Toward Privacy in Public Databases ⋆

Shuchi Chawla[1], Cynthia Dwork[2], Frank McSherry[2], Adam Smith[3] ⋆⋆, and Hoeteck Wee[4]

[1] Carnegie Mellon University, shuchi@cs.cmu.edu
[2] Microsoft Research SVC, {dwork,mcsherry}@microsoft.com
[3] Weizmann Institute of Science, adam.smith@weizmann.ac.il
[4] University of California, Berkeley, hoeteck@cs.berkeley.edu

In Memoriam
Larry Joseph Stockmeyer
1948–2004

Abstract. We initiate a theoretical study of the *census problem*. Informally, in a census individual respondents give private information to a trusted party (the census bureau), who publishes a *sanitized* version of the data. There are two fundamentally conflicting requirements: *privacy* for the respondents and *utility* of the sanitized data. Unlike in the study of secure function evaluation, in which privacy is preserved to the extent possible given a specific functionality goal, in the census problem privacy is paramount; intuitively, things that cannot be learned "safely" should not be learned at all.

An important contribution of this work is a definition of privacy (and privacy compromise) for statistical databases, together with a method for describing and comparing the privacy offered by specific sanitization techniques. We obtain several privacy results using two different sanitization techniques, and then show how to combine them via cross training. We also obtain two utility results involving clustering.

1 Introduction

We initiate a theoretical study of the *census problem*. Informally, in a census individual respondents give private information to a trusted party (the *census bureau*), who publishes an altered or *sanitized* version of the data. There are two fundamentally conflicting requirements: *privacy* for the respondents and *utility* of the sanitized data. While both require formal definition, their essential tension is clear: perfect privacy can be achieved by publishing nothing at all – but this has no utility; perfect utility can be obtained by publishing the data exactly as received from the respondents, but this offers no privacy. Very roughly, the

⋆ A full version of this paper may be found on the World Wide Web at http://research.microsoft.com/research/sv/DatabasePrivacy/.

⋆⋆ This research was done while A.S. was a student at MIT, partially supported by a Microsoft fellowship and by US ARO Grant DAAD19-00-1-0177.

sanitization should permit the data analyst to identify strong stereotypes, while preserving the privacy of individuals.

This is not a new problem. Disclosure control has been studied by researchers in statistics, algorithms and, more recently, data mining. However, we feel that many of these efforts lack a sound framework for stating and proving guarantees on the privacy of entries in the database. The literature is too extensive to survey here; we highlight only a few representative approaches in Section 1.2.

1.1 Summary of Our Contributions and Organization of the Paper

Definitions of Privacy and Sanitization. We give rigorous definitions of privacy and sanitization (Sections 2 and 3, respectively). These definitions, and the framework they provide for comparing sanitization techniques, are a principal contribution of this work.

For concreteness, we consider an abstract version of the database privacy problem, in which each entry in the database—think of an individual, or a particular transaction—is an unlabeled point in high-dimensional real space $\mathbb{R}^d$. Two entries (points) that are close in $\mathbb{R}^d$ (say, in Euclidean distance) are considered more similar than two entries that are far.

Our first step was to search the legal and philosophical literature to find a good English language definition of privacy relevant to statistical databases. The phrase "protection from being brought to the attention of others" in the writings of Gavison [19] resonated with us. As Gavison points out, not only is such protection inherent in our understanding of privacy, but when this is compromised, that is, when we have been brought to the attention of others, it invites further violation of privacy, as now our every move is examined and analyzed. This compelling concept – protection from being brought to the attention of others – articulates the common intuition that our privacy is protected to the extent that we blend in with the crowd; moreover, we can convert it into a precise mathematical statement: intuitively, we will require that, from the adversary's point of view, every point be indistinguishable from at least $t-1$ other points, where t is a threshold chosen according to social considerations. Sweeney's seminal work on k-anonymity is similarly motivated [31]. In general, we think of t as much, much smaller than n, the number of points in the database.

In analogy to semantic security [21], we will say that a sanitization technique is secure if the adversary's probability of breaching privacy does not change significantly when it sees the sanitized database, even in the presence of auxiliary information (analogous to the history variable in the definition of semantic security). As noted below, auxiliary information is (provably) extremely difficult to cope with [14].

