Publishing Naive Bayesian Classifiers: Privacy without Accuracy Loss

Barzan Mozafari
University of California Los Angeles
Los Angeles, US
barzan@cs.ucla.edu

Carlo Zaniolo
University of California Los Angeles
Los Angeles, US
zaniolo@cs.ucla.edu

ABSTRACT

We address the problem of publishing a Naïve Bayesian Classifier (NBC) or, equivalently, publishing the necessary views for building an NBC, while protecting privacy of the individuals who provided the training data. Our approach completely preserves the accuracy of the original classifier, and thus significantly improves on current approaches, such as randomization or anonymization, which typically degrade accuracy to preserve privacy. Current query-view security checkers address the question of 'Is the view safe to publish?' and are computationally expensive (often Π_2^p-complete). Here instead, we tackle the question of 'How to make a view safe to publish?' and propose a linear-time algorithm to publish safe NBC-enabling views.

We first show that a simple measure that restricts the ratios between the published NBC statistics is sufficient to prevent any breach of privacy. Then, we propose a linear-time algorithm to enforce this measure by producing perturbed statistics that assure both (i) individuals' privacy, and (ii) a classifier that behaves in the same way as the NBC trained on the original data. By carefully expressing the derived statistics using rational numbers, we can easily produce synthetic (sanitized) datasets. Thus, for any given dataset, we produce another dataset that is secure to publish (w.r.t. a uniform prior) and achieves the same classification accuracy. Finally, we extend our results by providing sufficient conditions to cope with arbitrary (non-uniform prior) distributions, and we validate their effectiveness in practice through experiments on real-world data.

1. INTRODUCTION

Recent advances in digitized information has led to escalation of global concerns on individuals' privacy [3, 2, 1]. Privacy-Preserving Data Mining (PPDM) has been proposed to address these concerns. However, we are now facing conflicting goals: On one hand, to protect the privacy of the individuals whose sensitive information is present in our database, we should not disseminate such databases. On the other hand, many other legitimate users/applications can benefit from such data. For example, studying and mining medical records, consumers' behavior or insurance history by analysts can often lead to invaluable statistical knowledge which benefits the society at large. PPDM methods seek to achieve these benefits without compromising privacy.

Scenarios. Privacy-preserving methods can be applied during (i) the data collection phase, (ii) the data publishing phase, or (iii) the data mining phase:

(i) Individuals may not trust any parties except themselves and therefore they perturb their sensitive data before submitting it to the server that does the publishing or the mining.

(ii) In a database-publishing scenario, a trusted party holds the individual records, and it either performs some perturbation over the raw data before publishing it, or it only publishes parts (views) of it.

(iii) The trusted party that holds individuals' data computes the mining models locally; then, instead of publishing the original data or even an anonymized/perturbed version of it, the trusted party only publishes the mining results—while making sure that the publication of these results does not compromise privacy.

While our work uses several techniques adapted from scenario (ii), its objectives are aligned with (iii), as illustrated by the following example.

Privacy breaches when publishing NBCs. Consider a database schema $T = \langle Adr, Age, Sal \rangle$, where the address field can be either Westwood Blvd. (W) or Palms St. (P), and age is either 30 or 40. The sensitive attribute is annual salary, which is either $\$50K$ or $\$70K$. Assume that we want to publish (or train) an NBC over this database, such that given $\langle Adr, Age \rangle$ the model can predict the person's salary; this means views $\langle Adr, Sal \rangle$ and $\langle Age, Sal \rangle$ must be released[1]—or alternatively, the counts of all such pairs from which these views can be built. The views in question are shown in Figure 1(a). The intended users will invoke the NBC formula (see eq.(2) in Section 3) to build a Bayesian classifier. However, malicious user Bob, who is trying to breach the privacy of Alice (she was part of the training data), will instead generate all possible instances that are consistent with his additional information that Alice lives on Westwood and that she is in her 40s[2]. Thus, Bob will obtain instances d_1 to d_{10}, shown in Figure 1(a).Then, for each d_i, Bob counts the ratio of the tuples $\langle W, 40, 70K \rangle$ over those that have $\langle W, 40 \rangle$ in their first two columns (all possible tuples that match his info about Alice). Thus, Bob gets $4/5$ for d_1, $3/4$ for d_2, d_3, d_4, d_5, and 1 for all the others (i.e. d_6 to d_{10}). Finally, by averaging these 10 different ratios, Bob infers that with a probability of $\frac{1}{10}(4/5 + 4 \times 3/4 + 5 \times 1) = 88\%$ Alice earns a

[1]We call such views NBC-enabling views—Section 3.

[2]In general, Bob does not need to know all the attributes of Alice to breach her privacy.

VLDB '09, August 24-28, 2009, Lyon, France

Published Views		All consistent instances with T_1				
$\pi_{Adr,Sal}(T_1)$	$\pi_{Age,Salary}(T_1)$	d_1	d_2	d_3	d_4	d_5
$W, 70K$	$40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$
$W, 70K$	$40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$
$W, 70K$	$40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$	$W, 40, 70K$
$W, 70K$	$40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$
$P, 70K$	$30, 70K$	$P, 30, 70K$	$P, 40, 70K$	$P, 40, 70K$	$P, 40, 70K$	$P, 40, 70K$
$W, 50K$	$40, 50K$	$W, 40, 50K$	$W, 40, 50K$	$W, 40, 50K$	$W, 40, 50K$	$W, 40, 50K$
$P, 50K$	$30, 50K$	$P, 30, 50K$	$P, 30, 50K$	$P, 30, 50K$	$P, 30, 50K$	$P, 30, 50K$
		d_6	d_7	d_8	d_9	d_{10}
		$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$
		$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$
		$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$	$W, 40, 70K$
		$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 40, 70K$
		$P, 30, 70K$	$P, 40, 70K$	$P, 40, 70K$	$P, 40, 70K$	$P, 40, 70K$
		$W, 30, 50K$	$W, 30, 50K$	$W, 30, 50K$	$W, 30, 50K$	$W, 30, 50K$
		$P, 40, 50K$	$P, 40, 50K$	$P, 40, 50K$	$P, 40, 50K$	$P, 40, 50K$

(a) View set V_1 and all its possible worlds

Published Views		All consistent instances with T_2					
$\pi_{Adr,Sal}(T_2)$	$\pi_{Age,Sal}(T_2)$	d'_1	d'_2	d'_3	d'_4	d'_5	d'_6
$W, 70K$	$40, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$
$W, 70K$	$40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$	$W, 40, 70K$	$W, 30, 70K$	$W, 40, 70K$
$P, 70K$	$30, 70K$	$P, 30, 70K$	$P, 40, 70K$	$P, 40, 70K$	$P, 30, 70K$	$P, 40, 70K$	$P, 40, 70K$
$W, 50K$	$40, 50K$	$W, 40, 50K$	$W, 40, 50K$	$W, 40, 50K$	$W, 30, 50K$	$W, 30, 50K$	$W, 30, 50K$
$P, 50K$	$30, 50K$	$P, 30, 50K$	$P, 30, 50K$	$P, 30, 50K$	$P, 40, 50K$	$P, 40, 50K$	$P, 40, 50K$

(b) View set V_2 and all its possible worlds

Figure 1: NBC-enabling views for two tiny databases and their corresponding worlds

$70K$ salary. Bob could have a prior knowledge, e.g. he knew the overall distribution of salaries, but not the dependence of salary on other attributes This assumption is solely for the sake of this example. In general, we do not restrict Bob's prior knowledge. Thus, if his prior belief on Alice earning $70K$ was $\frac{5}{7} = 71\%$, after seeing those views, there would be a significant breach of Alice's privacy (from 71% to 88%).

Now instead, suppose that the views in question were the ones shown in Figure 1(b), and Bob did the same exhaustive computation over all possible instances, shown as d'_1 to d'_6 in Figure 1(b). In this case, the ratio of the tuples $\langle W, 40, 70K\rangle$ over all the tuples having $\langle W, 40\rangle$ averaged over $d'_1, \cdots, d'_6$ is $\frac{1}{6}(2/3+1/2+1/2+1+1+1) = 78\%$. Comparing these two sets of NBC-enabling views, clearly the latter case was safer to publish as it only moved Bob's prior knowledge from 71% to 78% instead of 88% in case of the former set of views. As discussed later, privacy breach [14] is a measure that limits the amount of additional knowledge that the attacker can obtain from the published data.

The key observation to be made is that although these two sets of views V_1 and V_2 (Figure 1) are so different in terms of privacy, the two NBCs built from them, will still return the same results for any tuple to be classified. For example if the test input is $\langle P, 30\rangle$, the NBC built on V_1 predicts the class label as $50K$ because $5\frac{1}{5}\frac{1}{5} < 2\frac{1}{2}\frac{1}{2}$. The prediction from the second classifier (built on V_2) is again $50K$ because $3\frac{1}{3}\frac{1}{3} < 2\frac{1}{2}\frac{1}{2}$ (A review of NBC formula is given in Section 3, see eq. (2)). The reader can also check the consistency of these two classifiers for all other possible inputs[3].

Despite its simple formulation, NBC has proved to be one of the most effective classifiers in practice and in theory [12]. However, as suggested by the above example, given an unsafe NBC, it is possible to find an equivalent one that is safer to publish. In short, the objective of this paper is determining whether a set of NBC-enabling views are safe to publish (against the aforementioned computation by Bob), and if not, how to find a secure database that produces the same NBC model.

Problem statement. In this paper, we assume a single trusted party who has a dataset containing sensitive personal information on some individuals. The goal is to publish an NBC model (which consists of NBC-enabling views or counts, described in Section 3), such that the privacy of the individuals who provided our training data is protected. The privacy guarantees that we provide here are the well-known notions of no privacy breach [14] and t-closeness [24], which we reformulate for the case of view publishing.

Attack model. The computational power of the attacker consists of considering all possible worlds that are consistent with the set of published views, and then counting the number of tuples that he/she is interested in, to compute the probability of the desired predicate.

Previous work has focused on the privacy breach risk that is inherent in publishing a black-box predictor, i.e., providing the public with the functionality of making predictions, while completely concealing the mechanisms and statistics by which they are derived (see discussion in [17], and Section 2). Here, we assume the risk of publishing a black-box predictor was deemed acceptable, but then the black box proved impractical (e.g., computationally intractable[16]). Therefore, this paper tackles the question of whether, rather than the mythical black box, we can instead divulge the simplest of classifiers, i.e., an NBC, and still offer the same privacy guarantees.

