Attacks on Privacy and deFinetti's Theorem

Daniel Kifer
Penn State University

ABSTRACT

In this paper we present a method for reasoning about privacy using the concepts of exchangeability and deFinetti's theorem. We illustrate the usefulness of this technique by using it to attack a popular data sanitization scheme known as Anatomy. We stress that Anatomy is not the only sanitization scheme that is vulnerable to this attack. In fact, any scheme that uses the random worlds model, i.i.d. model, or tuple-independent model needs to be re-evaluated.

The difference between the attack presented here and others that have been proposed in the past is that we do not need extensive background knowledge. An attacker only needs to know the nonsensitive attributes of one individual in the data, and can carry out this attack just by building a machine learning model over the *sanitized* data. The reason this attack is successful is that it exploits a subtle flaw in the way prior work computed the probability of disclosure of a sensitive attribute. We demonstrate this theoretically, empirically, and with intuitive examples. We also discuss how this generalizes to many other privacy schemes.

Categories and Subject Descriptors

H.1 [**Models and Principles**]: Miscellaneous—*Privacy*

General Terms

Security

1. INTRODUCTION

Many organizations are in possession of data sets that they would like to release to the public. In many cases, such data sets also contain sensitive information which should not be revealed. Examples include GIC, which collected health insurance data for Massachusetts state employees [46]; AOL, which collected search log data from its users [6]; and Netflix, which collected movie ratings from its customers [39].

While the release of such data can benefit both the public and the organizations themselves (through a better understanding of medical data, better information retrieval technology, and better collaborative filtering algorithms), there is a fear that sensitive information about individuals in the data will be revealed. This fear is well-founded, since inadequate data sanitization allowed the identification of the medical records of the governor of Massachusetts in the GIC data [46]; it allowed the identification (and subsequent interview) of an AOL user by reporters from the New York Times [6]; and it allowed for the potential identification of Netflix subscribers based on posts in blogs and newsgroups [40].

With such motivation, many schemes for publishing sanitized data have been proposed. Informally, we say that a data set has been *sanitized* if it is impossible or very difficult for an attacker to infer sensitive information from the data. Difficulty could either result from a high average-case computational complexity [18] or from the amount of extra information an attacker needs to collect in order to breach privacy [35, 36, 10]. Thus it is clear that when designing a method for sanitizing data, one should also reason about attacks available to an attacker.

While many proposed sanitization schemes rely solely on the perceived complexity of their data transformations, there has been a growing body of work that investigates strategies an attacker may use to breach privacy. These include linking attacks [46, 21], exploitation of properties of the sanitization algorithm [48, 20], use of background knowledge [35, 36, 10, 23, 5] and reasoning about how an attacker's prior belief changes into a posterior belief [19, 37, 35, 42].

In this paper we present an attack using data mining techniques that are based on a deep statistical theorem known as deFinetti's representation theorem [45]. Using this representation theorem, we show potential vulnerabilities in sanitization schemes that are based on random worlds [35, 36, 10] and schemes that assume the attacker believes the data are generated i.i.d. from a distribution P (that is known to the attacker) [19] and schemes that assume the attacker believes that each tuple t_i is placed in the database independently of other tuples with probability p_i (also known to the attacker) [37, 42]. As a proof of concept, we will show how to exploit this vulnerability in a privacy scheme known as Anatomy [49] and we will present experimental results supporting our claim. We stress, however, that Anatomy is not the only scheme with this vulnerability. We chose to illustrate our attack on Anatomy because Anatomy is easy to explain, because it requires an attack algorithm that we consider to be interesting, and because there is insufficient space to present an attack for every vulnerable sanitization

ACM SIGMOD '09 Providence, RI, USA

scheme that has been proposed. The main idea of the attack is to build a Bayesian network, such as a Naive Bayes classifier, with a twist: instead of predicting the sensitive attribute of a new individual, we predict the sensitive attribute of an individual that was part of the training set (i.e. the sanitized data).

The outline of this paper is as follows. In Section 2 we introduce our notation. In Section 3 we explain the i.i.d. model, the random worlds model, and the tuple independent model for reasoning about privacy; we discuss their drawbacks, and we define and advocate the use of *exchangeability* (which is the focus of deFinetti's Representation Theorem) for reasoning about privacy. In Section 4, we will discuss partition-based sanitization schemes, of which Anatomy is an example. In Section 5 we will discuss how to attack partition-based schemes and present a specific algorithm for attacking Anatomy (note that because of differences in sanitization schemes, the general attack methodology is the same, but the specific algorithms are different). In Section 6, we show the experimental results from our attack. In Section 7 we will discuss related work and point out a few more sanitization schemes that we believe are vulnerable, and we will present conclusions in Section 8.

2. NOTATION

We begin with some notation. Let $\mathcal{D}$ be a database relation with an attribute S that is considered to be sensitive[1] and attributes $R_1, \ldots, R_n$ which are considered to be non-sensitive. For example, S may be the disease of an individual while R_1 might be the height. We will abuse notation and let S and $R_1, \ldots, R_n$ also denote the corresponding domains of the attributes. Let D be an instance of relation $\mathcal{D}$ with k tuples and let $\mathcal{A}$ be a sanitization algorithm (possibly randomized) which converts D into $\mathcal{A}(D)$, a sanitized data set. In privacy preserving data publishing, the goal is to choose a sanitization algorithm $\mathcal{A}$ such that $\mathcal{A}(D)$ is safe to release (i.e. releasing $\mathcal{A}(D)$ will not violate the privacy of individuals whose tuples are in D).

We define *privacy* generically in terms of changes of belief in order to explain the broad applicability of a data mining attack based on exchangeability and deFinetti's theorem.

DEFINITION 1 (PRIVACY). *Let δ be a measure of distance between probability distributions, let b be a real number, and let $P_T(x.S)$ be the prior beliefs of attacker T about the sensitive attribute of x. A sanitization scheme $\mathcal{A}$ maintains the privacy of x against T if $\delta(P_T(x.S), P_T(x.S|\,\mathcal{A}(D))) < b$.*

For the case of Anatomy and related work [35, 36, 10, 49], we will specialize this definition to model an attacker in an initial state of ignorance (with a uniform prior):

DEFINITION 2 (PRIVACY). *For each $s_i \in S$, let b_i be some number between 0 and 1. An sanitization scheme $\mathcal{A}$ maintains privacy if an attacker cannot use $\mathcal{A}(D)$ infer that $P(x.S = s_1) > b_i$ for some sensitive value s_i and some individual x whose tuple appears in D.*

There are two things to note about Definitions 1 and 2: we did not specify how an attacker performs inference and we did not require the attacker to guess the true sensitive value $x.S$ of some individual x.

Clearly we should not allow any arbitrary inference system (such as a crystal ball) because many such systems lack credibility. The acceptability of an inference system should be judged by the privacy community as a whole. The random worlds model, the i.i.d. model, and the tuple-independent model (all to be discussed in Section 3) have been considered to be reasonable and so should not be abandoned. However, they do not sufficiently protect privacy and so should be supplemented with additional reasoning systems (such as those presented in this paper).

Also, an attacker does not have to correctly guess the true value $x.S$ in order to cause harm to x. For instance, suppose the attacker decides that $x.S$ =AIDS with probability 0.9. Even if x is perfectly healthy, the disclosure of such a statement can be harmful to x if the attacker is able to convince a sizable set of people that the inference procedure was reasonable. Thus random worlds, the i.i.d. model, and tuple-independent model should not be discarded since on the surface they seem reasonable. However, as we will show in this paper, a more sophisticated attacker can make a better inference.

3. MODES OF REASONING

In this section we discuss three modes of reasoning that are common in sanitization schemes: the random worlds model, the i.i.d. model, and the tuple-independent model. We will point out the inadequacies of these models, explain deFinetti's Representation Theorem, and advocate its use for reasoning about privacy.