Histograms Preserve Privacy. A *histogram* for a database is a partition of $\mathbb{R}^d$ along with the exact counts of the number of database points present in each region. Histograms are prevalent in official statistics, and so they are a natural technique to consider for sanitization. We analyze a *recursive histogram sanitization*, in which the space is partitioned recursively into smaller regions (called cells) until no region contains $2t$ or more real database points. Exact counts of the numbers of points in each region are released. The intuition is that we reveal more detailed information in regions where points are more densely clustered. We prove a strong result on the privacy provided when the data are drawn uniformly from the d-dimensional unit cube, in the absence of auxiliary information. Generalizations are discussed in Remark 4 and Section 7.

Density-based Perturbation Provides Utility. Section 5 describes a simple *input perturbation* technique, in which noise from a spherically symmetric distribution (such as a Gaussian) is added to database points. The magnitude of the noise added to a real database point is a function of the distance to the point's t-th nearest neighbor. The intuition for this sanitization is two-fold. On one hand, we are "blending a point in with its crowd," so privacy should be preserved. On the other hand, points in dense regions should be perturbed much less than points in sparse regions, and so the sanitization should allow one to recover a lot of useful information about the database, especially information about clusters and local density.

We formalize the intuition about utility via two results. First, we show a worst-case result: an algorithm that approximates the optimal clustering of the sanitized database to within a constant factor gives an algorithm that approximates the optimal clustering of the real database to within a constant factor. Second, we show a distributional result: if the data come from a mixture of Gaussians, then this mixture can be learned from the perturbed data. Our algorithmic and analysis techniques necessarily vary from previous work on learning mixtures of Gaussians, as the noise that we add to each point depends on the sampled data itself.

The intuition about privacy—namely, that this style of perturbation blends a point in with its crowd—is significantly harder to turn into a proof. We explain why privacy for this type of sanitization is tricky to reason about, and describe some simplistic settings in which partial results can be proven.

Privacy of Perturbed Data via Cross-Training. In Section 6 we describe a variant for which we can again prove privacy when the distribution is uniform over the d-dimensional unit cube. The idea is to use cross-training to get the desirable properties of histograms and spherical perturbations. The real database points are randomly partitioned into two sets, A and B. First, a recursive histogram sanitization is computed for set B. As stated above, this information can be released safely. Second, for each point in set A we add random spherical noise whose magnitude is a function of the histogram for B (it is based on the diameter of the B-histogram cell into which the point falls). We release the histogram for set B and the perturbed points from set A. Since the only information about the first set used for the perturbation is information provably safe to reveal, the privacy of the points in the first set is not compromised. An additional argument is used to prove privacy of the points in the second set. The intuition for the utility of the spherical perturbations is the same as before: points which lie in dense regions will lie in small histogram cells, and so will be perturbed little, thus preserving clusters approximately.

Open Questions. Our work suggests a rich set of fascinating questions, several of which we have begun exploring together with other researchers. We mention some of these in Section 7.

1.2 Related Work

We briefly highlight some techniques from the literature. Many additional references appear in the full paper (see the title page of this paper for the URL).

Suppression, Aggregation, and Perturbation of Contingency Tables. Much of the statistics literature is concerned with identifying and protecting sensitive entries in contingency tables (see, e.g., [12, 22]). For example, in the 2-dimensional case, for discrete data, these are frequency counts, or histograms, indicating how many entries in the database match each possible combination of the values of two particular attributes (e.g. number of cars and number of children). It is common to regard an entry as sensitive if it corresponds to a pair of attribute values that occurs at most a fixed number of times (typically 1 or 2) in the database. One reason for regarding low-count cells as sensitive is to prevent linkage with other databases: if a given pair of attribute values uniquely identifies an individual, then these fields can be used as a key in other databases to retrieve further information about the individual.

Input Perturbation. A second broad approach in the literature is to perturb the data (say via swapping attributes or adding random noise) before releasing the entire database, or some subset thereof (such raw, unaggregated entries are typically called *microdata*). Various techniques have been considered in the statistics [34, 35, 28] and data mining [4, 1, 16] communities. In some cases, privacy is measured by how (un)successfully existing software re-identifies individuals in the database from the sanitized data and a small amount of auxiliary information. In other cases, the notion of privacy fails to take into account precisely this kind of auxiliary information. The work of Efvimievski et al. [16] is a significant, encouraging exception, and is discussed below.