Contributions. By reformulating the notion of privacy breach in the context of view publishing, we derive sufficient conditions that are independent from (i) the predicate that the attacker is after, and (ii) the amount of his prior knowledge about the individual's attributes. Said conditions also guarantee that the attacker can never gain knowledge on an individual's sensitive-attribute (class label) in excess of the specified privacy limit. Thus, for NBC-enabling views, we show that total privacy (i.e., elimination of privacy breaches) can always be enforced when the background knowledge is uniform, while retaining perfect utility in terms of the NBC accuracy. We extend our results by providing sufficient conditions to cope with arbitrary (non-uniform) distributions, and we validate their effectiveness in practice through experiments on real-world data. We propose a simple and efficient (i.e., linear-time) algorithm for transforming a given set of NBC-enabling views into another set of views that (i) guarantees the required privacy level, (ii) imposes no accuracy loss in terms of building an NBC (unlike general-purpose techniques, such as randomization and k-anonymity).

Overview of the paper. The rest of this is organized as follows. After reviewing related work in Section 2, we provide a brief

[3]A brute-force decision procedure for checking the equivalence of two classifiers is exponential, but later we proposed a linear-time algorithm that guarantees their equivalence.

background on NBC in Section 3. In Section 4 we reformulate the notion of row-level privacy breach [14] to suit view publishing, followed by our results on safety conditions in Section 5. Our first algorithm for uniform distributions is proposed in Section 6, which is extended for arbitrary distributions in Section 7. Finally in Section 8, we validate the effectiveness of our algorithms on real-world data. We conclude in Section 9.

2. RELATED WORK

We briefly discuss closely related lines of prior work to clarify the context of our result—for a more general survey see [34] and references within.

Perturbation Methods. Such methods come in two flavors.

1. General-purpose approaches include but are not limited to randomization [4, 15, 25, 13], k-anonymity [33], l-diversity [26]. Here, the goal is to guarantee the requested privacy level by generalization, obfuscating, randomizing, permutation, suppression or sanitization while minimizing the information loss. Several attacks have been proposed against such approaches(e.g. [26] for k-anonymity, [19, 18, 30] for randomization), and they face efficiency issues as well (e.g. [27] for k-anonymity and [5] for randomization). However, generic information-theoretic measures of error in the raw data are sufficient but not necessary conditions for high accuracy of particular mining models. Thus, while the former is not possible in some cases [32], the latter might be still feasible. As a usual trade-off, accuracy loss is a downside of aforementioned general-purpose methods—see Section 8.2.

2. Ad-hoc methods are designed for a particular mining algorithm. They suppress or sanitize those parts of the model that violate privacy before publishing it. For example [7, 36] are for frequent pattern mining.

Query-View safety checking. A pioneering work here is [28] that addressed the query-view security problem, considering the sensitive information as a set of secret views (or queries) whose safety must be checked once other views or query results are published. However, their measure of perfect security is very strict, requiring that prior and posterior knowledge of the attacker must remain exactly the same after publishing the views which disallows many practically acceptable cases. Similar problems for database publishing and integration systems have been studied in [11, 31]. In particular, the 'Guarantee 3' in [31] is more similar to our assumption, as it ensures that an attacker who lacks other external knowledge about the possible sources cannot learn anything more. Violation of k-anonymity in view publishing was studied in [39]. In such approaches the complexity is usually prohibitive, e.g. deciding this problem for conjunctive views is Π_2^p-complete [28]. Moreover, their result is a 'safe/unsafe' answer, and does not provide a method for making the view safe to publish without loosing information. In this paper we consider simpler views (NBC-enabling views) but provide an efficient algorithm to make them safe.

Privacy breach. We extend the existing notion of privacy breach introduced by Evfimievski et al. [14], which relates the attacker's prior/posterior beliefs before/after seeing the perturbed data. Evfimievski et al. assume that each individual publishes her own tuple after applying some perturbation methods. However, in our context, individuals have trusted a single data publisher, who is in charge of perturbing the entire database before publishing it. Also, our algorithms are deterministic, while they exploit probabilistic methods (e.g, randomization). However, there is still a close connection between the two. In particular, our Lemma 2 corresponds to Statement 1 in [14], where their γ corresponds to our ρ. Furthermore, previous work on prior/posterior information proved that no anonymization can achieve both privacy and utility when the attacker's prior knowledge is already too large [32].

Mining result privacy. Reference [17] addresses the question of 'when can a classifier be published (to be freely invoked) without violating privacy?'. However, it assumes that the classifier can be published as a black-box whose inside representation cannot be seen. Similarly, [16] proposes a multi-party approach requiring a separate rule for all possible tuples. Representing an NBC as a rule-based classifier involves an exponential number of rules while our method uses linear time and memory (in input size).

3. NOTATIONS

Let the original database T be an instance of a relation[4] defined as $R = \langle A_1, \cdots, A_n, C\rangle$ in which A_i's are (the domains of) the attributes and C is (the domain of) the class label. Each tuple is associated with an individual. For example, in Figure 1, class label is the salary while address and age are A_i's. In order to build an NBC, the only views that need to be published are $\pi_{A_i,C}(T)$ for all $1 \leq i \leq n$, and $\pi_C(T)$. We use π for relational projection, and Π to denote product. Also, since throughout this paper we allow duplicate tuples, one can reconstruct these projection views by knowing how many times each pair of values have occurred together. In other words, *equivalent* to publishing these views, one can instead publish the following counts. For $1 \leq i \leq n, \forall t \in A_i, c \in C$, define:

$$N^i_{t,c} = |\sigma_{A_i=t \wedge C=c}(T)|$$

also $\forall c \in C$ define:

$$P_c = |\sigma_{C=c}(T)|$$

For example, in Figure 1(a), $N^{Adr}_{W,70K} = 4$, $P_{50K} = 2$ and so on. In practice, NBCs are usually published using these counts (either normalized as ratios or in their absolute value) due to their better memory efficiency over the view representation. Throughout this paper we shall switch between these two equivalent representations as needed to simply the discussion.

Using these counts, we can express the NBC's probability estimation as follows. For all $\tau = (t_1, \cdots, t_n) \in A_1 \times \cdots \times A_n$ and for all $c \in C$, the NBC's prediction is:

$$Pr[Class(\tau) = c] = \frac{\frac{P_c}{|T|}\prod_i(\frac{N^i_{t_i,c}}{P_c})}{|\sigma_{A_1=t_1 \wedge \cdots \wedge A_n=t_n}(T)|/|T|} \qquad (1)$$

Since the NBC goal is to compare $Pr[Class(\tau) = c]$ and $Pr[Class(\tau) = c']$ when $c \neq c'$, we can further simplify eq. (1) by ignoring those terms which are independent of the class label, and only compare

$$X_{\tau,c} = P_c \cdot \prod_i \frac{N^i_{t_i,c}}{P_c} \text{ and } X_{\tau,c'} = P_{c'} \cdot \prod_i \frac{N^i_{t_i,c'}}{P_{c'}} \qquad (2)$$

For simplicity, in this paper we assume that P_c counts are always non-zero, and therefore eq. (2) is always well-defined. As P_c and $N^i_{t,c}$ counts are sufficient for building an NBC, we use the pair (P, N) as the signature for each NBC . Thus, the problem (or input) size is $O(\sum_{i=1}^n |C| \cdot |A_i|)$.

In real-world datasets, there can be multiple sensitive attributes. Moreover, different individuals can have different privacy concerns, e.g. some people may consider their age more sensitive than their salary. For simplicity, in this paper we assume that C is the only sensitive information in T for the following reasons. It can be easily shown that all (non-class) attributes will benefit from the same or greater level of privacy that our results provide for the class label

[4]Throughout this paper we use the terms 'database', 'table' and 'relation' interchangeably.

Notation	Explanation
A_i	(domain of) i-th attribute
C	(domain of) the class label
$N^i_{t,c}$	# of tuples with label c, and value t for the i-th attribute
P_c	# of tuples with label c
(P, N)	NBC-enabling viewset composed of P and N counts
$X_{\tau,c}$	NBC score for tuple $\langle \tau, c \rangle$
I	a given quasi-identifier
I_0	(Alice's) value for I
D	all instances that have at least one tuple with $I = I_0$

Table 1: Notation summary.

C. Intuitively, this is due to the fact that in NBC-enabling views, we always release more information about C than about any other A_i's, as C appears in n views while each A_i appears in only one view. Informally, this means that knowing the values for some of the A_i's associated with Alice, after seeing the NBC, Bob can learn more about her class label rather than her unknown A_i's. Furthermore, multiple (sensitive or non-sensitive) class labels can always be combined together to form a single class label.

4. PRIVACY BREACH FOR VIEWS

In this section, we adapt the notion of privacy breach [14] to our context, where views are published by a single publisher (See Section 2). We define a quasi-identifier I as a non-empty subset of A_i attributes, whose values for Alice are known to Bob. We refer to the tuple made of these values as I_0, or simply say $I = I_0$. For instance, if $I = \langle A_1, A_3 \rangle$, any $\langle t_1, t_3 \rangle \in A_1 \times A_3$ can be a possible I_0. Also let D denote the family of all table instances whose projection on I contains I_0 as a tuple, that is $D = \{d \mid \exists t \in d, t.I = I_0\}$ where t is a tuple and d is a table instance. Table 3, summarizes our notation.

Privacy breach relates the adversary's prior and posterior knowledge about some property $Q : C \to \{True, False\}$ of the class label C in a tuple t, namely $Q(t.C)$. For example, one possible $Q(c)$ can be $c = HIV \vee c = Cancer$, where the domain is the disease types in a hospital. Here, we are overloading C (the domain of class labels) to also denote the class label of a tuple t. Thus, $Q(t.C)$ is defined as $Q(c)$ when $t.C = c$ for some $c \in C$. Let $\mathbf{P_1^{Q,I_0}}$ and $\mathbf{P_2^{Q,I_0}}$ be respectively the adversary's prior and posterior knowledge on a given property Q, defined as:

$$\mathbf{P_1^{Q,I_0}} = \sum_{d \in D} \mathbf{P}[Q(t.C) | t \in d, t.I = I_0] \cdot \mathbf{P}[d] \quad (3)$$

$$\mathbf{P_2^{Q,I_0}} = \sum_{d \in D} \mathbf{P}[Q(t.C) | t \in d, t.I = I_0] \cdot \mathbf{P}[d | V(d) = V_0] \quad (4)$$

Here, $\mathbf{P}[Q(t.C) | t \in d, t.I = I_0]$ is the probability that, in the table instance d, property Q is true for the class label of a tuple t that is consistent with Bob's quasi-identifier about Alice ($t.I = I_0$). Note that Bob knows that one such tuple must be associated with Alice[5]. For example, if there are two tuples in d that satisfy $t.I = I_0$, but Q is only true for one of them, Bob knows that given d, with a probability of 50%, the property Q holds for the class label of Alice. Moreover, since $d \in D$, there exists at least one such tuple (i.e., Alice) satisfying $t.I = I_0$ and therefore, this conditional probability is always well-defined.