3.1 Random Worlds model, IID model, Tuple-Independent model

The *random worlds* model [4] is commonly used to reason about attackers who do not have probabilistic opinions about the data[2] but ostensibly are willing to learn [35, 36, 10, 49]. Initially, by appealing to the principle of indifference, the attacker believes that all instances of $\mathcal{D}$ with k tuples are equally likely (technically, each assignment of attribute values to an individual is considered equally likely). Each instance of $\mathcal{D}$ corresponds to a possible world and the attacker does not know which is the real world. Thus to compute $P(x.S$ =AIDS) from the sanitized data $\mathcal{A}(D)$, the attacker will examine the instances D' for which $\mathcal{A}(D')$ equals $\mathcal{A}(D)$ and will compute the fraction of such instances in which $x.S$ =AIDS. We note that [35] also used a more complicated version of the random worlds model and that it has the same drawbacks as the i.i.d. model, which we discuss next. For this reason we omit further discussion of the more complex version of random worlds.

In the *i.i.d. model*, the attacker believes that the tuples were generated identically and independently from a data-generating distribution P (which is known to the attacker). This is the model that is used, for example, in *gamma amplification* and *$\rho_1 - to - \rho_2$ privacy breaches* [19]. The justification for this is that somehow the attacker has learned what P is (we disagree with the plausibility of this claim in Section 3.3). Note that in some cases the P that the attacker believes in does not have to be the "true" distribution. The common theme of sanitization methods based on the i.i.d.

[1] The restriction to one sensitive attribute is made purely for ease of explanation.

[2] Their opinions are expressed in propositional logic.

Tuple ID	Smoker?	Lung Cancer?
1	n	n
2	n	n
⋮	⋮	⋮
98	n	n
99	n	n
100	n	n
101	y	y
102	y	y
⋮	⋮	⋮
198	y	y
199	y	y
200	y	?

Table 1: A data set related to smoking

model of reasoning is that it does not matter which specific P is chosen by the attacker.

The *tuple-independent model* is commonly used in probabilistic databases [11] and has also been used to evaluate sanitization schemes related to *perfect privacy* [37] and $\alpha\beta$ *anonymization* [42]. Here it is assumed that the attacker believes each tuple t_i has probability p_i of appearing in a database instance and the appearance of tuple t_i is independent of the appearance of tuple t_j for $j \neq i$. The p_i for each tuple are considered to be known by the attacker. This model is more general than the i.i.d. model since the tuples do not have to be identically distributed.

Despite their appealing nature, these 3 reasoning schemes have a weakness which at first seems counter-intuitive: *they place severe restrictions on an attacker's ability to learn.* We illustrate this idea with a simple example here (we will then show how this relates to privacy breaches with a more detailed example based on the sanitization scheme known as Anatomy [49] in Section 4).

Consider Table 1. This table has 200 tuples and, aside from the tuple id, contains two binary attributes: whether an individual smokes and whether an individual has lung cancer. Note that all information is present in the table except for the lung cancer status of tuple 200.

If we use random worlds to reason about tuple 200, we would start with the belief that every table of 200 tuples is equally likely. After seeing Table 1, we would conclude that only two tables are now possible: one where tuple 200 has lung cancer and one where tuple 200 does not have lung cancer. Furthermore, we would consider both tables to be equally likely, so we would conclude that tuple 200 has lung cancer with probability 0.5. This is, in fact, the same probability we would have given before we had seen Table 1.

Now let us use the i.i.d. model. This requires us to select $p_1 = P(\text{smoker} \wedge \text{lung cancer})$, $p_2 = P(\text{nonsmoker} \wedge \text{lung cancer})$, $p_3 = P(\text{smoker} \wedge \text{no lung cancer})$, and $p_4 = P(\text{nonsmoker} \wedge \text{no lung cancer})$ before seeing the data. Since we believe that tuples are generated i.i.d. by the probability distribution P, we would reason that since we know tuple 200 is a smoker, the probability that tuple 200 has lung cancer is $p_1/(p_1+p_3)$. Note that this is the same probability we would have given before we had seen Table 1 if we had known that tuple 200 was a smoker.

For the tuple-independent model we need to select $p_1^{(x)} = P(\text{id}= x \wedge \text{smoker} \wedge \text{lung cancer})$, $p_2^{(x)} = P(\text{id}= x \wedge \text{nonsmoker} \wedge \text{lung cancer})$, $p_3^{(x)} = P(\text{id}= x \wedge \text{smoker} \wedge \text{no lung cancer})$, and $p_4^{(x)} = P(\text{id}= x \wedge \text{nonsmoker} \wedge \text{no lung cancer})$. After seeing the data, we would conclude that the missing value is lung cancer with probability $p_1^{(200)}/(p_1^{(200)} + p_3^{(200)})$ which again is the same probability we would have given before we had seen Table 1 if we had known that tuple 200 was a smoker.

In all three cases, even though there appears to be a strong correlation between smoking and lung cancer in the population from which the table was sampled, neither the i.i.d. model nor the random worlds model nor the tuple-independent model accounted for it (without specifying it exactly in advance). In other words, the table did nothing to change our beliefs. On the other hand, it seems reasonable that our estimate of the probability that tuple 200 has lung cancer should increase after seeing such data because we should learn about the correlation between attributes.

3.2 Exchangeability and deFinetti's Theorem

The error in reasoning (for random worlds, the i.i.d. model, and the tuple-independent model) is very subtle. The error, it turns out, is that all three models assume that the tuples are independent of each other *and* that we believe they are generated by a particular distribution P. In fact, if we don't commit to a specific probability distribution P, then the apparent paradox can be avoided. To better understand this, we first introduce the concept of exchangeability [45] and then describe the representation theorem of deFinetti.

DEFINITION 3 (EXCHANGEABILITY). *A sequence X_1, $X_2, \ldots$ of random variables is exchangeable if every finite permutation of these random variables has the same distribution.*

For example, the flips of a coin are exchangeable: the probability of seeing $HHHTT$ is the same as the probability of seeing $THTHH$, no matter what the bias of the coin is. Furthermore, every i.i.d. sequence of random variables is exchangeable. However, exchangeability is more general.

Consider the following scenario. There are two biased coins; the first coin lands heads with probability 1 and the second coin lands tails with probability 1. A game show host then selects one of two coins at random with equal probability (without telling us which one it is) and proceeds to generate a sequence of k coin flips. Given a sequence of coin flips, each permutation of this sequence is equally likely so the coin flips generated by this procedure are exchangeable. It is also easy to see that the coin flips are not independent: if the result of the first flip is a heads, the result of the second flip must also be a heads. Thus from the first coin flip, we learn more about the coin and thus we are able to better predict the second coin flip. On the other hand, if we had known which coin was selected, then we would learn nothing new from the first coin flip. Thus if we had known the selected coin, then we would have considered the coin flips to be i.i.d., but since we do not know which coin was selected, then after every coin flip we learn more about this coin and this affects our beliefs about future flips. Therefore we would not consider the coin flips to be i.i.d. (this is, in fact, what allows us to avoid the situation in Section 3.1 where the attacker would not change his beliefs despite overwhelming evidence to the contrary).

deFinetti's Representation Theorem generalizes this simple example to essentially an arbitrary sequence of exchangeable random variables.

THEOREM 1 (DEFINETTI'S THEOREM [45]). *Let $\{X_n\}_{n=1}^{\infty}$ be an exchangeable sequence of random variables on a Borel space $(\mathcal{X}, \mathcal{B})$. Then there is a unique random probability measure $\mathcal{P}$ over $(\mathcal{X}, \mathcal{B})$ such that conditioned on $\mathcal{P} = P$, the X_i are distributed i.i.d with distribution P.*

deFinetti's theorem states that an exchangeable sequence of random variables is mathematically the same as choosing a data-generating distribution at random and then generating the data as independent draws from this distribution. In our example, the set of coins represents the set of data-generating distributions. The game show host chooses one of these data-generating distributions (i.e. coins) at random and creates the data by flipping the selected coin repeatedly.

This is easily generalized to an infinite number of possible coins. To each coin we associate a *bias*, which is a number between 0 and 1 and represents the particular coin's probability of landing heads. Since we don't know which coin (and associated bias) is selected, we can place a uniform distribution over the biases. Thus the game show host selects a bias uniformly at random and then proceeds to flip the corresponding coin to generate data.