Imputation. A frequent suggestion is for the census bureau to learn $\mathcal{D}$ and then publish artificial data obtained by sampling from the learned distribution (see, e.g., [29]). We see two difficulties with this approach: (1) we want our sanitized database to reflect (possibly statistically insignificant) "facts on the ground". For example, if a municipality is deciding where to run a bus line, and a certain geographic region of the city has a higher density of elderly residents, it may make sense to run the bus line through this region. Note that such "blips" in populations can occur even when the underlying distribution is uniform; (2) any model necessarily eliminates information; we feel it is not reasonable to assume that the sanitizer can predict (privacy-respecting) statistical tests that may be invented in the future.

k-Anonymity, Input Aggregation, and Generalization. Similarly to input perturbation, one can also suppress or aggregate fields from individual records to reduce the amount of identifying information in the database. A database is said to be k-anonymized if every modified entry in the sanitized database is the same as at least k others [32]. The intuition is that privacy is protected in this way by

guaranteeing that each released record will relate to at least k individuals. This requirement on the sanitization does not directly relate to what can and cannot be learned by the adversary. For example, the definition may permit information to be leaked by the choice of which records to aggregate (there are many aggregations that will make a database k-anonymous), or from the fact that certain combination of attribute values does not exist in the database. Information may also be gleaned based on the underlying distribution on data (for example, if the suppressed attribute is sex and the number of identical records with sex suppressed is two).

Interactive Solutions. In *query monitoring,* queries to an online database are audited to ensure that, even in the context of previous queries, the responses do not reveal sensitive information. This is sometimes computationally intractable [25], and may even fail to protect privacy, for example, in the setting in which the adversary knows even one real database record [11].

A related approach is *output perturbation*, in which a query control mechanism receives queries, computes exact answers, and then outputs a perturbed answer as the response to the query [9, 5]. This approach can sometimes be insecure, intuitively, because noise added in response to multiple queries can cancel out (see [2, 11]). The limitation can be shown to be inherent: if the number of queries to the database is large (even polynomial in the number of entries (rows) in the database), the amount of noise added to answers must be large [11]. By restricting the total number of queries allowed, one can in fact circumvent this and get strong privacy guarantees while adding much less noise [11, 15]. This approach is not available in our context: in an interactive solution, the query interceptor adds fresh noise to the response to each query; in our context, a noisy version of the database is constructed and published once and for all. Although this seems to make the problem more difficult, there are obvious advantages: the sanitization can be done off-line; the real data can be deleted or locked in a vault, and so may be less vulnerable to bribery of the database administrator.

A Recent Definitional Approach. The definitions of privacy in [11, 15, 16] (written concurrently with this work – see [13]) are consonant with our point of view, in that they provide a precise, meaningful, provable guarantee. All three follow the same paradigm: for every record in the database, the adversary's confidence in the values of the given record should not significantly increase as a result of interacting with or exposure to the database. The assumption is that the adversary can name individual records in the database; this captures the setting in which the database is multi-attribute, and the adversary has somehow, out of band, figured out enough about some individual to construct a query that effectively names the individual. Even in such a setting, it should be impossible to learn the value of even a single additional binary data field or the value of any predicate of the data tuple.

The work of Evfimievsky et al. [16] is in our model, ie, it describes a sanitization method (in this case, for transactions). Specifically, it is an input-perturbation technique, in which items are randomly deleted from and added to transactions. As we understand their work, both their specific technique and their definitions only consider applying the same, fixed, perturbation to each point in the database, independent of the other points in the database. Neither our definitions nor our techniques make this assumption. This both enhances utility and complicates privacy arguments.

Cryptographic Approaches. Much work in cryptography has focused on topics closely related to database privacy, such as private information retrieval and secure function evaluation (see, e.g., [18] and [20]). These problems are somewhat orthogonal to the one considered here. In secure function evaluation, privacy is preserved only to the extent possible given a specific functionality goal; but which functions are "safe" in the context of statistical databases? The literature is silent on this question.