[5]In a *row-level* publishing scenario [14] the owner of each row is known once its content is revealed. However, in our case (a *table-level* publishing scenario) the attacker also has some *quasi-identifier* of the victim(s) that helps him restrict all the possible rows in the table to a few.

In eq. (3) and (4), $\mathbf{P}[d]$ is the probability that the original table was d, while $\mathbf{P}[d | V(d) = V_0]$ is the conditional probability of the same event, knowing that the answer of a view V on d was V_0.

DEFINITION 1 (PRIVACY BREACH FOR VIEWS). *Let Q be any property on the sensitive class label C. For a given table T and a (set of) view(s) V, whose answer over T is V_0, we say that publishing $V(T) = V_0$ causes a privacy breach with respect to a pair of given constants $0 < L_1 < L_2 < 1$, if either of the following holds:*

1. *Upward L_1-to-L_2:* $\mathbf{P_1^{Q,I_0}} < L_1 < L_2 < \mathbf{P_2^{Q,I_0}}$.
2. *Downward L_2-to-L_1:* $\mathbf{P_2^{Q,I_0}} < L_1 < L_2 < \mathbf{P_1^{Q,I_0}}$.

Returning to our example in Section 1, the first set of views (Figure 1(a)) caused an upward 0.51-to-0.8 privacy breach, as the prior and posterior were 50% and 88%, respectively. With respect to the same privacy level (i.e., $L_1 = 0.51$ and $L_2 = 0.8$), the second set of views (Figure 1(b)) would be safe to publish, as their prior/posterior were 50% and 78%, respectively. However, if we had a more strict privacy policy, say $L_1 = 0.5$ and $L_2 = 0.6$, none of those viewsets would be safe to be published. Roughly speaking, the notion of privacy breach reflects the degree to which a change in the adversary's prior knowledge is tolerated.

In Sections 4 through 6, we assume a uniform distribution of the database instances, whereby all $d \in D$ are equally likely in the absence of any views. Also, after seeing the view(s), all instances in S are equally likely, where $S = \{d \in D | V(d) = V_0\}$ contains all instances satisfying the given view(s). This assumption is similar to that in [35]. We will remove these uniformity assumptions in Section 7. Thus, we have the following result[6].

STATEMENT 1. *Let I_0 be the value of a given quasi-identifier I, and let V_0 be the value of a given view $V(T)$. If there exist some $m_1, m_2 > 0$ such that for all $c \in C$:*

$$\frac{m_1}{|C|} \leq \frac{1}{|S|} \sum_{d \in S} \mathfrak{P}_d^c \leq \frac{m_2}{|C|} \quad (5)$$

where $\mathfrak{P}_d^c = \mathbf{P}[t.C = c | t \in d, t.I = I_0]$, then for any property Q and any pair of $L_1, L_2 > 0$ publishing $V = V_0$ will not cause any upward or downward privacy breaches w.r.t. L_1 and L_2, provided that the following amplification criterion holds:

$$\frac{m_2}{m_1} \leq \frac{L_2}{L_1} \cdot \frac{1 - L_1}{1 - L_2} \quad (6)$$

Intuitively, Statement 1 implies that a view V should not be too specific toward a particular class label. Publishing a view, causes many table instances to be ruled out, and therefore the mean of the $\mathfrak{P}_d^c$ values for the *remaining* ones, must be *relatively close* to the mean of $\mathfrak{P}_d^c$ values for *all* instances. This closeness, is determined by constraints (5) and (6) which are functions of the given security requirements (i.e., L_1,L_2). Moreover, the same closeness must hold for all class labels $c \in C$.

Note that although Statement 1 provides a sufficient condition for a view publishing to be safe, finding such m_1, m_2 that satisfy the constraints (5) and (6) requires computing $\mathfrak{P}_d^c$ values for all $d \in S$, and $c \in C$. However, the following lemma introduces yet another condition that is sufficient to satisfy those constraints, but only requires computing the means of $\mathfrak{P}_d^c$ values for different $c \in C$. An efficient algorithm for enforcing this new condition will be proposed in Section 6.

[6]This, and other omitted proofs can be found in [29].

LEMMA 2. *For a given quasi-identifier $I = I_0$, a given view $V(T) = V_0$ is safe to publish against any L_1-to-L_2 privacy breaches, if there exists $\rho > 1$ such that the following conditions hold:*

$$\frac{\rho + \rho^2(|C|-1)}{\rho + |C| - 1} < \frac{L_2}{L_1} \cdot \frac{1-L_1}{1-L_2} \tag{7}$$

and for all $c, c' \in C$:

$$\frac{\sum_{d\in S} \mathfrak{P}_d^c}{\sum_{d\in S} \mathfrak{P}_d^{c'}} < \rho \tag{8}$$

PROOF. We prove by showing that the conditions above imply Statement 1. To do that, we need to find numbers m_1, m_2 for which conditions(5) and (6) hold. By means of (8) for all $c, c' \in C$:

$$\frac{1}{\rho} \cdot \sum_{d\in S} \mathfrak{P}_d^{c'} \leq \sum_{d\in S} \mathfrak{P}_d^c \leq \rho \cdot \sum_{d\in S} \mathfrak{P}_d^{c'}$$

Using this observation and the fact that:

$$\sum_{c\in C} (\frac{1}{|S|} \cdot \sum_{d\in S} \mathfrak{P}_d^c) = 1$$

it can be proved by contradiction that for all $c \in C$:

$$\frac{1}{1+\rho(|C|-1)} \leq \frac{1}{|S|} \cdot \sum_{d\in S} \mathfrak{P}_d^c \leq \frac{\rho}{\rho+|C|-1}$$

$$\frac{1}{|C|}\frac{|C|}{1+\rho(|C|-1)} \leq \frac{1}{|S|} \cdot \sum_{d\in S} \mathfrak{P}_d^c \leq \frac{1}{|C|}\frac{\rho\cdot|C|}{\rho+|C|-1}$$

Therefore, by choosing $m_1 = \frac{|C|}{1+\rho(|C|-1)}$ and $m_2 = \frac{\rho\cdot|C|}{\rho+|C|-1}$ condition (5) is satisfied. Also condition (6) holds, because according to (7):

$$\frac{m_2}{m_1} = \frac{\frac{\rho\cdot|C|}{\rho+|C|-1}}{\frac{|C|}{1+\rho(|C|-1)}} = \frac{1+\rho(|C|-1)}{1+(1/\rho)(|C|-1)} < \frac{L_2}{L_1} \cdot \frac{1-L_1}{1-L_2}$$

□

Condition (8) is similar to the notion of amplification in randomization methods for the row-level publishing scenario [14]. Thus, we use their terminology, referring to ρ as amplification. Notice that for every $\rho > 1$:

$$\frac{m_2}{m_1} = \frac{\frac{\rho\cdot|C|}{\rho+|C|-1}}{\frac{|C|}{1+\rho(|C|-1)}} = \frac{1+\rho(|C|-1)}{1+(1/\rho)(|C|-1)} > 1$$

Also,

$$\lim_{\rho\to 1^+} \frac{1+\rho(|C|-1)}{1+(1/\rho)(|C|-1)} = 1$$

These imply that for any given $g > 1$, we can find a $\rho > 1$ such that $\frac{m_2}{m_1} < g$. On the other hand, by definition $0 < L_1 < L_2 < 1$. So we have: $\frac{L_2}{L_1} \cdot \frac{1-L_1}{1-L_2} > 1$. Therefore for any given L_1, L_2, by choosing $g = \frac{L_2}{L_1} \cdot \frac{1-L_1}{1-L_2} > 1$ we can select the largest possible ρ for which $\frac{m_2}{m_1} < g$ and then only check whether condition (6) holds, since condition (7) is automatically satisfied.

Hence, Lemma 2 allows us to recast our privacy goal as that of checking/enforcing condition (8) for a given ρ, assuming that maximum allowed amplification is determined by formula (7), where L_1 and L_2 are the privacy parameters specified by the user. Although this check is a sufficient and not a necessary condition for avoiding privacy breaches related to a given ρ, it is still a weak-enough condition to allow us to publish any classifier without any accuracy loss (after some transformation, Sections 5 and 6).

5. SAFETY CONDITION FOR NBC VIEWS

While checking for condition (8) on an arbitrary set of views might not be an easy task, in Lemma 3 we provide a sufficient condition for NBC-enabling views. In Section 6, we prove that this condition can always be achieved by replacing the original views with synthesized/sanitized ones that both satisfy condition (8) and result in the same classification behavior. Below and in the rest of this paper, we refer to NBC-enabling views simply as viewsets and use their (P, N) representation.

LEMMA 3. *With respect to a given I_0 as the value of a quasi-identifier I, and a given amplification ratio ρ, the viewset (P, N) is safe to publish, if for all $c, c' \in C$, $1 \leq i \leq n$ and $t \in A_i$ the following conditions hold:*

$$0 < \frac{P_{c'}}{P_c} \leq \sqrt[|I|]{\rho} \quad \textit{and} \quad 0 < \frac{N^i_{t,c}}{N^i_{t,c'}} \leq \sqrt[|I|]{\rho} \tag{9}$$

Lemma 3 is a sufficient criterion that ensures the safety of a viewset publication, only when a ρ parameter and a quasi-identifier are both given. However, in practice the same privacy guarantee must be provided for all individuals and for all possible quasi-identifiers (i.e., all non-empty I's and I_0's). To resolve this issue we make the following observation.

Since the condition (9) is a function of $|I|$, and not of I or I_0, all quasi-identifiers that have the same cardinality (i.e., number of attributes) can be blocked at the same time, once we ensure this condition for one particular pair of I and I_0. Moreover, note that $1 \leq |I| \leq n$ and

$$\sqrt[n]{\rho} < \sqrt[n-1]{\rho} < \cdots < \sqrt[1]{\rho}$$

Thus, all privacy breaches for all quasi-identifiers of any cardinality can be blocked by simply blocking the one with largest cardinality, namely n. Therefore, we have the following corollary.