In the general case, we view the generation of data from an exchangeable sequence of random variables as a two-step process: first we select the parameters of a probability distribution (such as the bias of a coin) from a prior probability over parameters and then, using these parameters, we generate the data (such as coin flips). Based on the data, we can use Bayesian reasoning to compute the posterior distribution of the parameters. For example, after seeing many heads, we consider the coins with high bias to be more likely to be the true coin that generated the data. Thus each coin flip gives us more information about the true coin.

We give a concrete example of this using Table 1. We can treat $p_1 = P(\text{smoker} \wedge \text{lung cancer})$, $p_2 = P(\text{nonsmoker} \wedge \text{lung cancer})$, $p_3 = P(\text{smoker} \wedge \text{no lung cancer})$, and $p_4 = P(\text{nonsmoker} \wedge \text{no lung cancer})$ as unknown parameters with a uniform prior. Thus any choice of (p_1, p_2, p_3, p_4) for which $0 \leq p_1 \leq 1, 0 \leq p_2 \leq 1, 0 \leq p_3 \leq 1, 0 \leq p_4 \leq 1$ and $p_1 + p_1 + p_2 + p_4 = 1$ is equally likely. The probability of randomly selecting a choice of parameters and then generating Table 1 is:

$$\frac{1}{3!} \int_{\substack{0 \leq p_1 \leq 1, 0 \leq p_2 \leq 1 \\ 0 \leq p_3 \leq 1, 0 \leq p_4 \leq 1 \\ p_1 + p_2 + p_3 + p_4 = 1}} p_1^{100} p_2^0 p_3^0 p_4^{100} dp_1\ dp_2\ dp_3\ dp_4$$

$$+ \frac{1}{3!} \int_{\substack{0 \leq p_1 \leq 1, 0 \leq p_2 \leq 1 \\ 0 \leq p_3 \leq 1, 0 \leq p_4 \leq 1 \\ p_1 + p_2 + p_3 + p_4 = 1}} p_1^{99} p_2^0 p_3^1 p_4^{100} dp_1\ dp_2\ dp_3\ dp_4$$

where the first integral corresponds to the probability of seeing Table 1 with the last tuple having lung cancer and the second integral corresponds to the probability of seeing the table with the last tuple not having lung cancer. Evaluating these integrals results in the probability of the table being:

$$\frac{1}{3!} \frac{100!100!}{203!} + \frac{1}{3!} \frac{99!100!}{203!}$$

with the first term representing the probability of seeing Table 1 with the last tuple having lung cancer and the second term corresponding to the probability of seeing the table with the last tuple not having lung cancer. Thus *after we have seen the table* we update our probabilistic estimate that the last tuple in Table 1 has lung cancer to the following value:

$$\frac{\frac{1}{3!} \frac{100!100!}{203!}}{\frac{1}{3!} \frac{100!100!}{203!} + \frac{1}{3!} \frac{99!100!}{203!}} = \frac{100}{101}$$

On the other hand, because of the uniform prior over parameters, our belief *before seeing the table* that the last tuple had lung cancer would have been 0.5. Thus, in contrast to the random worlds, tuple-independent, and i.i.d. models, exchangeability allowed us to learn about the correlation between smoking and lung cancer for the population represented by Table 1.

An interesting aspect of the tuple-independent model is that it adds a new wrinkle to the analysis. According to this model, the tuples in a relation instance are not identically generated but if we were to collect additional relation instances $D_1, D_2, \ldots$ (where D_1 is our current relation instance) then these instances would be independent and identically distributed. Thus we can view the sequence of relation instances $D_1, D_2, \ldots$ to be exchangeable by first parametrizing the p_i (probabilities that tuple i appears) so that different tuples can share some of the parameters and then by placing a prior distribution over these parameters. Thus we would believe that the p_i are first chosen (without being revealed to us) and then are used to generate a relation instance. By virtue of the p_i (and the parameter values) being unknown to us, the tuples in a relation instance are suddenly correlated from our point of view, but if we were told the true p_i values then the correlation disappears (this is the essence of exchangeability!). If, based on some tuples, we learn that smoking is correlated with lung cancer, we would believe that our estimate of the p_i for other tuples that are smokers and have lung cancer should increase relative to our estimate of the p_i for tuples that are smokers and do not have lung cancer. Thus the appearance of some tuples changes our belief about the appearance of other tuples. While Miklau and Suciu [37] noted that known positive correlations between tuples will affect their analysis of perfect privacy, they did not consider the fact that correlations may be induced simply by lack of definite knowledge of the true values of the p_i.

To summarize, what we have described is the Bayesian approach to inference. In fact, deFinetti's theorem is the cornerstone of this approach. It also shows that the previous approaches to reasoning about privacy were degenerate: an attacker only considered one possibility as the true data generating distribution. Such an attacker could not change his mind about the bias of the coin even after seeing 1 million consecutive heads or 1 million consecutive tails. In this sense, attackers considered in prior work were very obstinate. In contrast, those attackers who use deFinetti's theorem are willing to learn about the data generating distribution. We will illustrate the use of exchangeability and deFinetti's theorem for reasoning about privacy (and attacking sanitized data) in Section 5.

3.3 Is Perfect Knowledge the Worst-Case?

Having discussed exchangeability and deFinetti's theorem, we would like to address another concern about the use of the i.i.d. model for reasoning about privacy. Much work assumes that the attacker knows the true data-generating distribution. This is usually justified in one of two ways: this distribution can be learned from similar data, or this

scenario represents the worst-case of what an attacker can infer about an individual. We disagree with both justifications for the following reasons.

First, there is no such thing as the "true data-generating distribution". It is a Platonic ideal, like the circle, which is useful for data modeling and thus can act as a model for an attacker's beliefs. However, a privacy definition which relies on an attacker learning the true value of something that does not exist is hard to justify philosophically. Second, learning anything resembling a "true data-generating distribution" requires a similar dataset. However, what we learn from two similar datasets will not be exactly the same. For example, a hospital in Pennsylvania and a hospital in California may produce similar data, but there will be differences due to random variations (the same phenomenon that makes it unlikely that two sequences of coin flips will be identical) and sources of bias such as differences in diets and pollution levels in each state. Thus if an attacker chose to model the data probabilistically, there would still be uncertainty about which probabilities to use and this should be reflected in the attack: an attacker should not act as if the probabilities are completely certain (as in the case of the i.i.d. model).

We also believe that attackers who act as if they know the true distribution (e.g. by using the i.i.d. model) are not more dangerous than attackers who do not. The reason that is the latter kind of attacker would update his or her data model using information from the sanitized data (the first kind of attacker will not) and will then use it to make inferences about the target individual. Thus the beliefs of the second attacker could change more than the beliefs of the first attacker. This larger change from prior to posterior beliefs represents a greater threat to privacy.

Thus we believe that attackers based on exchangeability are more realistic when reasoning about privacy than are attackers based on the random worlds, tuple-independent, and i.i.d. models[3]. Nevertheless, those models should not be discarded because an attacker does not have to be rational or statistically sophisticated to cause harm - an attacker only needs to be convincing. Thus these models should also be used in the evaluation of probabilistic privacy breaches.

4. ANATOMY AND PARTITION-BASED SANITIZATION SCHEMES

In this section we discuss the class of partition-based sanitization schemes and we give an overview of Anatomy, which is a particular instance of such a scheme. We will discuss how probabilities of attribute disclosure are currently estimated in such schemes and then in Section 5 we will present an algorithm that predicts probabilities more accurately.

Let D be an instance of a relation with k tuples, a sensitive attribute S (such as disease), and n nonsensitive attributes $R_1, \ldots, R_n$ (such as demographic information). For example, Table 2 shows an instance of a relation with 12 tuples and the sensitive attribute "Disease".

A *partition-based* sanitization scheme partitions the tuples into disjoint groups and publishes certain statistics about each group. For example in the *generalization* model, which is also known as *global recoding*, used by k-anonymity [46],

[3]The i.i.d model is really a special case of exchangeability where the prior over probability distributions is a point mass.