2 A Formal Definition of Privacy

2.1 What do *We* Mean by "Privacy"?

As mentioned above, our notion of privacy breach is inspired by Gavison's writing on protection from being brought to the attention of others. This phrase articulates the common intuition that our privacy is protected to the extent that we blend in with the crowd. To convert this intuition into a precise mathematical statement we must abstract the concept of a database, formulate an adversary (by specifying the information to which it has access and its functionality), and define what it means for the adversary to succeed.

2.2 Translation into Mathematics

Under our abstraction of the database privacy problem, the real database (RDB) consists of n unlabeled points in high dimensional space $\mathbb{R}^d$, each drawn independently from an underlying distribution $\mathcal{D}$. Intuitively, one is one's collection of attributes. The census bureau publishes a sanitized Database (SDB), containing some n' points, possibly in a different space. This is a very general paradigm; it covers the case in which the SDB contains only summary information.

To specify security of a cryptographic primitive we must specify the power of the adversary and what it means to break the system. Since the goal of our adversary is to "single out" a record in the database, we call the adversary an *isolator*. The isolator takes two inputs – the sanitized database and auxiliary information (the auxiliary information is analogous to the history variable in the definition of semantic security). The isolator outputs a single point $q \in \mathbb{R}^d$. This completes the description of the functionality of the adversary. Note that the definition admits adversaries of unbounded computational power. Our results require no complexity-theoretic assumptions[5].

We next define the conditions under which the adversary is considered to have succeeded in isolating. The definition is parameterized with two values: a

[5] We do not object to using complexity-theoretic assumptions. We simply have not yet had a need to employ them.

privacy threshold t, intuitively, the size of the "crowd" with which one is supposed to blend in, and an *isolation parameter* c, whose use will be clear in a moment. Roughly, c helps to formalize "blending in".

For a given isolating adversary $\mathcal{I}$, sanitized database SDB, and auxiliary input z, let $q = \mathcal{I}(\mathrm{SDB}, z)$ ($\mathcal{I}$ may be a randomized algorithm). Let δ be the distance from q to the nearest real database point, and let x be an RDB point at distance δ from q. Let $B(p, r)$ denote a ball of radius r around point p. If the d-dimensional ball of radius $c\delta$ and centered at q contains at least t real database points, that is, if $|RDB \cap B(q, c\delta)| \geq t$, then the adversary *fails* to isolate x. Otherwise, the adversary succeeds.

We will give a slightly more general definition shortly. First we give some intuition for the definition. The adversary's goal is to single out someone (*i.e.*, some RDB point) from the crowd, formalized by producing a point that is much closer to some $x \in \mathrm{RDB}$ than to $t-1$ other points in the RDB. The most likely victim x is the RDB point closest to q. So q "looks something like" x. On the other hand, if $B(q, c\delta)$ contains at least t RDB points, then q also looks almost as similar to lots of (*i.e.*, $t-1$) other RDB points, so x hasn't really been singled out.

Note that the definition of isolation is a relative one; the distance requirement for success varies according to the local density of real database points. This makes sense: if we name an intersection in New York City, there are perhaps a few hundred people living at or very near the intersection, so our point is "close to" many points (people) in the RDB. In contrast, if we name an intersection in Palo Alto, there are perhaps 10 people living near the intersection[6]. More generally, we have the following definition:

Definition 1. ((c, t)-isolation) *Let y be any RDB point, and let $\delta_y = \|q - y\|$. We say that q* (c, t)-isolates *y if $B(q, c\delta_y)$ contains fewer than t points in the* RDB*, that is, $|B(q, c\delta_y) \cap \mathrm{RDB}| < t$.*

We frequently omit explicit mention of t, and speak of c-isolation. It is an easy consequence of the definitions that if $q = \mathcal{I}(\mathrm{SDB}, z)$ fails to c-isolate the nearest RDB point to q, then it fails to c-isolate even one RDB point.

