first checks the hash table. If the request is new, $\mathcal{M}$ generates a new answer m, add it to the hash table and returns it; otherwise, it returns the existing answer. Such a method requires a large amount of storage space if there are many different queries and/or the database $\mathcal{X}$ is updated frequently. Also, this method may be not safe either: if $\mathcal{X}$ is updated frequently, and each update is small (such that the changes of $\mathsf{q}(\mathcal{X})$ is small in each update), adversaries may still have a good estimation on $\mathsf{q}(\mathcal{X})$.

3. $\mathcal{M}$ generates a new answer to every request. That means adversary is able to learn the distribution of $\mathcal{M}(\mathsf{q}, \mathcal{X})$. It implicitly requires that even if $\mathcal{A}$ knows the distribution of $\mathcal{M}(\mathsf{q}, \mathcal{X})$, he does not know $\mathsf{q}(\mathcal{X})$, or the relationship between $\mathsf{q}(\mathcal{X})$ and $\mathcal{M}(\mathsf{q}, \mathcal{X})$.

5 Tradeoff Between Privacy and Utility

In this section, we will discuss the relation between privacy and utility. Intuitively, the need for privacy is incompatible with the need for utility: to protect privacy, a data provider wishes to hide information, while a data user wishes to collect as much information as he could. We provide a quantitative analysis for the tradeoff between privacy and utility, showing that the amount of information gain is bounded by the largest possible privacy loss of the data provider. To get more information, the data user should negotiate with the data provider to increase the limit of privacy leak.

The utility of a data mining method is how much knowledge can be discovered from its results. If the result is generated by a privacy preserving data mining method $\mathcal{M}(\mathsf{q}, \mathcal{X})$, its utility should be measured by the "distance" to the true result $\mathsf{q}(\mathcal{X})$. More precisely, the output of $\mathcal{M}(\mathsf{q}, \mathcal{X})$ is a random variable; the utility is the closeness between the random variable and the true value $\mathsf{q}(\mathcal{X})$. Intuitively, if $\mathcal{M}(\mathsf{q}, \mathcal{X})$ is close to $\mathsf{q}(\mathcal{X})$, the result is likely to be useful, otherwise it is not.

Some literatures (e.g., [24]) employ the *entropy* of $\mathcal{M}(\mathsf{q}, \mathcal{X})$ to estimate utilities for perturbation-based methods. The idea behind is, the more "concentrated" are the results (which are drawn from $\mathcal{M}(\mathsf{q}, \mathcal{X})$, the closer they are to the real result $\mathsf{q}(\mathcal{X})$. However, this definition is correct if and only if the true answer is always at the "center" of the random variable, it does not apply to all methods.

Utility of a Data Mining Method. Before we define the utility of a method $\mathcal{M}$ on query q and database $\mathcal{X}$, we first introduce one kind of adversaries called *honest users* to determine the utility of a method. An honest user is a deterministic algorithm that, given the answer of $\mathsf{q}(\mathcal{X})$, he correctly computes a knowledge knlg, where $\mathsf{knlg} : \mathcal{F} \mapsto \{0, 1\}$. More specifically, we define an honest user $\tilde{\mathcal{A}} = \tilde{\mathcal{A}}^{\mathsf{knlg}}$ as an adversary who takes part in the Game 2. He would like to compute a knowledge predicate (a privacy predicate) knlg on some $r \in \mathcal{X}$. He is an honest user if he always wins the game, that is, $\tilde{\mathcal{A}}$ designs query $\mathsf{q} = \mathsf{q}^{\mathsf{knlg},\mathsf{pub}(r)}$, after he receives the true result of $\mathsf{q}(\mathcal{X})$, he returns one bit as result such that

$$\tilde{\mathcal{A}}(\mathsf{pub}(r), \mathsf{q}(\mathcal{X})) = \mathsf{knlg}(r) \tag{8}$$

for any $\mathcal{X}$, any choice of $r \in \mathcal{X}$, and any appropriately chosen knlg.