Tuple ID	Gender	Age	Zip Code	Disease
1	M	25	90210	AIDS
2	F	43	90211	AIDS
3	M	29	90212	Cancer
4	M	41	90213	AIDS
5	F	41	07620	Cancer
6	F	40	33109	Cancer
7	F	40	07620	Flu
8	F	24	33109	None
9	M	48	07620	None
10	F	40	07620	Flu
11	M	48	33109	Flu
12	M	49	33109	None

Table 2: Original Table

Tuple ID	Gender	Age	Zip Code	Disease
1	*	25-49	9021*	AIDS
2	*	25-49	9021*	AIDS
3	*	25-49	9021*	Cancer
4	*	25-49	9021*	AIDS
5	*	25-49	0762*	Cancer
7	*	25-49	0762*	Flu
9	*	25-49	0762*	None
10	*	25-49	0762*	Flu
6	*	25-49	3310*	Cancer
8	*	25-49	3310*	None
11	*	25-49	3310*	Flu
12	*	25-49	3310*	None

Table 3: Global Recoding

each group must have at least k tuples and the domain of each nonsensitive attribute is coarsened. An example of a 4-anonymous table created using generalizations is shown in Table 3. In this table, the zip code has been coarsened by replacing the last digit with a $*$, age has been coarsened into intervals of length 25, and gender has been suppressed.

More flexible versions of generalizations, collectively known as *local recoding*, have also been proposed (see, for example, [32, 22, 2]). In local recoding, the domain of the nonsensitive attributes can be coarsened in a different way for each group. An example of a table created by local recoding is shown in Table 4. Note that age is coarsened in different ways in each group.

The original motivation behind such schemes is that if an attacker knows the nonsensitive attributes of an individual in the table, the attacker cannot be certain which tuple belongs to that individual. Such an attacker would not be able to identify the tuple belonging to an individual in Tables 3 and 4 with resolution better than a group of size 4. Note that this is little consolation to individuals with tuple id's 1,2,3, and 4. They appear in a group with 3 AIDS patients and 1 cancer patient. Any individual known to be in this group is likely to have AIDS, even if the tuple corresponding to the individual cannot be exactly identified.

Machanavajjhala et al. [35] have shown formally that if an attacker knows that a target individual corresponds to tuples 1,2,3 or 4 in Table 3, and if the attacker uses the random worlds reasoning model, then the attacker will conclude that this individual has AIDS with probability 3/4 (essentially because 3 out of 4 of the tuples in this group must have AIDS). Similarly, if an attacker knows that a target individual corresponds to tuples 1,2,3, or 4 in Table 4 and reasons using random worlds, then the attacker will conclude that the target individual has AIDS with probability

Tuple ID	Gender	Age	Zip Code	Disease
1	*	25-45	9021*	AIDS
2	*	25-45	9021*	AIDS
3	*	25-45	9021*	Cancer
4	*	25-45	9021*	AIDS
5	*	40-50	07620	Cancer
7	*	40-50	07620	Flu
9	*	40-50	07620	None
10	*	40-50	07620	Flu
6	*	20-50	33109	Cancer
8	*	20-50	33109	None
11	*	20-50	33109	Flu
12	*	20-50	33109	None

Table 4: Local Recoding

Tuple ID	Gender	Age	Zip	GID
1	M	25	90210	1
3	M	29	90212	1
7	F	40	07620	1
8	F	24	33109	1
2	F	43	90211	2
5	F	41	07620	2
9	M	48	07620	2
10	F	40	07620	2
4	M	41	90213	3
6	F	40	33109	3
11	M	48	33109	3
12	M	49	33109	3

GID	Disease
1	AIDS
1	Cancer
1	Flu
1	None
2	AIDS
2	Cancer
2	None
2	Flu
3	AIDS
3	Cancer
3	Flu
3	None

Table 5: Quasi-identifier and Sensitive Tables

3/4. Thus Machanavajjhala et al. proposed a family of privacy definitions known as ℓ-diversity which, based on the random worlds model, place restrictions on the frequencies of various sensitive attributes in each group of the partition (an alternative restriction on frequencies called t-closeness [33] has also been proposed, but we are not aware of any reasoning models associated with it).

Xiao et al. [49] observed that queries over the sanitized data can be answered with better accuracy if the domains are not coarsened. Instead they proposed a scheme called Anatomy which disassociates the sensitive attributes from the non-sensitive attributes and relies on random worlds reasoning for privacy. The output of anatomy consists of two tables: a quasi-identifier table (Table 5, on the left) and a sensitive table (Table 5, on the right). The quasi-identifier table contains the entire nonsensitive attribute information, in addition to a group id GID (when tuples are partitioned into groups, a unique GID is assigned to each group). The sensitive table contains the sensitive values that appear in a particular group. Collectively, these two tables are known as the *anatomized table*. Anatomy is essentially a lossy join decomposition using the group id. Thus in group 1, it is not possible to determine whether tuple 1 has AIDS, cancer, flu, or no disease. Note that in each group, every sensitive value is different and so the group size is the same as the parameter ℓ in ℓ-diversity. Using the random worlds model, this means that tuple 1 has AIDS with probability 0.25, cancer with probability 0.25, flu with probability 0.25, and no disease with probability 0.25.

It has been believed by many that, in terms of privacy, the only difference between Anatomy and approaches based on generalization is that Anatomy only makes it easier for an attacker to determine whether an individual is in the table or not (since nonsensitive information can frequently be used to uniquely identify an individual [46]). Once an attacker knows that an individual is in a particular group, it was believed that the attacker's inference from anatomized tables or generalized tables would be the same (i.e. count the fraction of tuples with AIDS in the group to determine probability of AIDS). However, we will show in Section 5 that this extra nonsensitive information can allow an attacker to compute better probabilities.

Tuple ID	Smoker?	GID
1	y	1
2	y	1
3	n	2
4	n	2
5	y	3
6	n	3
7	y	4
8	y	4
9	n	5
10	n	5
11	y	6
12	n	6

GID	Disease
1	Cancer
1	Flu
2	Flu
2	None
3	Cancer
3	None
4	Cancer
4	None
5	Flu
5	None
6	Cancer
6	None

Table 6: Quasi-identifier and Sensitive Tables

5. VULNERABILITIES OF ANATOMY

In this section we begin with an example that illustrates a vulnerability in Anatomy. We will then formally show how to exploit this vulnerability in Section 5.1 and then we will discuss vulnerabilities in other partition-based sanitization schemes in Section 5.2.

Consider Table 6. It shows the quasi-identifier table and the sensitive table from an anatomized version of an (unknown) original data set that came from a (fictitious) hospital. The nonsensitive attribute records whether or not an individual smokes and the sensitive attribute is the individual's disease. Note that each group has size 2. If an attacker wanted to estimate the probability that tuple 12 in group 6, a non-smoker has cancer, then the random worlds model suggests this probability is 0.5 because exactly half of the diseases in group 6 are cancer.

However, an attacker who is willing to learn may make the observation that whenever a smoker is in a group, then cancer is one of the diseases that appear in the group (this occurs in groups 1,3,4, and 6). On the other hand, groups composed entirely of nonsmokers do not contain cancer (this is true of groups 2 and 5). Thus an attacker may reason that the appearance pattern of smoking and cancer in this table is evidence of a correlation between smoking and cancer in the hospital's population. Therefore, even though tuple 12 (a non-smoker) appears in a group where one individual has cancer and the other is healthy, the probability that tuple 12 does have cancer should be less that 0.5. This is because tuple 11, a smoker in the same group, is more likely to have cancer according to the correlation exhibited by this table. In fact, we computed the probability of tuple 12 having cancer and tuple 11 being healthy to be approximately 0.16 (details omitted due to lack of space), which is significantly lower than the random worlds estimate.

Thus we see that the random worlds approach does not allow for learning about correlations between the sensitive and nonsensitive attributes because it does not allow for learning about one group from another group. On the other hand, common sense tells us that there is a leakage of information

between groups: the correlation structure in the rest of the table provides us with information about group 6.