For any point p (not necessarily in the RDB), we let τ_p be the minimum radius so that $B(p, \tau_p)$ contains t RDB points. We call this the *t-radius of p.*

3 Definition of Sanitization

Suppose one of us publishes all our information on the web — that is, we publish our RDB point x where the adversary can find it — so that the point is part of the adversary's auxiliary information. Clearly, the adversary can isolate x by setting $q = x$ (in which case $\delta_x = 0$ and $B(q, c\delta_x)$ contains only x — we assume no two points are identical). It seems unfair to blame the sanitizing procedure for this isolation; indeed, there is an adversary simulator that, without access to the

[6] This analogy was suggested by Helen Nissenbaum.

sanitized database, can also isolate x, since x is part of the auxiliary information. We are therefore concerned with how much seeing the sanitized database helps the adversary to succeed at isolating even one RDB point. Intuitively, we do not want that seeing the SDB should help "too much". Our notion of "too much" is fairly relaxed. Letting ε denote the probability that isolation may occur, we tend to think in terms of, say, $\varepsilon = 1/1000$. This says that about one out of every 1,000 sanitized databases created may be vulnerable to an isolation event[7]. The parameter ε can be a function of d and n. Note, however, that ε cannot depend only on n – otherwise privacy could be improved simply by the introduction of additional points.

More formally, a *database sanitizer*, or simply *sanitizer* for short, is a randomized algorithm that takes as input a real database of some number n of points in $\mathbb{R}^d$, and outputs a sanitized database of some number n' of points, in a possibly different space $\mathbb{R}^{d'}$.

A sanitizer is *perfect* if for every distribution $\mathcal{D}$ over $\mathbb{R}^{n \times d}$ from which the real database, for all isolating adversaries $\mathcal{I}$, and points are drawn, there exists an adversary simulator $\mathcal{I}'$ such that with high probability over choice of RDB, for all auxiliary information strings z, the probability that $\mathcal{I}(\mathrm{SDB}, z)$ succeeds minus the probability that $\mathcal{I}'(z)$ succeeds is small. The probabilities are over the coin tosses of the sanitization and isolation algorithms. We allow the sanitizer to depend on the parameters c, t, and also allow $\mathcal{I}, \mathcal{I}'$ to have access to $\mathcal{D}$.

More precisely, let ε be a parameter (for example, $\varepsilon = 2^{-d/2}$). We require that for all $\mathcal{I}$ there exists an $\mathcal{I}'$ such that, if we first pick a real database $\mathrm{RDB} \in_R \mathcal{D}$, then with overwhelming probability over RDB, for all z,

$$\forall S \subseteq \mathrm{RDB}\, |Pr[\exists x \in S : \, I(SDB, z) \text{ isolates } x] - Pr[\exists x \in S : \, I'(z) \text{ isolates } x]| < \varepsilon$$

where the probabilities are over the choices made by $\mathcal{I}$, $\mathcal{I}'$, and the sanitization algorithm.

Remark 1. The use of the sets S in the definition is for the following reason. Suppose the adversary knows, as part of its auxiliary information, some point $y \in \mathrm{RDB}$. For every $x \neq y \in \mathrm{RDB}$, we want x's chances of being isolated to remain more or less unchanged when the adversary is given access to the sanitized database. Thus, if we were to write the more natural $|\Pr[\exists x \, \mathcal{I}(\mathrm{SDB}, z) \text{ isolates } x] - \Pr[\exists x \, \mathcal{I}'(z) \text{ isolates } x]| < \varepsilon$, then we might have $z = y \in \mathrm{RDB}$, $\mathcal{I}'(z) = y$, and $\mathcal{I}(\mathrm{SDB}, z)$ could (somehow) isolate a different point $x \neq y \in \mathrm{RDB}$ with probability one. This is clearly unsatisfactory.

This is excessively ambitious, and in fact a nontrivial perfect sanitizer does not exist [14]. However, by specifying the ideal we can begin to articulate the value of specific sanitization techniques. For a given technique, we can ask what can be proved about the types of distributions and auxiliary information for which it can ensure privacy, and we can compare different techniques according to these properties.

[7] The adversary has no oracle to tell it when it has succeeded in isolating an RDB point.

4 Histograms

Consider some partition of our space $\mathbb{R}^d$ into disjoint cells. The histogram for a dataset RDB is a list describing how many points from the dataset fall into each cell. A sanitization procedure is *histogram-based* if it first computes the histogram for the dataset and bases the output only on that information.