Existence of Honest Users. Note that such honest user and knlg may not always exist; if $\mathsf{pub}(r)$ is not unique, $\tilde{\mathcal{A}}$ may not win the game with probability 1. For example, assume that there are two records r and r' which have the same public information, but one's rental rate is 150 and the other is 250. Then $\tilde{\mathcal{A}}$ may only

Game 2. Game of Honest User.

1: A target record r is chosen uniformly at random from the database $\mathcal{X}$.
2: User $\tilde{\mathcal{A}}$ is given the public information $\mathsf{pub}(r)$.
3: User $\tilde{\mathcal{A}}$ chooses a knowledge predicate knlg.
4: Based on knlg, $\tilde{\mathcal{A}}$ designs query $\mathsf{q} = \mathsf{q}^{\mathsf{knlg},\mathsf{pub}(r)}$.
5: An oracle $\mathcal{O}$ responses to $\tilde{\mathcal{A}}$ with the true result of $\mathsf{q}(\mathcal{X})$.
6: $\tilde{\mathcal{A}}$ outputs a bit b.
7: User $\tilde{\mathcal{A}}$ wins the game if $b = \mathsf{knlg}(r)$.

have $\frac{1}{2}$ chance in computing knowledge predicate "does r's rental rate greater than 200", no matter how $\tilde{\mathcal{A}}$ designed the query $\mathsf{q}^{\mathsf{knlg},\mathsf{pub}(r)}$. We assume here that the public information $\mathsf{pub}(r)$ is unique in database $\mathcal{X}$, so that such intelligent adversaries always exist.

Loosely speaking, we can think of a honest user as a person who lives in a world that people trust each other, he will treat any result given to him as the true $\mathsf{q}(\mathcal{X})$ and computes $\mathsf{knlg}(r)$ from it. Now suppose we replace the oracle $\mathcal{O}$ by a privacy preserving method $\mathcal{M}$, $\tilde{\mathcal{A}}$ remains innocent. Game 2 becomes the Game 1 (on Page 10), with knlg being priv.

Definition 4 (*Utility of a Single Result* m). *Given a single result $m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})$, the utility of m with respect to a honest user $\tilde{\mathcal{A}} = \tilde{\mathcal{A}}^{\mathsf{knlg}}$ is defined by a function $u_{\tilde{\mathcal{A}}}$:*

$$u_{\tilde{\mathcal{A}}}(r, m) = \begin{cases} 1 \ if \tilde{\mathcal{A}}(\mathsf{pub}(r), m) = \mathsf{knlg}(r) \\ 0 \ otherwise \end{cases} \tag{9}$$

If an honest user $\tilde{\mathcal{A}}$ correctly computes knlg given m, we say that m is useful ($u_{\tilde{\mathcal{A}}}(m) = 1$), otherwise it is not ($u_{\tilde{\mathcal{A}}}(m) = 0$).

Given results from $\mathcal{M}(\mathsf{q}, \mathcal{X})$, $\tilde{\mathcal{A}}$ evaluates the utilities of single results, we define the utility of the method as the expectation of utilities.

Definition 5 (*Utility of* $\mathcal{M}(\mathsf{q}, \mathcal{X})$). *The utility (with respect to $\tilde{\mathcal{A}} = \tilde{\mathcal{A}}^{\mathsf{knlg}}$) of a method $\mathcal{M}$ on query q and data set $\mathcal{X}$ is defined as the expected utility of the random variable $\mathcal{M}(\mathsf{q}, \mathcal{X})$:*

$$\begin{aligned} &\mathcal{U}_{\tilde{\mathcal{A}}}(\mathcal{M}(\mathsf{q}, \mathcal{X})) \triangleq E[u_{\tilde{\mathcal{A}}}(r, m) | r \leftarrow \mathcal{X}, m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X})] \\ &= \sum_{r,m} u_{\tilde{\mathcal{A}}}(r, m) \times \Pr[r \leftarrow \mathcal{X}, \mathcal{M}(\mathsf{q}, \mathcal{X}) = m] \end{aligned} \tag{10}$$