5.1 An Attack Algorithm

We will use the phrase *deFinetti Attack* to refer to the general class of attacks that build a statistical model and place a prior distribution over its parameters. In this section, we will present a particular instance of a deFinetti attack, including an algorithm. This attack will correspond to an attacker with no prior knowledge but with an ability to learn.

The main idea is to model the correlations between sensitive and nonsensitive attributes using a Bayesian network. Exchangeability and deFinetti's representation theorem will be invoked by placing a prior distribution over the parameters of this network (see Section 3.2). We will choose a uniform so that we simulate an attacker who is in a state of ignorance but is willing to learn. The attack algorithm is presented in Algorithms 1 and 2. Throughout this section we assume that the attribute values are all discrete. Now we discuss the derivation.

The Bayesian network we will discuss here and experimentally evaluate in Section 6 is known as Naive Bayes. We chose it for two reasons. First, it is easy to explain. Second, it sometimes performs remarkably well in other machine learning tasks especially when the amount of data is limited (the data sanitization performed by Anatomy essentially decreases the effective data size because of the lossy join decomposition). A common justification for this phenomenon is that the errors made by Naive Bayes can be attributed to its bias (i.e. the assumptions it makes) and variance (the accuracy with which it can estimate parameter values using limited data). More complex Bayesian networks can decrease bias by adding more parameters and hence will increase variance. For limited amounts of data it is believed that the decrease in variance for Naive Bayes outweighs its increase in bias and results in smaller errors [15].

For a tuple t with nonsensitive attributes $R_1, \ldots, R_n$ and sensitive attribute S, Naive Bayes models the probabilities as:

$$P(t.R_1 = r_1, \ldots, t.R_n = r_n, t.S = s) = P(t.S = s) \prod_{i=1}^{n} P(t.R_i = r_i \mid t.S = s) \quad (1)$$

which means that it considers the nonsensitive attributes to be conditionally independent of each other given the value of the sensitive attribute. The parameters of this model are $P(t.S = s)$ for all s in the domain of S, and $P(t.R_i = r_i \mid t.S = s)$ for all i, for all r_i in the domain of R_i and for all s in the domain of S.

To employ exchangeability and deFinetti's theorem, we put a prior over the parameters. To model an attacker who is in a state of ignorance but is willing to learn, we make this prior a uniform prior. Thus, for example $P(t.S)$ is equally likely to be any probability distribution over the sensitive attribute S, and for each i and s, $P(t.R_i|t.S = s)$ is equally likely to be any (conditional) probability distribution over the values of the attribute R_i given s. After seeing the data, some of these distributions will be more likely than others.

Now, let t be the tuple corresponding to a target individual whose nonsensitive attribute values are known to us: $t.R_1 = r_1, \ldots, t.R_n = r_n$. Let T be the output of Anatomy

Algorithm 1 Attack

Require: target tuple t
Require: sensitive value s
1: Use Knuth shuffle to select a permutation π that assigns sensitive values in a group to tuples in that group.
2: **for** $i = 1$ to num_outer_iterations **do**
3: Resample Naive Bayes parameters using the distributions in Equations 4 and 5
4: $\pi \leftarrow$ SamplePermutation(π)
5: results$[i] = \pi$
6: **end for**
7: counter $\leftarrow 0$
8: **for** $i = \frac{\text{num\_outer\_iterations}}{2}$ to num_outer_iterations **do**
9: $\pi' \leftarrow$results$[i]$
10: **if** π' assigns value s to $t.S$ **then**
11: counter $\leftarrow$ counter $+1$
12: **end if**
13: **end for**
14: return $\frac{counter}{\text{num\_outer\_iterations}/2}$ as estimate of $P(t.S = s \mid T)$

Algorithm 2 SamplePermutation

Require: π = current permutation
1: **for** each group id *gid* **do**
2: **for** $i = 1$ to num_inner_iterations **do**
3: Use Knuth shuffle to select a permutation $\pi'(gid)$ for sensitive values in group *gid*.
4: $f \leftarrow f(\pi(gid))$ (using Equation 7)
5: $f' \leftarrow f(\pi'(gid)$ (using Equation 7)
6: $p = \min\{1, f'/f\}$
7: With probability p, $\pi(gid) \leftarrow \pi'(gid)$
8: **end for**
9: (Optional: Resample Naive Bayes parameters from distributions in Equations 4 and 5 using the updated permutation)
10: **end for**
11: return π

on the original data (which contains t). Let $t_1, \ldots, t_{k-1}$ denote the rest of the tuples and let $t_i.R_j = \alpha_j^{(i)}$ be their corresponding nonsensitive attribute values (this information is provided by the quasi-identifier table). Let $T^{s'}$ be the quasi-identifier and sensitive tables obtained from T by removing tuple t from the quasi-identifier table and by removing the sensitive value s' from the group containing t in the sensitive table. Informally, $T^{s'}$ represents the rest of the sanitized data after $t.S$ has been assigned value s'. We will use $\pi^{s'}$ to represent any permutation which, for each group of $T^{s'}$, assigns the sensitive values present in that group to tuples in that group (we will need to perform a summation over all such permutation). Let $\pi_i^{s'}$ represent the sensitive value that is assigned to tuple t_i by permutation $\pi^{s'}$.

Since our goal is to attack target individual t, we are interested in the value of $P(t.S = s \mid T)$. First,

$$P(t.S = s \mid T) = \frac{P\left(T^s \ \wedge \ t.S = s \ \bigwedge_{i=1}^{n} t.R_i = r_i\right)}{\sum_{s' \in S} P\left(T^{s'} \ \wedge \ t.S = s' \ \bigwedge_{i=1}^{n} t.R_i = r_i\right)} \quad (2)$$

Next,

$$P\Big(T^{s'} \ \wedge \ t.S = s' \ \bigwedge_{i=1}^{n} t.R_i = r_i\Big)$$
$$= \sum_{\pi^{s'}} P\Big(t.S = s' \ \bigwedge_{i=1}^{n}(t.R_i = r_i)$$
$$\bigwedge_{j=1}^{k-1}\Big[t_j.S = \pi_j^{s'} \bigwedge_{i=1}^{n} t_j.R_i = \alpha_i^{(j)}\Big]\Big)$$
$$= \int \sum_{\pi^{s'}} P(t.S = s') \prod_{i=1}^{n} P(t.R_i = r_i | t.S = s') \times$$
$$\prod_{j=1}^{k-1} P(t_j.S = \pi_j^{s'}) \prod_{i=1}^{n} P(t_j.R_i = \alpha_i^{(j)} | t_j.S = \pi_j^{s'}) dP \quad (3)$$

where the summation is over all permutations (which, for each group of $T^{s'}$, assign the sensitive values present in that group to tuples in that group) and where dP represents the uniform distribution over the model parameters.

Thus to compute the probability that the target individual has value s for the sensitive attribute, we need to substitute Equation 3 into Equation 2. Unfortunately, this probability cannot be determined analytically. Furthermore, the integral in Equation 3 appears to be intractable for the following reason: the problem of counting the number of perfect matchings in a bipartite graph is easily reducible to the computation of the summation inside this integral. Counting the number of perfect matchings in bipartite graphs is known to be #P-complete [47].

Despite this difficulty, there is good news. Even though computing the probabilities in Equation 3 (and therefore also in Equation 2) may be hard in general, it does not mean that it is hard for every instance of an anatomized table. The standard approach[4] to solving these kinds of problems is to use Markov Chain Monte Carlo methods [44]. More specifically, we will use a Gibbs sampler outer loop (Algorithm 1) with a Metropolis-Hastings inner loop (Algorithm 2).