COROLLARY 4. *With respect to a given amplification ratio ρ, the viewset (P, N) is safe to publish, if for all $c, c' \in C$, $1 \leq i \leq n$ and $t \in A_i$ the following conditions hold:*

$$0 < \frac{P_{c'}}{P_c} \leq \sqrt[n]{\rho} \quad \textit{and} \quad 0 < \frac{N^i_{t,c}}{N^i_{t,c'}} \leq \sqrt[n]{\rho} \tag{10}$$

Next, we show how this leads us to an efficient algorithm for transforming viewsets.

6. FROM UNSAFE VIEWS TO SAFE ONES

The previous section provided the sufficient conditions for avoiding any privacy breach with respect to a given ρ. Now the next question is 'what if condition (10) for NBC-enabling views of a particular database does not hold?'. To address this question, we provide a linear-time algorithm that enables us to transform the original set of views into a safe set of views which satisfies the safety condition of Corollary 4, and has the 'same quality' for the purpose of building an NBC. We next clarify this notion of 'same quality' more formally.

6.1 Equivalent views in building NBCs

In this section, we define the notion of equivalent sets of views (or counts) in terms of building an NBC. As mentioned in Section 3, the class prediction for a tuple τ is determined by the $X_{\tau,c}$ values in the following way. If there is a class label c_0 such that for all $c \in C\backslash\{c_0\}$, $X_{\tau,c_0} > X_{\tau,c}$, obviously the classifier's prediction will be a_0. However, to break the ties, there is also a pre-assigned precedence order among class labels. Namely, if $X_{\tau,c} =$

$X_{\tau,c'}$ then the classifier prediction goes to the one that has a higher precedence. In this paper, for the sake of presentation and without loss of generality, we assume that the class labels are numbers from 1 to $|C|$, and the larger the class label the higher the precedence. For example, if $C = \{1, 2, 3\}$ and $X_{\tau,1} = X_{\tau,2} = X_{\tau,3}$, the classifier's prediction will be class label 3. In case of a recommendation system where we need an ordered prediction from the classifier, the order would be 3 first,2 next and 1 last.

DEFINITION 2 (NBC-EQUIVALENCE). *Let f and f' be two functions that map each element of $\prod_i A_i \times C$ to a non-negative real number. We call f and f'* NBC-equivalent, *if $\forall \tau \in \prod_i A_i, \forall c, c' \in C$, $c < c'$:*

$$f(\tau, c) \leq f(\tau, c') \Leftrightarrow f'(\tau, c) \leq f'(\tau, c') \quad (11)$$

$$f(\tau, c) > f(\tau, c') \Leftrightarrow f'(\tau, c) > f'(\tau, c') \quad (12)$$

It is easy to show that NBC-equivalence is in fact reflexive, symmetric, and transitive. The real value that an NBC assigns to each $(\tau, c) \in \prod_i A_i \times C$ is its estimation of $Pr[Class(\tau) = c]$ which is computed using equations (1) or (2). Informally, Definition 2 implies that we are interested not in the actual values but in preserving the total *order* among them, namely $f(\tau, c_1), f(\tau, c_2), f(\tau, c_3), \cdots$ for **all** possible τ.

Notice that in many contexts, the classifier prediction is determined only by the label that has the highest associated probability, which means that all those classifiers whose first prediction (i.e., $\text{ArgMax}_c\{X_{\tau,c}\}$) is the same, have the same effect. However, there are some applications such as recommendation systems where the entire ranking matters. Thus, our notion of equivalent classifiers (Definition 2) preserves the entire ranking as well.

6.2 Transformation algorithms for unsafe views

So far, we established the safety of publishing a viewset when the sufficient condition holds (see (9) in Lemma 3 and (10) in Corollary 4). Now the next problem is what if the original viewset does not satisfy this condition? In the following, we present an algorithm that solves this problem by transforming an arbitrary viewset into an NBC-equivalent one that is safe to publish. A high-level pseudo code of this algorithm consists of four successive steps (Figure 2), where each step is a linear-time computation. The main part of this algorithm takes place in *Step 2* which makes the viewset safe to publish, by lowering the ratio between the counts until they satisfy eq. (10). The key idea of this step, is the following observation. Raising all the counts to the same power does not change the classification; In other words a set of NBC-equivalent viewsets is closed under exponentiation. For example, one could raise all the P and N values in eq. (2) to a fixed power, say $\frac{1}{100}$, without changing the order between $X_{\tau,c}$ and $X_{\tau,c'}$ for all τ, c and c'. Therefore, by choosing a small-enough power, the ratio between the resulting numbers goes down while the original classifier does not change.

However, the initial viewset might contain zero counts which will result in undefined ratios (i.e. ∞). Thus, before applying *Step 2*, in *Step 1* we carefully replace all those zeros with small-enough positive numbers in such a way that none of the existing inequalities are affected. Moreover, after raising all the numbers to the same power the following condition will no longer hold:

$$P_c = \sum_{t \in A_i} N^i_{t,c}$$

This issue will be resolved in *Step 3*. Finally, in *Step 4* we normalize the counts before publishing them.

In Figure 2, each step takes a viewset (P, N) as input and returns a new viewset which will be denoted by $(\mathbb{P}, \mathbb{N})$; These viewsets are

Algorithm SafetyTransform(V, ρ)
Input:
V is the given view consisting of $N^i_{t,c}$'s and P_c's;
ρ amplification ratio (see Lemma 3)
Description:
Step1(V): Replace all those $N^i_{t,c}$'s that are 0 to non-zero
Step2(V,$\rho^{\frac{1}{2n+3}}$): Scale down all $N^i_{t,c}$'s to new rational numbers that satisfy the given amplification ratio
Step3(V): Adjust the numbers such that again $\sum_t \mathbb{N}^i_{t,c} = \mathbb{P}_c$
Step4(V): Normalize the numbers or turn them into integers
Return V

Figure 2: High-level steps for moving an unsafe view towards a safe one.

provably NBC-equivalent. The output from each step is given as the input to the next step. Thus, due to the transitivity of NBC-equivalence, at the end of these four steps (when the last viewset is safe to be published w.r.t. a given ρ), the resulting NBC is still equivalent to the original one. Next, we present each step in detail and prove their correctness separately (For a running example, refer to [29]).

6.2.1 Step 1

The pseudo code for *Step 1* is given in Figure 3. In each iteration of the main loop (Line 2), a zero is replaced with a positive number. Therefore, at the end, there will be no zeros left (Remember that P values were positive, Section 3). Also, by a careful implementation, Line 2.1 will only take constant time. Therefore, the total running time for the main loop (Line 2) and the initialization (Line 1) is linear, with respect to the problem input size. Thus, all that remains to be proved is that the output of *Step 1* is NBC-equivalent to its input viewset, formally stated below.

Algorithm Step1(P, N)
Input:
(P, N) is the given viewset;
Description:
1: For each $c \in C$,
  For each A_i,
    $M^i_c \leftarrow \text{Max}\{N^i_{t,c} \mid t \in A_i\}$
    $m^i_c \leftarrow \text{Min}\{N^i_{t,c} > 0, +\infty \mid t \in A_i\}$
  $M_c \leftarrow \prod_i M^i_c$
  $m_c \leftarrow \prod_i m^i_c$
2: For each $c \in C$ in descending order,
  For each A_i,
    For each $t \in A_i$,
      If $N^i_{t,c} = 0$,
        $S^i_{t,c} = \text{Min}\{\frac{m_{c'}}{M_c} \cdot M^i_c \mid c' \in C \backslash \{c\}\}$
        $\mathbb{N}^i_{t,c} \leftarrow s$, where $0 < s < S^i_{t,c}$
2.1: Update M^i_c,m^i_c, M_c and m_c accordingly
      Else $\mathbb{N}^i_{t,c} \leftarrow N^i_{t,c}$
Return $(P, \mathbb{N})$

Figure 3: Step 1 - Removing zeros.

STATEMENT 5 (STEP 1 IS NBC-PRESERVING). *After algorithm* Step 1, (P, N) *and* $(\mathbb{P}, \mathbb{N})$ *are NBC-equivalent.*

PROOF. [7] Since non-zero counts have not changed, we only

[7] As we used $\mathbb{THIS}$ font to denote the output from each step, let

Algorithm Step2((P,N), ρ)
Input:
(P,N) is the given viewset;
$\rho > 1$ is the requested amplification ratio (Corollary 4)
Description:
1: $w \leftarrow \frac{\text{Max}\{N^i_{t,c}, P_i \mid 1\le i\le n, t\in A_i, c\in C\}}{\text{Min}\{N^i_{t,c}, P_i \mid 1\le i\le n, t\in A_i, c\in C\}}$
2: Choose a k such that $k \ge \frac{n\cdot\log w}{\log\rho}$
3: For each $c \in C$,
  For each A_i,
    $\mathbb{P}_c \leftarrow \sqrt[k]{P_c}$
    For each $t \in A_i$,
      $\mathbb{N}^i_{t,c} \leftarrow \sqrt[k]{N^i_{t,c}}$
4: Express the $\mathbb{P}_c$ and $\mathbb{N}^i_{t,c}$ values using rational numbers, with enough precision.
Return $(\mathbb{N}, \mathbb{P})$.