Notice that the intrinsic knowledge $p = p^{\mathsf{knlg}}$ of knlg also provides some "utility": even without querying to $\mathcal{M}$, there are adversaries that can achieve some utility, which we refer to as intrinsic utility. Assume, w.l.o.g., that $p > 0.5$. We fix an adversary (denoted as $\mathcal{A}' = \mathcal{A}^{\mathsf{knlg}}$) which outputs 0 with probability 1 (that is, $\mathcal{A}'(r, m) = 0$ for all r and m). We define the intrinsic utility as the expected utility evaluated by $\mathcal{A}'$ (perhaps with a little abuse of notation):

Definition 6 (*Intrinsic Utility of* knlg). *The intrinsic utility of knowledge predicate* knlg *is defined as the expected utility evaluated by a user* $\mathcal{A}' = \mathcal{A}^{\mathsf{knlg}}$

$$\mathcal{U}_{\mathcal{A}'}(1) \triangleq E[u_{\mathcal{A}'}(r,1)|r \leftarrow \mathcal{X}] \tag{11}$$

The reason we employ the special user $\mathcal{A}'$ to define intrinsic utility is this user is the one that achieves the largest possible utility among all guessing users, see "maximum success rate in guessing", Page 8 for more information.

The intrinsic utility is the one that obtained by a user without querying to $\mathcal{M}$, we define the utility gain of a method $\mathcal{M}$ on q and $\mathcal{X}$ as the utility of $\mathcal{M}(\mathsf{q},\mathcal{X})$ (w.r.t. $\tilde{\mathcal{A}}^{\mathsf{knlg}}$) minus the intrinsic utility of knlg.

Definition 7 (Utility Gain of $\mathcal{M}(\mathsf{q},\mathcal{X})$). *The utility gain (w.r.t.* $\tilde{\mathcal{A}}^{\mathsf{knlg}}$*) of* $\mathcal{M}(\mathsf{q},\mathcal{X})$ *is the difference between the utility of the method minus the prior utility of* knlg*, that is,*

$$\mathcal{G}_{\tilde{\mathcal{A}}}(\mathcal{M}(\mathsf{q},\mathcal{X})) \triangleq \mathcal{U}_{\tilde{\mathcal{A}}}(\mathcal{M},\mathsf{q},\mathcal{X}) - \mathcal{U}_{\mathcal{A}'}(1), \tag{12}$$

the (value of) utility of $\mathcal{M}(\mathsf{q},\mathcal{X})$ *depends on the choice of* $\tilde{\mathcal{A}}$*.*

The utility gain reflects how much information is gained from the data mining method. Intuitively, the larger is the gain, the better is the result. However, we show the following theorem, saying that the largest possible utility gain of a δ-private method, can not exceed the amount of privacy loss, namely, δ.

Theorem 1. *If* $\mathcal{M}$ *is* δ*-private, then*

$$\mathcal{G}_{\tilde{\mathcal{A}}}(\mathcal{M}(\mathsf{q},\mathcal{X})) \leq \delta + \frac{1}{\Omega(|\mathcal{X}|)}$$

for all q*, all* $\mathcal{X}$*, all* knlg *and all* $\tilde{\mathcal{A}}$*.*

Remark. We can achieve a similar result for δ-privacy-variant-I (but in different mathematics form). This theorem is an important result that shows the relation between privacy and utility: they are zero-summed in the sense that the utility gain is always less than the amount of largest possible privacy leak. Suppose a user wants to learn some knowledge knlg, he wants to get Δ utility gain versus his intrinsic knowledge, he have to ask his data providers to increase the privacy leak limit to at least Δ.

The Theorem also depicts a framework for potential privacy trading. Data providers can consider "selling" their private data, the price is the largest possible privacy leak; and data processors can purchase those data based on how much information they can learn from them.

6 The Relations Between δ-privacy and Other Existing Privacy Definitions

In this section, we will discuss the relationship between δ-privacy and existing privacy definitions, namely, differential privacy and data anonymity. We will show that differential privacy (with parameter δ) implies δ-privacy-variant-I, and t-closeness w.r.t. variational distance implies δ-privacy with $t = \delta$.

6.1 A Supporting Lemma

We will show a lemma used in the proof Lemmas 2 and 3. Reader may proceed to the next sections for now and come back for this lemma later.