The Gibbs sampler outer loop works as follows. We initially select a permutation π (which, for each group of T, assigns the sensitive values present in that group to tuples in that group) uniformly at random. This can be done using the Knuth shuffle [28]. Such a permutation assigns sensitive values to tuples. Temporarily treating this as the "true" assignment, we can easily compute the posterior distribution of the Naive Bayes parameters:

$$P(t.S) \sim Dir(1 + n_{s_1}, \ldots, 1 + n_{s_{|S|}}) \quad (4)$$
$$P(t.R_i \mid t.S = s) \sim Dir(n^{\pi}_{i,x_1,s} + 1, \ldots, n^{\pi}_{i,x_{|R_i|},s} + 1) \quad (5)$$

where $|S|$ is the size of the domain of sensitive attribute S; $|R_i|$ is the size of the domain of attribute R_i; $s_1, \ldots, s_{|S|}$ are the possible values of attribute S; $x_1, \ldots, x_{|R_i|}$ are the possible values of attribute R_i; n_s is the total number of times that sensitive value s appears in the sensitive table; $n^{\pi}_{i,x,s}$ is the number of tuples whose value for attribute R_i is x and which are assigned sensitive value s by the permutation π; and Dir is the Dirichlet distribution [8] (the notation $X \sim Dir(\ldots)$ means that the random variable X is distributed according to the distribution $Dir(\ldots)$). Note that Equation 5 represents a separate probability distribution for each i and $s \in S$.

We sample new Naive Bayes parameters from these distributions. These new Naive Bayes parameters now induce a probability distribution over permutations. The probability of a permutation π' is proportional to:

$$P(t.S = \pi'_t) \prod_{i=1}^{n} P(t.R_i = r_i | t.S = \pi'_t)$$
$$\times \prod_{j=1}^{k-1} P(t_j.S = \pi'_j) \prod_{i=1}^{n} P(t_j.R_i = \alpha_i^{(j)} | t_j.S = \pi'_j) \quad (6)$$

where π'_t is the sensitive value that π' would assign to the target tuple t and π'_j is the sensitive value that π' would assign to the tuple t_j. Using the Metropolis-Hastings inner loop (Algorithm 2) we sample a new random permutation π using this probability distribution. Then we resample the Naive Bayes parameters according to the distributions in Equations 4 and 5, sample a new permutation π, resample the Naive Bayes parameters (based on this π), and so on. We repeat this procedure until this Markov chain has almost converged. Techniques for assessing convergence are discussed in [44].

Each iteration of this Gibbs sampler outer loop gives us a new permutation. Supposing there are M iterations, we have a set $\pi_1, \ldots, \pi_M$ of M permutations. We throw away the first $M/2$ permutations as these are generated before the Markov chain has neared convergence (also known as the burn-in period). Then to compute $P(t.S = s|T)$ we simply compute the fraction of the permutations $p_{M/2}, \ldots, p_M$ that assign sensitive value s to tuple t [5]. Note that using this procedure it is possible to simultaneously compute $P(t.S = s|T)$ for multiple target tuples by computing the appropriate fraction of the permutations.

The Gibbs sampler outer loop requires us to sample a permutation according to the current value of the parameters sampled from the distribution in Equations 4 and 5. This is done using a Metropolis-Hastings approach. A permutation π is composed of one permutation for each group. Let $\pi(gid)$ be the permutation corresponding to the group with group id gid. The score associated with $\pi(gid)$ is defined as:

$$f(\pi(gid)) = \prod_{\{j \ | \ t_j \in \text{Group } gid\}} P(t_j.S = \pi(gid)_j)$$
$$\times \prod_{i=1}^{n} P(t_j.R_i = \alpha_i^{(j)} \mid t_j.S = \pi(gid)_j) \quad (7)$$

where $\pi(gid)_j$ is the sensitive value assigned to tuple t_j in Group gid. For each group, we perform many iterations of the following steps (until approximate convergence of the resulting Markov chain [44]): compute $f(\pi(gid))$ for the current permutation, use the Knuth shuffle to select a candidate permutation $\pi'(gid)$ for this group, compute $f(\pi'(gid))$, then replace $\pi(gid)$ with $\pi'(gid)$ with probability $\min\{1, \frac{f(\pi'(gid))}{f(\pi(gid))}\}$. The current permutation at the last

[4]The EM algorithm [13] is not recommended here because the resulting distribution is multimodal. Even if EM manages to find a maximum likelihood estimator, it has little meaning if other local maxima are almost as good.

[5]It is common to throw out data generated by some iterations after the burn-in period. This practice is called *thinning* and is not required because of the Ergodic theorem. See also [44], Lemma 12.2.

iteration is then returned. The entire attack algorithm is summarized in Algorithms 1 and 2.

5.2 Other partition-based schemes

While we provided full details for an attack on Anatomy, it is not our intention to claim that Anatomy is bad while other partition-based schemes are good. Thus in this section we sketch possibilities for attacking other partition-based sanitization schemes such as global and local recoding.

For global recoding, the construction of the Naive Bayes model as described in Section 5.1 will not work mainly because it has no information that helps to discriminate between certain attribute values. For example, if we globally generalize age into the age ranges $[0-5]$, $[6-10]$, $[11-15]$, $[16-45]$, $[46-50]$, and $[51-55]$, we will not learn how ages 16 and 45 differ in their correlations with the sensitive attributes because the available information is too coarse (we cannot distinguish between ages 16 and 45 from the sanitized data). However, by using semantic similarity between attribute values, we can overcome this problem. We may postulate that the effect of age x on the sensitive attribute (such as disease) is similar to the effect of age $x+1$. We can model such similarity by using, for example, flexible bayes [25] (which augments Naive Bayes with a kernel density estimate). The end result is that if we have some groups with age ranges such as $[46-50]$ and $[51-55]$ with higher incidence of heart disease than the rest of the groups, we may learn that age is correlated with heart disease. Now suppose there is a group with age range $[16-45]$ which contains 3 tuples, one of which has heart disease. Suppose also that the target individual is in this group and is known to be 45 years old. This new model will essentially allow us to reason that the other two individuals in the group are probably younger and therefore less likely to have heart disease than the target individual. Thus we may be able to predict probabilities better than the random worlds model.

Local recoding is clearly in between global recoding and Anatomy in terms of how much information about the nonsensitive attributes is hidden. Thus we might see one group with age range $[0-10]$, another with age range $[9-12]$, and another with age range $[11-15]$. Because of the overlaps, we may be able to distinguish slightly between the effects of ages 9 and 10 on the sensitive value even without using fancier models such as flexible bayes [25]. Again, this can let us predict probabilities better than the random worlds model. We believe that algorithms that strive to minimize information loss metrics for local recoding (such as Mondrian [32] and space-filling curve techniques [22]) make it possible to compute even better probabilities. Thus we believe that lower information loss in the nonsensitive attributes does indeed come at the expense of privacy in the sensitive attributes.

6. EXPERIMENTS

In this section we present experiments demonstrating the effectiveness of our attack against Anatomy. We implemented the attack code in Python and ran it on a machine with an Intel dual core 3.16GHz processor with 4GB main memory. The data used came from the Adult Dataset from the UCI Machine Learning Repository [3].

First, we removed all tuples with missing values from the data. This resulted in a data set with 30162 tuples. We retained the attributes *workclass*, *relationship*, *gender*, *salary class* (whether it is above or below \$50K), and *occupation*. We treated occupation as the sensitive attribute, and it had 14 distinct values.

We generated anatomized tables with 2 tuples per group, 3 tuples per group, and 4 tuples per group (since 4 does not evenly divide 30162, an anatomized table with 4 tuples per group also has a few groups of size 5). Thus these tables satisfy ℓ-diversity for $\ell = 2, 3, 4$, respectively. For each anatomized table, we ran our attack algorithm 20 times, each time there was a different initial starting point (line 1 of Algorithm 1). This allowed us to monitor convergence of the algorithm since similar results from each of the 20 runs despite different starting points was an indication of convergence. We ran the outer loop in Algorithm 1 50,000 times. In the case of Algorithm 2, we were able to directly sample permutations for groups of size 2, 3, and 4 (by computing the exact probability of each permutation). For larger group sizes we ran the inner loop of Algorithm 2 until there were 100 successful assignments in line 7 (typically this required over 2,000 iterations of the inner loop).

Attacking anatomized tables with 2 tuples per group took roughly 7.5 hours per run, attacking anatomized tables with 3 tuples per group took roughly 14 hours per run, and attacking anatomized tables with 4 tuples per group took 2 days per run. For very large group sizes, an asymptotic approach could also be possible. In any case, we note that the high computational cost would not be much of a deterrent to an attacker. First, our code was not especially optimized for speed and we believe that the number of iterations was a very conservative choice. Furthermore an attacker has little time pressure after release of the "sanitized" data (a month or more of computational time seems reasonable), and the possibility of algorithmic breakthroughs cannot be ruled out.