Figure 4: Step 2 - Enforcing the amplification condition.

need to consider those $\tau = \langle t_1, \cdots, t_n, c\rangle \in (\prod_i A_i)\times C$ for which $\exists i, N^i_{t_i,c} = 0$. For all such τ, $X_{\tau,c} = 0$. Thus, we need to show that for any c' for which $X_{\tau,c'} > 0$, we will have: $\mathbb{X}_{\tau,c} < \mathbb{X}_{\tau,c'} = X_{\tau,c'}$. Also, for any other $c' > c$ where $X_{\tau,c'} = 0$, we must show: $\mathbb{X}_{\tau,c} \le \mathbb{X}_{\tau,c'}$. To show this, notice that at any point in time M_c and $m_{c'}$ represent the maximum and minimum possible values of non-zero factors in $X_{\tau,c}$'s and $X_{\tau,c'}$'s, respectively. Therefore, $S^i_{t,c}$ is the maximum value that can be assigned to $\mathbb{N}^i_{t,c}$ such that the NBC inequality still holds. For the equality case, if $X_(\tau, c') = 0$ then because of the descending order of c's in removing zeros (Line 2) we are guaranteed that $f(\tau, c') > 0$, for all $c' > c$ when processing c. And in the case of $c' < c$, since in (P,N) their corresponding counts were both zero, and c has precedence over c', any positive number for $\mathbb{N}^i_{t,c}$ in $(\mathbb{P},\mathbb{N})$ will not change the classifier. □

6.2.2 Step 2

The pseudo code for *Step 2* is given in Figure 4. Note that performing k^{th} root (Line 3) preserves the NBC-equivalence. Moreover, since this operation scales down the numbers, the amplification requirement will be satisfied if k is chosen carefully. k is chosen (Line 2) such that the largest ratio between each pair of the original counts will be less than ρ. Also, w in Line 1 is always defined, as no zero count is left after *Step 1*. Thus, one can show that:

LEMMA 6. *At the end of Line 3 in Step 2, (P,N) and $(\mathbb{P},\mathbb{N})$ are NBC-equivalent and for all $c, c' \in C$, $1 \le i \le n$ and $t \in A_i$, we have:*

$$0 < \frac{\mathbb{P}_{c'}}{\mathbb{P}_c} \le \sqrt[n]{\rho} \quad \textit{and} \quad 0 < \frac{\mathbb{N}^i_{t,c}}{\mathbb{N}^i_{t,c'}} \le \sqrt[n]{\rho} \tag{13}$$

However, the more important challenge here is how to approximate the new numbers with rational numbers such that NBC-equivalence is not violated (We need them to be rational if we want to turn them into another synthesized database—see Section 6.2.4). In the following, $\widetilde{x}$ denotes a rational number approximation of x. To see why an arbitrary *fixed* precision may cause trouble, consider the following example.

Preserving ties. Suppose that the number of attributes is $n = 2$ and that for some t, t', originally we had $N^1_{t,1}\times N^2_{t',1} = 4\times 4 = 16$ and $N^1_{t,2}\times N^2_{t',2} = 2\times 8 = 16$. Assuming that $P_1 = P_2 = 100$, the original NBC would predict the class label as $c = 2$, because of the tie and the precedence of class $c = 2$ over class $c = 1$. Now, in case of $k = 2$ (i.e., $\sqrt{\ }$), if we used[8] a precision of 10^{-2} we would have $\mathbb{N}^1_{t,1}\times\mathbb{N}^2_{t',1} = 2\times 2 = 4$ and $\mathbb{N}^1_{t,2}\times\mathbb{N}^2_{t',2} = 1.41\times 2.83 = 3.9903$. Also $\mathbb{P}_1 = \mathbb{P}_2 = 10$. Thus, the new NBC would predict the class label as $c = 1$ which is inconsistent with the original NBC. Our solution to this issue is to use different precisions for the counts associated with different classes, such that the magnitude of the error goes in favor of the higher-precedence classes. In other words, if $c > c'$, over-approximate $\mathbb{N}^i_{t,c}$ and $\mathbb{N}^i_{t,c'}$ such that $0 < \widetilde{\mathbb{N}}^i_{t,c'} - \mathbb{N}^i_{t,c'} < \widetilde{\mathbb{N}}^i_{t,c} - \mathbb{N}^i_{t,c}$. By doing the opposite to $\mathbb{P}_c$ values, we can ensure that whenever $X_{\tau,c} = X_{\tau,c'}$, then $\widetilde{\mathbb{X}}_{\tau,c} \ge \widetilde{\mathbb{X}}_{\tau,c'}$. However, this can cause another issue, described next.

Preserving inequalities. For $c > c'$, since the over-approximation of the $\mathbb{N}^i_{t,c}$ values was larger than that of $\mathbb{N}^i_{t,c'}$ (and the opposite direction for $\mathbb{P}$), for some τ it can happen that we originally had $X_{\tau,c'} > X_{\tau,c}$ but now $\widetilde{\mathbb{X}}_{\tau,c'} \le \widetilde{\mathbb{X}}_{\tau,c}$. This results in a different classification. To address both of these issues we use the following result, which can be derived from the theory of Taylor series.

STATEMENT 7. *A real $k > 0$, a natural $n > 1$, and finite sets $Y_1, \cdots, Y_r \subset I\!N^{1/k} = \{x | x^k \in I\!N\}$ are given. For any $\epsilon > 0$, there exists a series $0 < \lambda'_r < \lambda_r < \cdots < \lambda'_1 < \lambda_1$ for which we can find a rational $\widetilde{x}$ for each $x \in \bigcup_{i=1}^{r} Y_i$, such that for any $1 \le i \ne j \le r$, $(x_1, \cdots, x_n) \in Y^n_i$ and $(y_1, \cdots, y_n) \in Y^n_j$:*

$$\textit{If } i < j: \prod_{s=1}^{n} x_s = \prod_{s=1}^{n} y_s \Rightarrow \prod_{s=1}^{n} \widetilde{x}_s \le \prod_{s=1}^{n} \widetilde{y}_s \tag{14}$$

$$\epsilon < \prod_{s=1}^{n} x_s - \prod_{s=1}^{n} y_s \Rightarrow \prod_{s=1}^{n} \widetilde{x}_s > \prod_{s=1}^{n} \widetilde{y}_s \tag{15}$$

Also for any $z_i \in Y_i, 1 \le i \le r$:

$$z_i + \lambda'_i < \widetilde{z}_i < z_i + \lambda_i \tag{16}$$

Notice that Statement 7 only preserves those original inequalities whose differences were at least ϵ. In order to preserve all inequalities, the following statement provides a lower bound on such an ϵ for our special case.

STATEMENT 8. *Let $M = Max\{N^i_{t,c} | c \in C, 1 \le i \le n, t \in A_i\}$. If there exist $\langle t_1, \cdots, t_n\rangle \in A_1\times\cdots A_n$ such that $\prod_{i=1}^{n} N^1_{t_i,c} \ne \prod_{i=1}^{n} N^1_{t_i,c'}$, then for any $k > 1$:*

$$|\sqrt[k]{\prod_{i=1}^{n} N^1_{t_i,c}} - \sqrt[k]{\prod_{i=1}^{n} N^1_{t_i,c'}}| \ge \frac{1}{k\cdot M^{\frac{n(k-1)}{k}}} \tag{17}$$

A symmetric approximation for $\mathbb{P}_c$ values can be derived in the opposite direction, but is omitted here for lack of space. Also, using a similar technique used in *Step 1*, we can ensure that the amplification condition between $\widetilde{\mathbb{N}}^i_{t,c'}$ and $\widetilde{\mathbb{P}}_c$ values still holds.

6.2.3 Step 3

The purpose of *Step 3* (Figure 5) is to assert that each $\mathbb{P}_c$ is actually equal to the sum of its corresponding $\mathbb{N}^i_{t,c}$ values, a condition that could have been violated in *Step 1* and *2*. In the following

$\mathbb{X}_{\tau,c}$ be similarly defined by formula (2) where X is replaced with $\mathbb{X}$, N with $\mathbb{N}$ and P with $\mathbb{P}$.

[8]The same problem can happen even for much higher precisions, as long as it is a fixed precision.

Algorithm Step3(P, N)
Input:
(P, N) is the given viewset;
Description:
1: For each $c \in C$,
For each A_i,
1.1: $S_i^c \leftarrow \sum_t N_{t,c}^i$
1.2: $R_0^c \leftarrow \frac{\prod_i S_i^c}{(P_c)^n}$
For each A_i,
1.3: $R_i^c \leftarrow \frac{R_0^c \cdot P_c}{S_i^c}$
2: For each P_c,
2.1: $\mathbb{P}_c \leftarrow R_0^c \cdot P_c$
For each $N_{t,c}^i$,
2.2: $\mathbb{N}_{t,c}^i \leftarrow R_i^c \cdot N_{t,c}^i$
Return $(\mathbb{P}, \mathbb{N})$

Figure 5: Step 3 - Adjust the numbers such that again $\sum_t \mathbb{N}_{t,c}^i = \mathbb{P}_c$

statement, we also show the degree to which the amplification ratio can change as a result of this step, and that the NBC-equivalence is still preserved.

STATEMENT 9. *Given a viewset (P, N), the new view generated by algorithm* Step 3*, say $(\mathbb{P}, \mathbb{N})$, has the following three properties:*

a. *Realistic view:* $\forall c \in C, 1 \leq i \leq n, \mathbb{P}_c = \sum_t \mathbb{N}_{t,c}^i$.
b. *Classification preserving:* $\forall c \in C, \tau \in \prod_i A_i$, $X_{\tau,c} = \mathbb{X}_{\tau,c}$.
c. *Amplification ratio: If* $\exists \rho > 1$ *s.t. (i)* $\forall x, y \in \{P_c | c \in C\}, 0 < \frac{x}{y} < \rho$ *and (ii)* $\forall x, y \in \{N_{t,c}^i | c \in C, 1 \leq i \leq n, t \in A_i\}, 0 < \frac{x}{y} < \rho$*, then we have:*
(iii) $\forall x, y \in \{\mathbb{P}_c | c \in C\}, 0 < \frac{x}{y} < \rho^{2n+3}$ *and (iv)* $\forall x, y \in \{\mathbb{N}_{t,c}^i | c \in C, 1 \leq i \leq n, t \in A_i\}, 0 < \frac{x}{y} < \rho^{2n+3}$

6.2.4 Step 4

After *Step 3*, $N_{t,c}^i$'s and P_c's are positive rational numbers that are (i) NBC-equivalent to the original counts and (ii) safe to publish. Now these rational numbers can be turned into integers again in *Step 4* in a straightforward manner. Having these positive integers ($\mathbb{N}_{t,c}^i$'s and $\mathbb{P}_c$'s), they can easily be used to make a new synthesized database. Based on the users' preference we can either publish the views (the tuples in each view will be permuted independently), or solely publish their corresponding integer counts, namely (P, N). Another choice is to always normalize these counts before publishing them, as such counts are enough for building an NBC even without revealing the actual size of the original database.

6.3 Uncertainty and Indistinguishability

Two important aspects of any privacy technique are uncertainty and indistinuishability [38, 37]. Indistinuishability is defined as the inability of telling the difference among individuals in a group. Uncertainty requires that the attacker cannot tell the sensitive value of an individual among a group of values. Non-probabilistic uncertainty is often based on whether the sensitive value can be uniquely inferred from the released data [22, 8, 20, 7] while probabilistic uncertainty concerns whether the cardinality of the set of possible sensitive values inferred for an individual is large enough and is often based on data distribution [39, 26, 14, 4, 28]. Our technique provides a high degree of both uncertainty and indistinguishability.