We will show that if two random variables $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are *close*, then the r.v.'s generated by running $\mathcal{A}$ on $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are still close.

Denote $\mathcal{A}(\mathcal{Y})$ the binary-valued random variable generated by randomized algorithm $\mathcal{A}$, whose input is sampled from discrete random variable $\mathcal{Y}$. We argue that if the distance between two random variables $\mathcal{Y}_1$ and $\mathcal{Y}_2$ is less than δ, then $\mathcal{A}(\mathcal{Y}_1)$ and $\mathcal{A}(\mathcal{Y}_2)$ also have distance less than δ. On the other hand, if for all algorithm $\mathcal{A}$, $\mathcal{A}(\mathcal{Y}_1)$ and $\mathcal{A}(\mathcal{Y}_2)$ are close, then $\mathcal{Y}_1$ and $\mathcal{Y}_2$ must be close to each other.

A very important question arises: how to define the distance of two random variables? There are many way to define it, for example, variational distance, Kullback-Leibler distance, Earth Mover's Distance, just to name a few. In this paper, we use the variational distance to define statistical distance of two random variables.

Definition 8 (Variational Distance of Two Random Variables *[15]*). *Suppose* $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *are probability distribution functions (p.d.f.) of two discrete random variables (with respect sample space* Ω_1 *and* Ω_2*). The two random variables are statistically close if for some* $\delta > 0$*,*

$$\sup_{S \subseteq \Omega_1 \cup \Omega_2} |\mathbf{P}_1(S) - \mathbf{P}_2(S)| \leq \delta.$$

If the random variables are discrete ones, the above formula becomes

$$\max_{S \subseteq \Omega_1 \cup \Omega_2} \left\{ \sum_{y \in S} (\mathbf{P}_1(y) - \mathbf{P}_2(y)) \right\} \leq \delta. \tag{13}$$

Variation distance is the maximum difference between the probabilities that two p.d.f.'s can assign to the same event.

Lemma 1. *Suppose* $\mathcal{A}$ *is a randomized algorithm whose output is one bit,* $\mathcal{Y}_1$ *and* $\mathcal{Y}_2$ *are two discrete random variables with sample space* Ω*.* $\mathcal{A}(\mathcal{Y}_1)$ *and* $\mathcal{A}(\mathcal{Y}_2)$ *satisfy*

$$\Pr[\mathcal{A}(\mathcal{Y}_1) = b] \leq \Pr[\mathcal{A}(\mathcal{Y}_2) = b] + \delta \tag{14}$$

for $b = 0, 1$ *if and only if the variational distance of* $\mathcal{Y}_1$ *and* $\mathcal{Y}_2$ *are less than* δ*, i.e., for any subset* $S \subseteq \Omega$*,*

$$\Pr[y \in S | y \leftarrow \mathcal{Y}_1] \leq \Pr[y \in S | y \leftarrow \mathcal{Y}_2] + \delta. \tag{15}$$

Lemma 1 will be used to show the relationship between δ-privacy and differential privacy, and between δ-privacy and t-closeness. Note that even if Eq. (14) is true for both $b = 0$ and 1, it does not imply that

$$\Pr[\mathcal{A}(y) = f(y) | y \leftarrow \mathcal{Y}_1] \leq \Pr[\mathcal{A}(y) = f(y) | y \leftarrow \mathcal{Y}_2] + \delta$$

is true for all binary valued function f.

Corollary 1. *Let algorithm $\mathcal{A}$, random variables $\mathcal{Y}_1$ and $\mathcal{Y}_2$, sample space Ω be the ones defined in Lemma 1. The random variables $\mathcal{A}(\mathcal{Y}_1)$ and $\mathcal{A}(\mathcal{Y}_2)$ satisfy*

$$\Pr[\mathcal{A}(\mathcal{Y}_1) = b] \leq \Pr[\mathcal{A}(\mathcal{Y}_2) = b] \times \exp(\delta) \tag{16}$$

if and only if for any set $S \subseteq \Omega$,

$$\Pr[y \in S | y \leftarrow \mathcal{Y}_1] \leq \Pr[y \in S | y \leftarrow \mathcal{Y}_2] \times \exp(\delta). \tag{17}$$

6.2 Relation to Differential Privacy

In this section, we will show the relation between differential privacy and δ-privacy. If a privacy preserving data mining method is differentially private (with parameter ϵ), then it is δ-private-variant-I with parameter $\delta = \epsilon$. But on the other hand, a method satisfies δ-privacy does not necessarily mean it satisfies δ-differential privacy. That means, in terms of the maximum privacy leak, δ-differential privacy is more secure than δ-privacy, which, by Theorem 1, gives less flexibility in getting utility.