For example, in one dimension the cells would typically be sub-intervals of the line, and the histogram would describe how many numbers from the dataset fall into each of the intervals. In higher dimensions, the possibilities are more varied. The simplest partition divides space into cubes of some fixed side-length (say at most 1). That is, each cell is a cube $[a_1, a_1 + 1] \times \cdots \times [a_d, a_d + 1]$ for integers $a_1, ..., a_d$.

Our principal result for histograms is that, if the original data set RDB consists of n points drawn independently and uniformly from some large cube $[-1, 1]^d$, then the following sanitization procedure preserves privacy:

> **Recursive Histogram Sanitization:** Divide the cube into 2^d equal-sized subcubes in the natural way, *i.e.*, by bisecting each side at the midpoint. Then, as long as there exists a subcube with at least $2t$ points from RDB, further subdivide it into 2^d equal-sized cubes. Continue until all subcubes have fewer than $2t$ points, and release the exact number of points in each subcube.

Theorem 1. *Suppose that* RDB *consists of n points drawn i.i.d. and uniformly from the cube $[-1, 1]^d$. There exists a constant c_{secure} (given in Lemma 2) such that the probability that an adversary, given a recursive histogram sanitization as described above, can c_{secure}-isolate an* RDB *point is at most $2^{-\Omega(d)}$.*

The proof is quite robust (in particular, the error bound does not depend on n). Using the techniques developed in this section, several variants on the partitioning described above can be shown to preserve privacy, even under less rigid assumptions about the underlying distribution. Thus, our goals are to prove privacy of histogram sanitizations, to illustrate techniques that are useful for proving privacy, and to establish a result which we will need when we deal with cross-training-based sanitizers later on.

The technical heart of the proof of Theorem 1 is following proposition:

Proposition 2 *Suppose that the adversary knows only that the dataset consists of n points drawn i.i.d. from the cube $[-1, 1]^d$, and that we release the exact histogram (cell counts) for the natural partition of this cube into 2^d subcubes of side-length 1. Then the probability that the adversary succeeds at c-isolating a point for $c > 121$ is at most $2^{-\Omega(d)}$, as long as $t = 2^{o(d)}$.*

The constant 121 in the proposition can in fact be improved significantly, to approximately 30, with minor changes to the proofs in this section.

This result is strong—it essentially states that for any point q which the adversary might produce *after* seeing the histogram, the distance to q's nearest neighbor is at most a constant less than the distance between q and its $2^{o(d)}$-th nearest neighbor. When $n = 2^{o(d)}$, the result is perhaps less surprising: the distance between q and its nearest neighbor is $\Omega(\sqrt{d})$ with high probability, and $2\sqrt{d}$ is an upper bound on the distance from q to its farthest neighbor (assuming q is in the large cube $[-1, 1]^d$). For very large values of n (say $2^{\Omega(d)}$), the proof becomes much more involved.

Remark 2. We would like to understand the probability that the adversary isolates a point after seeing the sanitization, given reasonable assumptions about the adversary's a priori view of the database. Currently we assume that the underlying distribution is uniform on a d-dimensional hypercube. The following example shows that such a "smoothness" condition is necessary to obtain a bound on the adversary's probability of success, when a histogram of the data is released.

Consider the following distribution. In each of the 2^d subcubes of the hypercube, there is an infinite sequence of points $p_1, p_2, \ldots$. The probability density at point p_i is $\frac{1}{2^d}\frac{1}{2^i}$. That is, each subcube has equal mass, and within a subcube, mass is distributed over the infinite sequence of points in an exponentially decreasing manner. Now, if the adversary knows the number of points in a subcube, say m, then, she produces the point $q = p_{\log m}$ in that subcube. With a constant probability, there are at least one, but no more than t, points at q, and the adversary succeeds. On the other hand, without knowledge of the number of points in each subcube (as given by the histogram), the adversary simulator I' has an exponentially low probability of succeeding.

The next subsections sketch the proof of Proposition 2. The full version of the paper contains the details of the proof, as well as extensions to cover finer subdivisions of the cube and the recursive sanitization described above.

4.1 Simplifying the Adversary

We distinguish two related definitions of isolation. The adversary is always given the sanitized database SDB as input (the adversary may also receive side information about the real database—typically, in our setting, the distribution from which the points in the real database are drawn).