Uncertainty. The output of our algorithm is practically indistinguishable from the original data. The generated viewset looks like a real database, and in fact it is the original database if it was safe in the first place, i.e. *SafetyTransform* becomes an identity transformation. Thus, the adversary cannot tell whether he is dealing with the organl (safe) database or with a transformed one. Moreover, the adversary cannot uniquely find the original viewset by reversing our algorithm for the following reasons. Similar to [9], *SafetyTransform* introduces several layers of uncertainty throughout the transformation:
1. In Step1, Line 2, s values can be arbitrary/randomely chosen from the specifies interval.
2. In Step2, Line 2, any k value that satisfies the inequation can be arbitrary chosen.
3. In Step4, the final cardinality of the published database can be arbitrary chosen.

Although the *SafetyTransform* algorithm is known to the adversary, the data publisher does not need to announce the specific values chosen for the choices mentioned above. Next, we formally state why *SafetyTransform* also provides indistinguishability.

Indistinguishability. More strict notions (such as polynomial indistinguishability) are often used in cryptography, but in the database literature more practical metrics are usually applied, such as symmetric indistinguishability [38, 37], defined next.

DEFINITION 3 (SIND). *Consider a table T defined over a schema $T = \langle PA, SA \rangle$, where PA and SA are the public and sensitive attributes. A transformation $M()$ is said to provide symmetrically indistinguishable (SIND) if for any table instance d, where $M(d) = M(T)$, and for any two tuples $\langle p_1, s_1 \rangle, \langle p_2, s_2 \rangle \in d$ there exists another instance d' such that:*

1. $M(d') = M(T)$,
2. $\langle p_1, s_1 \rangle, \langle p_2, s_2 \rangle \notin d'$, *and*
3. $\langle p_1, s_2 \rangle, \langle p_2, s_1 \rangle \in d'$.

Note that we do not publish T but publish both $M()$ and its result on T, namely $M(T)$. Intuitively, SIND requires that one can swap the sensitive attributes between any two tuples, and the resulting table will still be a possible instance, i.e. it will be consistent with the published information that is $M(T)$. In our case, $M()$ consists of the NBC-enabling views followed by $SafetyTransform$ algorithm.

One can easily show that SIND is an equivalence binary relation, and thus, it will induce a partition on the set of tuples identifying SIND equivalence classes. SIND requires all the tuples to be in the same class, while a more practical notion can be similarly defined.

DEFINITION 4 (K-SIND). *We say a transformation $M()$ provides k-SIND, if each SIND equivalence class has a cardinality of at least k.*

Notice that k-anonymity is a special case of k-SIND property. Next result shows that *SafetyTransform* also provides such indistinguishability guarantees.

LEMMA 10. *The* SafetyTransform *algorithm provides k-SIND, where*

$$k = Min_{c \in C} P_c$$

PROOF. Note that any two tuples that have the same class label, can swap their sensitive attribute (i.e, their class label) without changing any of the NBC-enabling views. Thus, since the input viewsets are the same, *SafetyTransform* will also create the same output. Therefore, all tuples with the same class label form a SIND equivalence class. The smallest cardinality of such classes is the smallest P_c value. □

7. ARBITRARY PRIOR DISTRIBUTIONS

In Statement 1 and Lemmas 3 and 2, we assumed that the prior knowledge of the adversary is a uniform distribution over all class labels. In this section we extend our results to arbitrary (strictly-positive) distributions.

For simplicity, we assume that the prior knowledge of the adversary is in the form of a pmf (probability mass function) $\mathcal{F}$ that assigns non-zero probabilities to each class label. In general, the adversary's knowledge can be more specific, e.g. the probability of each class label given some quasi-identifiers, but here we do not discuss such cases.

According to [28], for any given set of views that contain an aggregate function, there exists a prior knowledge distribution that will change after publishing the views. Note that NBCs are also aggregate functions. Therefore, we make the assumption that the prior knowledge of the adversary (i.e., $\mathcal{F}$) is known to us, as the data publisher. This is a common assumption in the field [24, 9], which according to the above mentioned results (proven in [28]) cannot be easily avoided in the view publishing context. Thus, in practice, in order to protect privacy under the worst-case scenario, our publisher must assume that the adversary has access to the best publicly available knowledge about the application domain. For instance, in the case of medical data, a publisher must assume that the adversary knows the most recent statistics of different diseases and thus can accurately estimate $\mathcal{F}$. Hence, $\mathcal{F}(HIV) = 0.001$, $\mathcal{F}(Cancer) = 0.004$ and $\mathcal{F}(Cold) = 0.995$ might be a reasonable choice if the statistics show that on average 0.1% of patients (say, in US) have HIV and so on. Thus, the posterior knowledge that the adversary obtains after seeing the data published by a USA hospital should be as close as possible to 0.1%, for HIV cases at that hospital. This policy minimizes the additional information that our Bob will acquire about the hospital and patients such as Alice (who was treated there).

We next introduce a strong privacy measure that captures this notion of closeness between the prior and the posterior distributions, while the related algorithm is given in Section 7.2.

7.1 r-Closeness

We now introduce the notion r-closeness as follows:

DEFINITION 5 (r-CLOSENESS). *For $r > 1$, we say that publishing $V(T) = V_0$ satisfies r-closeness w.r.t. a given prior knowledge distribution $\mathcal{F}$, if for all $I = I_0$ and any property $Q(c)$ of the class label c, we have:*

$$\frac{1}{r} \leq \frac{\mathbf{P_2^{Q,I_0}}}{\mathbf{P_1^Q}} \leq r \tag{18}$$

where $\mathbf{P_2^{Q,I_0}}$ is the adversary's posterior knowledge defined in eq. (4) and, $\mathbf{P_1^Q}$ is his prior knowledge of property Q, now defined as:

$$\mathbf{P_1^Q} = \sum_{Q(c)} \mathcal{F}(c) \tag{19}$$

Note that the above definition is consistent with the intuition that the smaller r is, the more similar the posterior distribution is to the prior one. That is, when $r \approx 1$, the two distributions meet. The notion of r-closeness is semantically similar to that of t-closeness [24], which instead requires that the distance (either variational distance or KL distance) between the prior and posterior does not exceed t. In our r-closeness, the distance is defined by the maximum ratio of the two distributions on each possible class label. Thus, its syntactic definition is similar to the concept of 'Amplification' [14], which in turn corresponds to our ρ in Lemma 2. Analogous to

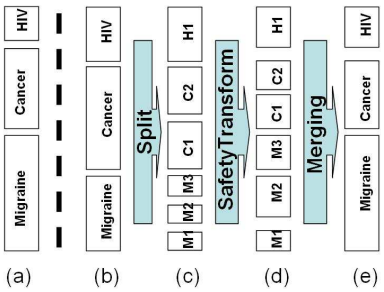

Figure 6: Visual demonstration of EST. (a) is the prior distribution of class labels, whose ratios are 1:2:3, proportionally. (b) is the original view of the data that deviates from the prior. Thus, (e) is the published view that must be more similar to (a) while still NBC-equivalent to (b).

Lemma 2 for privacy breach, the following result provides a sufficient condition to guarantee r-closeness. Notice that, r-closeness is a stronger form of privacy breach. In other words, once r-closeness is guaranteed, no privacy breach can occur w.r.t. any pair of L_1, L_2 where $\frac{L_2}{L_1} \geq r$.

STATEMENT 11. *Publishing $V(T) = V_0$ satisfies r-closeness w.r.t. a prior distribution $\mathcal{F}$, if for all $I = I_0$ and all $c, c' \in C$ we have:*

$$\frac{1}{r}\frac{\mathcal{F}(c)}{\mathcal{F}(c')} \leq \frac{P_2^{c,I_0}}{P_2^{c',I_0}} \leq r\frac{\mathcal{F}(c)}{\mathcal{F}(c')} \tag{20}$$

This sufficient condition enables us to use the algorithm *Safety-Transform* (Section 6.2) as a subroutine for enforcing r-closeness (if possible) w.r.t. an arbitrary strictly-positive prior distribution that is available to the adversary. This is discussed next.

7.2 Enforcing r-closeness

We first explain the general idea of the algorithm using the tiny example of Figure 6. For each one of the original class labels in $6(b)$, we create several new sub-labels, shown in $6(c)$. The number of sub-labels assigned to each original label is proportional to its prior probability, $\mathcal{F}$. Here, the prior ratio between HIV, Cancer, and Migraine was assumed to be $1:2:3$ resp., shown in $6(a)$. Then, we substitute the label of each tuple in $6(b)$ with one of its sub-labels, in $6(c)$. Each sub-label of a label gets the same share of the tuples that initially had that label, e.g. the tuples with Cancer in $6(b)$ are equally split between new labels $C1$ and $C2$. Now, provided that such a split is allowed (explained later), we can consider all these sub-labels (i.e., $H1$,$C1$,$C2$,$M1$,$M2$,$M3$ in $6(c)$) as new labels which now have a uniform prior distribution. Therefore, the required assumption for applying *SafetyTransform* holds. In the resulting view of this algorithm, shown in $6(d)$, the probabilities of different class labels are 'somewhat' close. Finally, by merging all class labels that were sub-labels of the same original label (e.g., the counts of $C1$ and $C2$ become somehow 'combined' as the new counts for $Cancer$ in $6(e)$), the new probabilities will be 'somewhat' similar to the prior. This is because the number of sub-labels for each label was chosen according to $\mathcal{F}$.

There are several technical difficulties that need to be resolved before such an algorithm works. In general, splitting and merging class labels are not necessarily NBC-preserving. Again, consider the tiny example in Figure 6. For a given τ, in $6(b)$ we may have $X_{\tau,Cancer} > X_{\tau,HIV}$, but $X_{\tau,C1} < X_{\tau,H1}$ in $6(c)$, as the counts for labels $C1$ and $C2$ are now half the counts for $Cancer$. Likewise for merging: $X_{\tau,M2} < X_{\tau,C1}$ in $6(d)$ may change to $X_{\tau,Migraine} > X_{\tau,Cancer}$ in $6(e)$. The algorithm which resolves

Algorithm EST((P, N), $\mathcal{F}$, r)
Input:
(P, N) is the given viewset;
$\mathcal{F}$ is the given pmf over the class labels;
r is the requested value for r-closeness;
Description:
1:**Resolving the ties:** such that $\forall c, c', \tau$: $X_{\tau,c} \neq X_{\tau,c'}$
2:**Split((P, N),$\mathcal{F}$):** Scale up $N^i_{t,c}$ and P_c values; then split each class label c according to $\mathcal{F}(c)$
3:**SafetyTransform((P, N),r):** Run the algorithm on new (sub) labels as if their prior distribution was uniform
4:**Merging((P, N)):** See if the new class labels can be merged back to the original labels, otherwise **Return FAIL.**
Return $(\mathbb{P}, \mathbb{N})$ **as the output from the last step**

Figure 7: Steps in EST

this problem, called *EST* (Extended Safety Transform), is provided in Figure 7. In the following, we explain each step of EST separately and address the aforementioned issues.