Differential privacy [6] is one of the most influential definition of privacy, which is widely adopted by output perturbation methods. It captures the idea that the result of the query should not leak information of any person, as if the target person were not included in the table.

Definition 9 (differential privacy *[6]*). *An algorithm $\mathcal{M}$ is said to satisfy ϵ-differential privacy if for all database $\mathcal{X}$ and $\mathcal{X}_1$ which only differ in one record, and for any set S of possible outcomes,*

$$\begin{aligned} &\Pr[m_1 \in S | m_1 \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X}_1)] \\ &\leq \Pr[m \in S | m \leftarrow \mathcal{M}(\mathsf{q}, \mathcal{X}) \in S] \times \exp(\epsilon). \end{aligned} \tag{18}$$

Lemma 2. *If $\mathcal{M}$ satisfies ϵ-differential privacy, it also satisfies ϵ-privacy-variant-I.*

Remarks. Lemma 2 shows that if a method is ϵ-differential privacy, then it is ϵ-private-variant-I, however, the reverse is not necessarily true: a method is δ-private is not necessarily δ-differentially-private.

6.3 Relation to Data Anonymity

Data anonymity is family of privacy preserving data mining methods. In this section, we will show that if a data anonymity method is t-close w.r.t. variational distance (which is the strongest requirement in the series), it is δ-private with $\delta = t$.

Instead of answering to queries, data anonymity methods publish a modified table for everyone to query on. Records in the database are grouped into equivalence classes sharing a same identifying tag, their secret fields remain unchanged.

Using the (original) public information of a target record, an adversary can locate the target to a specific class by the tags on the classes, but he does not know which one in the class is the target, since everyone in the group looks the same.

It is required that in each equivalence class, the distribution of the sensitive data is close to that of the entire table, otherwise, if for some class, the distribution is greatly different from that of the entire table, an adversary can conclude that targets in this class have special properties that other records do not have.

Definition 10 (t-closeness [13]). *An equivalence class is t-close to the whole table if the* distance *between the following two distributions is no more than a threshold t: the distribution of sensitive data in that equivalence class, and the one in the whole table. An anonymized table is t-close to the original table if all equivalence classes are t-close to the whole table.*

The definition of t-closeness does not specify how to measure the divergence of two distributions. In the following lemma, we will show that data anonymity methods satisfies t-closeness w.r.t. variational distance (Definition 8), then it is δ-private:

Lemma 3. *A privacy preserving method $\mathcal{M}$ publishes an anonymized table with equivalence classes, it is δ-private if the variational distance between the distribution of sensitive data in any equivalence class, and the distribution of sensitive data in the whole table, is no more than δ.*