To evaluate the success of our attack, we selected 1000 tuples at random and tried to predict their sensitive values from the anatomized tables. To measure success, we used three metrics: absolute error (ABS), sum-squared error (SSQ), and classification accuracy (ACC). Absolute error is defined as $\sum_{i=1}^{1000}\sum_{j=1}^{14}|s_{ij}-p_{ij}|$, where s_{ij} is 1 if the true sensitive value of tuple i is the value j and 0 otherwise, and p_{ij} is the predicted probability that tuple i has sensitive value j. Sum-squared error is defined as $\sum_{i=1}^{1000}\sum_{j=1}^{14}|s_{ij}-p_{ij}|^2$. For accuracy, we take the prediction for a tuple to be the sensitive value with highest predicted probability. The accuracy is then the fraction of times the predicted value equals the true value.

For each metric we measured its minimum and maximum values over 20 runs. We also measured the value of the metric on probabilities generated by pooling the samples from those 20 runs. As a baseline, we measured the absolute error, sum-squared error, and accuracy that would result from the random worlds model applied to Anatomy. Our results are summarized in Table 7.

First note that for each performance metric and for each anatomized table (corresponding to groups of size 2, 3, and 4), the minimum and maximum values of each run are clustered close together around the value of the metric evaluated on the pooled set of runs. This is a good indication of convergence for the attack algorithm.

The absolute (ABS) and sum-squared errors (SSQ) in predicting probabilities are lower for the deFinetti attack than for the baseline corresponding to random worlds reasoning.

Group Size	Measurement	Performance Metric		
		ABS	SSQ	ACC
2	Min	532.34	318.32	0.766
	Max	532.79	318.66	0.785
	Pooled	532.57	318.47	0.770
	Baseline	1000.00	500.00	0.500
3	Min	968.03	572.30	0.568
	Max	968.70	572.81	0.579
	Pooled	968.28	572.53	0.576
	Baseline	1333.33	666.67	0.333
4	Min	1243.24	746.17	0.400
	Max	1243.89	746.80	0.408
	Pooled	1243.63	746.51	0.406
	Baseline	1500.00	750.00	0.250

Table 7: Performance of the deFinetti attack

Note that as the group sizes increase, the difference between the deFinetti attack's errors and the baseline's errors decreases. This is a natural effect with two causes. First, the number of choices for the sensitive attribute increases, making it harder to determine which is the correct one. Second, as group sizes increase, the number of groups decrease and thus there are fewer chances to learn about individuals in one group from information in the other groups. The difference in sum-squared error between the deFinetti attack and the baseline almost disappears when groups have size 4. Since the other metrics still show a marked difference, this implies that the sum-squared error, which is commonly used for measuring discrepancies in probabilities, is not a good choice[6].

In terms of accuracy, the deFinetti attack clearly outperforms the baseline, with an improvement of 25% for groups of size 2, 23% for groups of size 3, and 15% for groups of size 4. Even though the accuracy dips below 50% on groups of size 4, *this does not mean that groups of size 4 are safe.* For roughly 11% of the tuples, the deFinetti attack assigned a probability of at least 0.8 to a sensitive value. For 67% of those tuples, that value was correct. This illustrates that even at larger group sizes, there are likely to be tuples at risk. It also shows that the Naive Bayes model used by this instance of the deFinetti attack was overly optimistic in assigning probabilities (this is a well-known problem for Naive Bayes in general [43]). Thus, while not perfect, it still gives much better results than the random worlds reasoning model that is currently in use.

7. RELATED WORK

There has been a tremendous amount of work on partition-based privacy schemes. One of the earliest approaches relevant here was the work of Sweeney [46] who combined the privacy definition k-anonymity with generalizations (global recoding). Since then there has been work on improving algorithms for finding k-anonymous tables [31, 7] and using local recoding techniques which create sanitized data with more information content [2, 32, 22].

k-Anonymity does not provide strong guarantees on privacy. This was discussed in detail by Machanavajjhala et al. [35], who provided a different privacy definition, known as ℓ-diversity that was applicable to partition-based schemes. They claimed that many algorithms for k-anonymity can be easily retrofitted to support ℓ-diversity. Ghinita et al. [22] and Xiao et al. [49] also provided custom algorithms for ℓ-diversity.

Other privacy definitions for partition-based schemes have also been provided. These include (c, k)-safety [36], privacy skyline [10], and t-closeness [33]. The latter definition did not provide formal privacy guarantees, but ℓ-diversity, (c, k)-safety, and privacy skyline came with privacy guarantees that provide bounds on the inference an attacker can make if the attacker has propositional knowledge about the individuals in the data and if the attacker uses the random worlds model (which was introduced by Bacchus et al. [4]). Thus it is likely that all of these definitions frequently underestimate the risks of disclosure of sensitive information. A technique called Injector, by Li and Li [34] mines the original data for negative association rules that are then used in the anonymization process. It also uses a variation of random worlds for reasoning about privacy and is also likely to underestimate the risk of disclosure. Du et al. [16] provide a privacy definition that assumes an attacker reasons using the maximum entropy principle. Maximum entropy also frequently cannot learn the correlations between attributes. It is easy to see that in Table 1, the maximum entropy principle will give the probability of tuple 200 having lung cancer is 0.5 despite the strong correlation exhibited by the data. Thus it is likely that this approach can also be attacked by an attacker using deFinetti's theorem.

Two very well-known privacy schemes that provide guarantees against attackers who use the tuple-independent and i.i.d. models of reasoning were developed by Miklau et al. in the context of perfect privacy [37] and by Evfimievski et al. in the context of γ-amplification and $\rho_1 - to - \rho_2$ privacy [19]. We believe that these schemes (and many others that are based on similar models) are therefore vulnerable to the deFinetti attack and a specific attack algorithm is an interesting area of future research. Rastogi et al. [42] investigate several reasoning models in addition to the tuple-independent model. They also consider an adversary who knows/believes in arbitrary correlations between tuples. They show this results in privacy leakage whenever a sanitization algorithm provides "meaningful" results. Correlations induced by exchangeability or some of its extensions known as *partial exchangeability* [14] are a subset of arbitrary correlations and we believe this subset represents more meaningful and more realistic beliefs about the relationships between tuples. Furthermore this allows an attacker to learn the correlations instead of having to produce them up front.

Aggarwal et al. [1] provide an algorithm for limiting the ability of an attacker to form and reason with association rules (that have high confidence) from the sanitized data (this is also a hot topic). Association rules do consider correlations between attributes. Nevertheless, we believe even these privacy schemes are vulnerable to the deFinetti attack. The reason is that association rules are a crude form of a probability estimate. Thus several weak association rules can be combined (for example, using the Naive Bayes formula in Equation 1) to yield a high probability. For example, if we have some evidence in the form of association rules that "smoking $\rightarrow$ cancer" and "chewing tobacco $\rightarrow$ cancer" then we can deduce that smoking and chewing tobacco probably increases the risk of cancer more than either activity in isolation. We can do this even without a reliable estimate of the confidence of the rule "smoking $\wedge$ chewing

[6]By squaring the differences in probabilities, sum-squared error essentially wipes out useful information

tobacco→ cancer. The ability to combine rules in this way was not considered.

The machine learning community is starting to look learning problems that are relevant to the privacy literature. For example, Chen et al. [9] and Xiong et al. [50] investigate how to learn from aggregate views such as those produced by the privacy scheme proposed by Kifer et al. [27] (and thus may possibly be used to attack this scheme). Quadrianto et al. [41] consider the problem of predicting the class labels (e.g. the values of a sensitive attribute) of a set of tuples where the training set is a group of tuples. In each group, the approximate number of tuples from each class are known (and similarly for the testing data set). This corresponds exactly to the problem of learning from an anatomized table. However, their approach requires the group sizes of the partition to be very large (so that the estimators they use can approximately converge to their true values) and thus is not applicable here. de Freitas et al. [12] also consider this learning problem. However, their approach is tuned for predicting the class attribute (e.g. sensitive attribute) of tuples not in the data (to maximize generalization performance). Furthermore, their algorithm works well when there is uncertainty about the number of sensitive values in each group. When this quantity is known with absolute certainty (as in our case), their algorithm has difficulty in learning its parameters.