Resolving the ties. As we see later in Lemma 12, we need to first resolve all possible ties in the original NBC, i.e. for all τ and $c \neq c'$, $X_{c,\tau} \neq X_{c',\tau}$. This can be easily done using the following simple technique. Find a small enough $\epsilon > 0$ such that adding it to all the counts of any of the class labels does not change any of the original inequalities. Finding such a number can be done in linear time, by a technique similar to that used in Section 6.2.2. Now consider an arbitrary series $0 < \epsilon_1 < \cdots < \epsilon_{|C|-1} < \epsilon_{|C|} = \epsilon$, and add ϵ_i to all the counts of the i-th class label. Since the i-th class label has priority over the j-th label, $i < j$, all ties will be broken towards the higher precedence label while none of the original inequalities are affected. Thus, NBC-equivalence is still preserved.

Before presenting the rest of this algorithm, we formally define the following operations on NBCs.

DEFINITION 6 (SPLIT, MERGING). *Let V (with P_c's and $N^i_{t,c}$'s) and $\mathbb{V}$ (with $\mathbb{P}_c$'s and $\mathbb{N}^i_{t,c}$'s) be two NBCs defined over the same set of attributes but with two different classes, i.e. $\langle A_1, \cdots, A_n, C\rangle$ and $\langle A_1, \cdots, A_n, \mathbb{C}\rangle$ respectively. Also consider a mapping $\Psi : \mathbb{C} \rightarrow C$ for which $\Psi^{-1}(c) \neq \phi$ for all $c \in C$. We call $\mathbb{V}$ a split of V if for all $\mathbb{c} \in \mathbb{C}$ and all t, i:*

$$\mathbb{N}^i_{t,\mathbb{C}} = \frac{1}{|\Psi^{-1}(c)|} N^i_{t,c} \quad \textbf{\textit{and}} \quad \mathbb{P}_{\mathbb{C}} = \frac{1}{|\Psi^{-1}(c)|} P_c$$

where $c = \Psi(\mathbb{c})$. Likewise, we call V a merging of $\mathbb{V}$ if for all $c \in C$ and all t, i:

$$N^i_{t,c} = |\Psi^{-1}(c)| \cdot \underset{\Psi(\mathbb{C})=c}{Min} \{\mathbb{N}^i_{t,\mathbb{C}}\} \quad \textbf{\textit{and}} \quad P_c = |\Psi^{-1}(c)| \cdot \underset{\Psi(\mathbb{C})=c}{Min} \{\mathbb{P}_{\mathbb{C}}\}$$

A split (or merging) is called NBC-preserving when for all τ and all $c, c' \in C$, the following holds: $X_{c,\tau} \leq X_{c',\tau}$ if and only if there exist $\mathbb{c}, \mathbb{c}' \in \mathbb{C}$ such that $\Psi(\mathbb{c}) = c$, $\Psi(\mathbb{c}') = c'$ and $\mathbb{X}_{\mathbb{C},\tau} \leq \mathbb{X}_{\mathbb{C}',\tau}$.

The following result provides a sufficient condition for a split (or merging) to be NBC-preserving.

LEMMA 12. *Let Ψ be the maximum number of sub-labels mapped to a single label, i.e. $\hat{\Psi} = \underset{c \in C}{Max}\{|\Psi^{-1}(c)|\}$. Also assume that neither V nor $\mathbb{V}$ has a tie. A split defined over Ψ is NBC-preserving if $\hat{\Psi} \leq \frac{M_1}{M_2}$, where M_1 and M_2 are the first and the second largest $N^i_{t,c}$ in V. Similarly, a merging defined over Ψ is NBC-preserving if $\hat{\Psi} \leq \frac{\mathbb{M}_1}{\mathbb{M}_2}$, where $\mathbb{M}_1$ and $\mathbb{M}_2$ are the first and the second largest $\mathbb{N}^i_{t,\mathbb{C}}$ in $\mathbb{V}$.*

Thus, when there are no ties, i.e. $\frac{X_{\tau,c}}{X_{\tau,c'}} \neq 1$, we have:

$$\underset{\tau,c,c'}{Min}\{\frac{X_{\tau,c}}{X_{\tau,c'}} > 1\} \geq \frac{M_1}{M_2} \quad \textbf{and} \quad \underset{\tau,c,c'}{Max}\{\frac{X_{\tau,c}}{X_{\tau,c'}} < 1\} \geq \frac{M_2}{M_1}$$

Referring to $\frac{M_1}{M_2}$ as S_M, Lemma 12 implies that one can multiply all the counts of a particular class label by any constant s, as long as it is in the interval $\frac{1}{S_M} \leq s \leq S_M$. Another interesting observation is that by exponentiating all the counts in V to the same power θ, we can enlarge this interval arbitrarily from either side (recall that exponentiation is always NBC-preserving). That is, $S_M \rightarrow \infty$ when $\theta \rightarrow \infty$, or equivalently $\frac{1}{S_M} \rightarrow 0^+$ when $\theta \rightarrow 0^+$. This is the main idea behind the *Split* step, described next.

Split. Let us assume that $\mathcal{F}(c)$ values are either rational numbers or are given in a precise-enough rational representation (like the method used in Section 6.2.2). Thus, we can find their greatest common divisor, say $\mathbb{F}$. That is, for all $c \in C$, $\mathcal{F}(c) = \mathbb{F} \cdot F_c$ for some positive integer F_c. Now for each $c \in C$, we create new sub-labels $c_1, \cdots, c_{F_c}$ that are all mapped to label c. Let $\tilde{F} = \underset{c \in C}{Max}\{F_c\}$. In order for the this split to be NBC-preserving, we first raise the original counts to a big enough power θ before applying the split. More precisely, for any $\theta >> \frac{log\tilde{F}}{log(S_M)}$ the conditions of Lemma 12 will be satisfied, since after raising the counts to the power of θ, we will have $S'_M = {S_M}^\theta > \tilde{F}$ where S'_M denotes the new value.

SafetyTransform subroutine and Merging. As previously mentioned, after performing a split, the new class (sub)labels come from a uniform distribution. This allows us to apply algorithm SafetyTransform after which a merging operation is performed as follows. For each $c \in C$, all sub-labels $c_1, \cdots, c_{F_c}$ are mapped back to c (new NBC counts are determined according to Definition 6). Assuming that such a mapping is possible (later, Statement 14 determines when it is possible), we have the following analysis. After SafetyTransform (according to the results in Section 6.2), for all quasi-identifiers I_0, all $c, c' \in C$ and all $c_i \in \Psi^{-1}(c), c'_j \in \Psi^{-1}(c')$:

$$\frac{1}{r} \leq \frac{P_2^{c_i,I_0}}{P_2^{c'_j,I_0}} \leq r \quad \Rightarrow \quad \frac{1}{r}\frac{F_c}{F_{c'}} \leq \frac{F_c \cdot P_2^{c_i,I_0}}{F_{c'} \cdot P_2^{c'_j,I_0}} \leq r\frac{F_c}{F_{c'}}$$
$$\Rightarrow \quad 7\frac{1}{r}\frac{F_c}{F_{c'}} \leq \frac{P_2^{c_i,I_0}}{P_2^{c',I_0}} \leq r\frac{F_c}{F_{c'}}$$

Therefore, the required conditions for Statement 11 hold, proving that r-closeness is satisfied once the merging step is possible. The following lemma summarizes the properties of this algorithm.

LEMMA 13. *EST runs in linear time, and when returning a view V' for a given prior distribution $\mathcal{F}$, a privacy level r (for r-closeness) and the original view V, V' is safe to publish w.r.t. r, yet is NBC-equivalent to V.*

Lastly, we provide a closed form to determine the best r-closeness (i.e., smallest r) that our algorithm can enforce without losing any accuracy.

STATEMENT 14. *For a given V and a prior distribution of class labels $\mathcal{F}$, EST generates an NBC-equivalent V' that guarantees r-closeness w.r.t. $\mathcal{F}$, if there exists a large enough θ for which the following condition holds:*

$$\tilde{F} < \left(\frac{S_M^\theta}{\tilde{F}}\right)^{\frac{\log r}{(2n^2+3n)(\theta \cdot \log \frac{M}{m} + \log \tilde{F})}} \tag{21}$$

where n, m, M are the number of attributes in V, the minimum count in V (after removing zeros), and the maximum count in V, respectively.

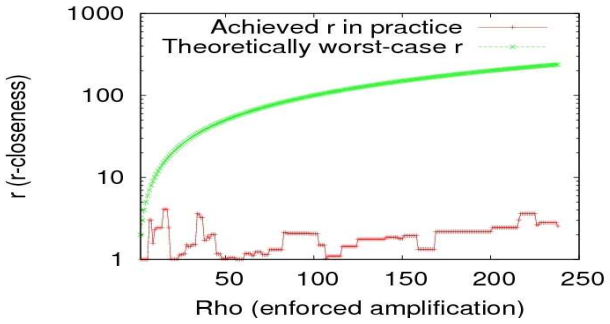

Figure 8: Achieved r-closeness on Adult dataset.

Taking the limit as $\theta \to \infty$, the condition simplifies to

$$\tilde{F} \leq S_M^{\frac{logr}{(2n^2+3n)\cdot \log \frac{M}{m}}}$$

It is worth mentioning that for uniform distributions $\tilde{F} = 1$, and this was why SafetyTransform could achieve any level of privacy. Moreover, note that Statement 14 is a sufficient but not a necessary condition. However, in practice, tighter analysis of S_M is possible which can lead to smaller values for r. Also, as shown in Section 8.1, the SafetyTransform algorithm provides much stronger amplification ratios than the requested r, due to its conservative approach. This means that in practice EST can call SafetyTransform with a much lower r than what is guaranteed by Statement 14, and still achieve the same or better r-closeness than what was initially requested. Thus, the applicability of EST depends on both the deviation of the actual data from $\mathcal{F}$ and the requested privacy level r (for r-closeness). If the distribution of the underlying data deviates too much from (F), apparently no one can guarantee a very small r without losing accuracy.