7 Conclusion

In this paper, we proposed δ-privacy, a new privacy definition for privacy preserving data mining. By analyzing adversaries' behaviors, δ-privacy directly tells the data provider the risk of privacy leak in a data mining process. We show that existing data privacy analysis mechanisms are also compatible to ours; actually, our method can be applied to any data mining process, not limited to particular ones. Another important contribution of our work is, we mathematically shows that the amount of information extracted from a data mining process does not exceed the amount of possible privacy loss; this idea seems not surprising, but to the best of our knowledge, we are the first one to rigorously prove this conjecture. A potential application for this is, we can develop a pricing system for "data trading," data providers could sell their data, get compensated based on their (possible) privacy loss; on the other hand, data users could purchase data based on the amount of information they could get. This trading system is possible under our δ-privacy framework.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. SIGMOD Rec. **29**(2), 439–450 (2000). http://doi.acm.org/10.1145/335191.335438
2. Brenner, H., Nissim, K.: Impossibility of differentially private universally optimal mechanisms. In: FOCS, pp. 71–80. IEEE Computer Society (2010)
3. Brickell, J., Shmatikov, V.: The cost of privacy: destruction of data-mining utility in anonymized data publishing. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 70–78. ACM, New York (2008)
4. Cormode, G., Procopiuc, C., Shen, E., Srivastava, D., Yu, T.: Empirical privacy and empirical utility of anonymized data. In: 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), pp. 77–82, April 2013
5. Delfs, H., Knebl, H.: Introduction to Cryptography - Principles and Applications. Information Security and Cryptography. Springer, Heidelberg (2007)
6. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). doi:10.1007/978-3-540-79228-4_1
7. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). doi:10.1007/11681878_14
8. Dwork, C., Pottenger, R.: Toward practicing privacy. J. Am. Med. Inform. Assoc. **20**(1), 102–108 (2013). http://jamia.bmj.com/content/20/1/102.abstract
9. Ganta, S.R., Kasiviswanathan, S.P., Smith, A.: Composition attacks and auxiliary information in data privacy. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 265–273. ACM, NY, USA (2008). http://doi.acm.org/10.1145/1401890.1401926
10. Ghosh, A., Roughgarden, T., Sundararajan, M.: Universally utility-maximizing privacy mechanisms. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, STOC 2009, pp. 351–360. ACM, NY, USA (2009). http://doi.acm.org/10.1145/1536414.1536464
11. Gupte, M., Sundararajan, M.: Universally optimal privacy mechanisms for minimax agents. In: Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, pp. 135–146. ACM, NY, USA (2010). http://doi.acm.org/10.1145/1807085.1807105
12. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman & Hall/Crc Cryptography and Network Security Series. Chapman & Hall/CRC, Boca Raton (2007)
13. Li, N., Li, T.: t-closeness: Privacy beyond k-anonymity and -diversity. In: Proceedings of IEEE 23rd International Conference on Data Engineering (ICDE 2007) (2007)
14. Li, T., Li, N.: On the tradeoff between privacy and utility in data publishing. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pp. 517–526. ACM, NY, USA (2009). http://doi.acm.org/10.1145/1557019.1557079
15. Lin, J.: Divergence measures based on the shannon entropy. IEEE Trans. Inform. Theory **37**(1), 145–151 (1991)
16. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–54. Springer, Heidelberg (2000). doi:10.1007/3-540-44598-6_3

17. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: L-diversity: privacy beyond k-anonymity. In: Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, p. 24 (2006)
18. McSherry, F., Mironov, I.: Differentially private recommender systems: Building privacy into the net. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 627–636. ACM, NY, USA (2009). http://doi.acm.org/10.1145/1557019.1557090
19. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, pp. 19–30. ACM, NY, USA (2009). http://doi.acm.org/10.1145/1559845.1559850
20. Parra-Arnau, J., Rebollo-Monedero, D., Forn, J.: Measuring the privacy of user profiles in personalized information systems. Future Gener. Comput. Syst. **33**, 53–63 (2014). http://www.sciencedirect.com/science/article/pii/S0167739X1300006X, special Section on Applications of Intelligent Data and Knowledge Processing Technologies; Guest Editor: Dominik lzak
21. Peters, F., Menzies, T., Gong, L., Zhang, H.: Balancing privacy and utility in cross-company defect prediction. IEEE Trans. Softw. Eng. **39**(8), 1054–1068 (2013)
22. Rebollo-Monedero, D., Parra-Arnau, J., Diaz, C., Forn, J.: On the measurement of privacy as an attackers estimation error. Int. J. Inf. Secur. **12**(2), 129–149 (2013). http://dx.doi.org/10.1007/s10207-012-0182-5
23. Sweeney, L.: K-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(5), 557–570 (2002). http://dx.doi.org/10.1142/S0218488502001648
24. Venkatasubramanian, S.: Measures of anonymity. In: Aggarwal, C.C., Yu, P.S. (eds.) Privacy-Preserving Data Mining. ADBS, vol. 34. Springer, Boston (2008). doi:10.1007/978-0-387-70992-5_4