Although there are quite a few attacks studied in the literature, their number is dwarfed by the amount of new privacy definitions and proposed privacy algorithms. Some of the most notable real-world attacks include the linking attack [46] on data sets that do not contain unique identifiers, the published attack on AOL data [6], and the proof-of-concept attack on Netflix data [40, 39]. In addition to these attacks, there have been some other attacks on proposed sanitization schemes. Machanavajjhala et al. [35] list some attacks on k-anonymity. Kargupta et al. [26] and Huang et al. [24] use spectral techniques to remove additive noise. Ganta et al. [21] show how to attack multiple independently released data sets that contain partial overlaps of individuals. Wong et al. [48] and Fang et al. [20] consider how to attack sanitization algorithms that are framed as an optimization problem. We believe they can be combined with the deFinetti attack to predict even better probabilities. Finally, there has also been some work on attacks on data sets where unique id's have been removed but where an attacker has background knowledge or access to external data sets. These include the linking attack by Sweeney [46] for relational tables, the attacks by Backstrom et al. [5] and Hay et al. [23] on social networks, the attack by Lakshmanan et al. [30] on itemsets, and the more sophisticated attack by Kumar et al. [29] on search logs (which can also be viewed as itemsets).

8. CONCLUSION

Our goal was to highlight potential weaknesses in many data sanitization techniques and to demonstrate a specific exploitation of such a weakness. The reason we were successful is because tools for reasoning about privacy are not sufficiently mature yet. We believe that a fruitful goal for the privacy community is to develop such tools, as they will determine whether it is feasible to use partition-based sanitization schemes, or whether approaches like differential privacy [17] should be used instead.

We demonstrated that reasoning methods based on the random worlds, tuple-independent, and i.i.d. models result in attackers that, in many cases, will not change their beliefs despite overwhelming evidence. Such attackers can be considered very obstinate. Nevertheless, such attackers can also be considered as a reasonable first approximation. The reason is that an attacker does not need to be correct to cause harm; the attacker only needs to be able to convince a sizable number of people that the reasoning is correct. For individuals without statistical training, these three models of reasoning do appear reasonable. However, those with statistical training are more likely to be convinced by more sophisticated reasoning models, such as the ones presented here. The idea of distinguishing between relatively naive and sophisticated attackers can be traced back to Muralidhar and Sarathy [38] who called them "casual snoopers" and "professional snoopers", respectively.

Extending these ideas, we believe that a useful area of future research is in the development of reasoning strategies for attackers with various levels of sophistication. The security of private information for various sanitization schemes (including new proposals) should then be evaluated against each of these models.

9. REFERENCES

[1] Charu C. Aggarwal, Jian Pei, and Bo Zhang. On privacy preservation against adversarial data mining. In *KDD*, 2006.

[2] Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Rina Panigrahy, Dilys Thomas, and Ahn Zhu. Achieving anonymity via clustering in a metric space. In *PODS*, 2006.

[3] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[4] F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller. From statistics to beliefs. In *AAAI*, 1992.

[5] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.

[6] Michael Barbaro and Tom Zeller. A face is exposed for AOL searcher no. 4417749. *New York Times*, August 9 2006.

[7] Roberto Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymity. In *ICDE*, 2005.

[8] Yvonne M. Bishop, Stephen E. Fienberg, and Paul W. Holland. *Discrete Multivariate Analysis: Theory and Practice*. Springer, 2007.

[9] Bee-Chung Chen, Lei Chen, David Musicant, and Raghu Ramakrishnan. Learning from agggregate views. In *ICDE*, 2006.

[10] Bee-Chung Chen, Kristen LeFevre, and Raghu Ramakrishnan. PrivacySkyline: Privacy with multidimensional adversarial knowledge. In *VLDB*, 2007.

[11] Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, 2004.

[12] Nando de Freitas and Hendrik Kück. Learning about individuals from group statistics. In *UAI*, pages 332–339, 2005.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em

algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[14] Persi Diaconis and David Freedman. De finetti's generalizations of exchangeability. In Richard C. Jeffrey, editor, *Studies in Inductive Logic and Probability, Volume II*, pages 233–249. University of California Press, 1980.

[15] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[16] Wenliang Du, Zhouxuan Teng, and Zutao Zhu. Privacy-maxent: integrating background knowledge in privacy quantification. In *SIGMOD*, 2008.

[17] Cynthia Dwork. Differential privacy. In *ICALP*, 2006.

[18] Fernando Esponda, Elena S. Ackley, Paul Helman, Haixia Jia, and Stephanie Forrest. Protecting data privacy through hard-to-reverse negative databases. *International Journal of Information Security*, 6(6):403–415, 2007.

[19] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy-preserving data mining. In *PODS*, 2003.

[20] Chengfang Fang and Ee-Chien Chang. Information leakage in optimal anonymized and diversified data. In *Information Hiding*, 2008.

[21] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *KDD*, 2008.

[22] Gabrial Ghinita, Panagoiotis Karras, Panos Kalnis, and Nikos Mamoulis. Fast data anonymization with low information loss. In *VLDB*, 2007.

[23] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. In *VLDB*, 2008.

[24] Zhengli Huang, Wenliang Du, and Biao Chen. Deriving private information from randomized data. In *SIGMOD*, June 2004.

[25] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *UAI*, 1995.

[26] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *ICDM*, 2003.

[27] Daniel Kifer and Johannes Gehrke. Injecting utility into anonymized datasets. In *SIGMOD*, 2006.

[28] Donald E. Knuth. *The art of computer programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional, Boston, MA, USA, 3rd edition, 1997.

[29] Ravi Kumar, Jasmine Novak, Bo Pang, and Andrew Tomkins. On anonymizing query logs via token-based hashing. In *WWW*, 2007.

[30] Laks V. S. Lakshmanan, Raymond T. Ng, and Ganesh Ramesh. To do or not to do: the dilemma of disclosing anonymized data. In *SIGMOD*, 2005.

[31] Kristen LeFevre, David DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, 2005.

[32] Kristen LeFevre, David DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, 2006.

[33] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, 2007.

[34] Tiancheng Li and Ninghui Li. Injector: Mining background knowledge for data anonymization. In *ICDE*, 2008.

[35] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007.

[36] David Martin, Daniel Kifer, Ashwin Machanavajjhala, Johannes Gehrke, and Joseph Halpern. Worst case background knowledge for privacy preserving data publishing. In *ICDE*, 2007.

[37] Gerome Miklau and Dan Suciu. A formal analysis of information disclosure in data exchange. In *SIGMOD*, 2004.

[38] Krishnamurty Muralidhar and Rathindra Sarathy. Security of random data perturbation methods. *ACM Transactions on Database Systems*, 24(4):487–493, 1999.

[39] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. http://arxiv.org/abs/cs/0610105, 2006.

[40] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy (SP)*, 2008.

[41] Novi Quadrianto, Alex J. Smola, Tibério S. Caetano, and Quoc V. Le. Estimating labels from label proportions. In *ICML*, pages 776–783, 2008.

[42] Vibhor Rastogi, Dan Suciu, and Sungho Hong. The boundary between privacy and utility in data publishing. In *VLDB*, 2007.

[43] Jason Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *ICML*, 2003.

[44] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 2005.

[45] Mark J. Schervish. *Theory of Statistics*. Springer, 1995.

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

[47] L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 1979.

[48] Raymond Wong, Ada Fu, Ke Wang, and Jian Pei. Minimality attack in privacy preserving data publishing. In *VLDB*, 2007.

[49] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *VLDB*, 2006.

[50] Hui Xiong, Michael Steinbach, and Vipin Kumar. Privacy leakage in multi-relational databases: a semi-supervised learning perspective. *VLDB Journal*, 15(4):388–402, 2006.