Moreover, another possibility is to trade-off accuracy loss against privacy (i.e., smaller r's) by performing the merging step (regardless of being NBC-preserving) with r' that ranges in $r \leq r' \leq r''$, where r is the required privacy level and r'' is the smallest value for r-closeness that satisfies Statement 14. At the extremes, choosing $r' = r$ completely ignores the accuracy loss, while $r' = r''$ preserves the accuracy, ignoring the requested privacy. We do not discuss such possible trade-offs here, as in this paper we focus only on the accuracy preserving case (whenever it is consistent with the requested privacy level, namely $r'' \leq r$).

8. EXPERIMENTS

The goal of our experiments is to evaluate (i) the effectiveness of our algorithms in practice (Section 8.1) and (ii) the accuracy loss imposed by other general-purpose techniques on NBCs (Section 8.2).

The experiments were conducted on a P4 machine running Linux, with 1GB RAM. Our algorithms (SafetyTransform & EST) were implemented in C++. We used the Adult dataset [6] which is a classic benchmark for privacy-preserving techniques. This dataset contains $32,561$ tuples from US Census data. The attributes that we used were Age, Years of education, Work hours per week, and Salary. The class label is based on salary which can be either $> 50K$ or $\leq 50K$. The running time of EST for processing this dataset was 2.920 seconds.

8.1 Amplification ratio and r-closeness

In the Adult dataset, the ratio of tuples with salary ≤ 50 to those with $> 50K$ was 24720 to 7841, i.e. $\tilde{F} \approx 3$. Moreover, in the original dataset, the minimum ρ (satisfying Corollary 4) was 238. By running SafetyTransform for $1 < \rho \leq 283$, we measured the actual r-closeness that is achieved by EST for this dataset, plotted in Figure 8. The actual r-closeness was measured by using combinatorial counting of all possible instances (that after the *Split* step, were assumed equally likely). As shown in Figure 8, the actual r-closeness is much better than the theoretical worst case (provided by Statement 14). This is due to the conservative upper bound derived from Lemma 3. This implies that, in practice, for a requested level of r-closeness, we can call SafetyTransform with a ρ that is much higher than r, and still preserve both NBC-equivalence and privacy level. This is because SafetyTransform achieves a much lower amplification guarantee than ρ^n. We have repeated this experiment with a different number of attributes, and also for synthetic datasets (both uniform and non-uniform distributions) and observed similar results.

8.2 NBC Accuracy

In this section, we have only focused on the effect of deterministic privacy methods on NBCs, but apparently the randomization techniques will also impose accuracy loss depending on their variance. Thus, we used k-anonymity as an example of a well-studied, general-purpose privacy technique, since it preserves the most accuracy compared to Entropy l-diversity, Recursive l-diversity, and t-closeness (see the experiments in [26, 24]). But even for k-anonymity, the accuracy loss was considerable for recall, as shown in Figure 9(b).

For anonymizing the Adult dataset we used Incognito implementation [23] which is a full-domain k-anonymity algorithm. We trained an NBC on the anonymized data (for different values of k) and compared the results with an NBC trained on the output of SafetyTransform, which is equivalent to training it on the original data. For ETS, we used $r = 1.3$ for a prior belief of 75% on $\leq 50K$ label. But note that the accuracy for both ETS and SafetyTransform are always equal to that obtained on the original data, regardless of the chosen value for r, and therefore we represent them all with the same (red) bar in Figure 9(a),(b). Each time, we used 50% of the tuples for training and the rest for testing. The overall accuracy of NBC does not drop much using k-anonymized data (about 5%, Figure 9(a)). However, the classification quality drops dramatically for less common classes. Since in the Adult dataset, tuples with salary of $> 50K$ are much fewer (one third) than those with $\leq 50K$, the recall for this smaller class is significantly affected, as shown in Figure 9(b). In many applications, classifying less common events is much more critical, e.g. in an online recommendation system or search engine advertisement, the probability of a click on a particular ad is very small. Also, since our algorithms retain the total order (NBC-equivalence), all metrics remain the same, such as accuracy, recall, precision and F-measure.

9. CONCLUSION AND FUTURE WORK

In this paper, we reformulated privacy breach for view publishing. We presented sufficient conditions that are easy to check/enforce, when the views in question are used to train Naïve Bayesian Classifiers (NBC). Indeed, we provided algorithms that (i) run in linear-time, (ii) guarantee the privacy of the individuals who provided the training data, (iii) incur zero accuracy loss in terms of building an NBC, (iv) work for any given database as long as the prior distribution is uniform, or it satisfies our sufficient condition. We validated the applicability and effectiveness of our algorithms by several experiments on real-world datasets.

Our proposed method has a clear advantage over general-purpose approaches, such as k-anonymity and randomization, that compromise the accuracy of information to achieve privacy. In a clear departure from these and other previous approaches that minimize the

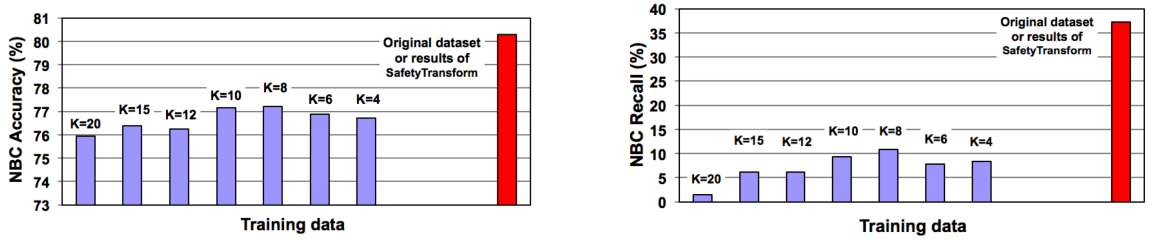

Figure 9: Drop in accuracy (a) and recall (b) of NBC when trained with k-anonymized data.

information loss in terms of the average error, we showed that for NBCs, a perfectly accurate mining model may still be achievable even if the average utility of a perturbation method is poor. This promising finding, also calls for more efforts in designing model-specific privacy-preserving approaches optimized for specific mining methods of wide usage. In the end, this could deliver more concrete benefits than seeking general-purpose techniques which have proven to be computationally complex and practically unrealistic [32].

NBCs are widely used in many successful classification and recommendation systems. Moreover, we are currently extending our techniques to general Bayesian networks. In fact, several problems including sensitivity analysis [10] on Bayesian networks can be reformulated using our notion of amplification. Some meta-algorithms such as bagging can be accommodated in a straightforward manner. Moreover, we are investigating the extension of our approach to augmented NBCs [21], decision trees built from NBCs, and incremental publication of NBCs over a data stream.

10. ACKNOWLEDGMENTS

The authors would like to thank Albert Lee and the reviewers for their insightful suggestions. This work was supported in part by NSF-IIS award: 0742267 'SGER: Efficient Support for Mining Queries in Data Stream Management Systems'.

11. REFERENCES

[1] Data mining: Staking a claim on your privacy. *Information and Privacy Commissioner, Ontario*, Jan. 1998.
[2] Directive on privacy protection. *European Union*, Oct. 1998.
[3] The end of privacy. *The Economist*, May 1999.
[4] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD*, 2000.
[5] S. Agrawal, V. Krishnan, and J. R. Haritsa. On addressing efficiency concerns in privacy-preserving mining. In *DASFAA*, 2004.
[6] A. Asuncion and D. Newman. UCI machine learning repository.
[7] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Blocking anonymity threats raised by frequent itemset mining. In *ICDM*, 2005.
[8] A. Brodsky, C. Farkas, and S. Jajodia. Secure databases: Constraints, inference channels, and monitoring disclosures. *TKDE*, 12(6), 2000.
[9] S. Bu, L. V. S. Lakshmanan, R. T. Ng, and G. Ramesh. Preservation of patterns and input-output privacy. In *ICDE*, 2007.
[10] H. Chan and A. Darwiche. Sensitivity analysis in Bayesian networks: From single to multiple parameters. In *UAI*, 2004.
[11] A. Deutsch and Y. Papakonstantinou. Privacy in database publishing. In *ICDT*, 2005.
[12] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3), 1997.
[13] W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *KDD*, 2003.
[14] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 2003.
[15] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *SIGMOD*, 2005.
[16] M. Kantarcioğlu and C. Clifton. Assuring privacy when big brother is watching. In *DMKD*, 2003.
[17] M. Kantarcioğlu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *KDD*, 2004.
[18] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *ICDM*, 2003.
[19] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowl. Inf. Syst.*, 2005.
[20] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. In *PODS*, 2005.
[21] E. Keogh and M. Pazzani. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *7th. Int'l Workshop on AI and Statistics*, 1999.
[22] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *PODS*, 2000.
[23] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, 2005.
[24] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, 2007.
[25] K. Liu, J. Ryan, and H. Kargupta. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *ITKDE*, 18(1), 2006.
[26] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.
[27] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, 2004.
[28] G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. In *SIGMOD*, 2004.
[29] B. Mozafari and C. Zaniolo. Privacy-preserving publication of naive bayesian classifier without accuracy loss. Technical report, UCLA, `http://wis.cs.ucla.edu/safeminer/index.htm`, 2008.
[30] K. Muralidhar and R. Sarathy. Security of random data perturbation methods. *ACM Trans. Database Syst.*, 24(4), 1999.
[31] A. Nash and A. Deutsch. Privacy in glav information integration. In *ICDT*, 2007.
[32] V. Rastogi, S. Hong, and D. Suciu. The boundary between privacy and utility in data publishing. In *VLDB*, 2007.
[33] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 10(5), 2002.
[34] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33(1), 2004.
[35] H. Wang and L. V. S. Lakshmanan. Probabilistic privacy analysis of published views. In *WPES*, 2006.
[36] T. Wang and L. Liu. Butterfly: Protecting output privacy in stream mining. In *ICDE*, 2008.
[37] C. Yao, L. Wang, X. S. Wang, C. Bettini, and S. Jajodia. Evaluating privacy threats in released database views by symmetric indistinguishability. In *J. of Computer Security*, To appear.
[38] C. Yao, L. Wang, X. S. Wang, and S. Jajodia. Indistinguishability: The other aspect of privacy. In *Secure Data Management*, 2006.
[39] C. Yao, X. S. Wang, and S. Jajodia. Checking for k-anonymity violation by views. In *VLDB*, 2005.