- A **ball adversary** produces a pair (q, r) where $q \in \mathbb{R}^n$ is a point in space and $r \in \mathbb{R}^+$ is a non-negative real number. The adversary succeeds if $B(q, r)$, the ball of radius r centered at q, contains at least one point in RDB, but $B(q, cr)$ contains fewer than t points in RDB (equivalently, $r < \tau_q/c$).
- A **point adversary** produces only a point $q \in \mathbb{R}^n$. The adversary succeeds at c-isolation if there is a point in D within distance τ_q/c of q, i.e. if there exists some r for which the corresponding ball adversary would have won.

We first prove Proposition 2 for ball adversaries since their behavior is easier to understand. At the end of the proof, we show that point adversaries do essentially no better than ball adversaries in this context, and so the restriction to ball adversaries is made without loss of generality.

4.2 Proof Sketch for Proposition 2

We sketch here the proof of Proposition 2. Recall that the points in the real database RDB are drawn uniformly from the large cube $[-1,1]^d$, and the sanitization consists of the number of points from RDB contained in each of the cubes obtained by dividing $[-1,1]^d$ once along each dimension. That is, a cell C is a d-dimensional hypercube of side-length and volume 1, which has one vertex at $(0,0,...,0)$ and the opposite vertex in the set $\{-1,+1\}^d$. The total number of points in the database is n, and we denote the number of points appearing in a cell C by n_C. The sanitization is simply the list of all 2^d values n_C.

Define a function $f : \mathbb{R}^n \to \mathbb{R}^+$ which captures the adversary's view of the data.

$$f(x) = \frac{n_C}{n} \cdot \frac{1}{\mathsf{Vol}(C)} \quad \text{for } x \in C. \tag{1}$$

The function f is a probability density function. The adversary does not see the data as being drawn i.i.d. according to f, but the function is useful nonetheless for bounding the adversary's probability of success.

Lemma 1. *If a ball adversary succeeds at c-isolation with probability* ε*, then there exists a pair* (q,r) *such that* $\Pr_f[B(q,r)] \geq \varepsilon/n$ *and* $\Pr_f[B(q,cr)] \leq (2t + 8\log(1/\varepsilon))/n$*.* [8]

The intuition for the proof of Lemma 1 is simple: it is sufficient to show that if one considers only the number of points landing in a particular region, there is almost no difference between the adversary's view of RDB and a set of n points sampled i.i.d. from f.

Roughly, Lemma 1 means that for a ball adversary to succeed, a necessary (but not sufficient) condition is that:

$$\frac{\Pr_f[B(q,r)]}{\Pr_f[B(q,cr)]} \geq \frac{\varepsilon/n}{(2t + 8\log(1/\varepsilon))/n} = \varepsilon/(2t + 8\log(1/\varepsilon)). \tag{2}$$

This means that it is sufficient to bound the ratio on the left-hand side above by some negligible quantity to prove privacy against ball adversaries (in fact, the upper bound need only hold as long as $\Pr_f[B(q,r)]$ is itself not too large). The better the bound on the ratio in Eqn. (2), the better the result on privacy. To get the parameters described in the statement of Proposition 2, it will be sufficient to prove a bound of $2^{-\Omega(d)}$ for the ratio: we can then think of ε as $2^{-\gamma d}$, for a constant $\gamma < 1$, and think of t as being $2^{-o(d)}$.

The upper bound, and the proof of Proposition 2, rely on the following lemma:

Lemma 2. *There is a constant* $1/60 < \beta < 1$ *such that, for any point* q*, radius* $r > 0$ *and cell* C *for which* $B(q,r) \cap C \neq 0$*, we have:*

[8] We are assuming that n is at least $2t + 8\log(1/\varepsilon)$. The assumption is not necessary, but simplifies the proofs. In fact, when n is small one can use completely different proofs from the ones described here which are much simpler.

1. *If $r \leq \beta\sqrt{d}$, then*

$$\frac{\mathsf{Vol}(B(q,r)\cap C)}{\mathsf{Vol}(B(q,3r)\cap C)} \leq 2^{-\Omega(d)}. \tag{3}$$

2. *If $r > \beta\sqrt{d}$, then for all cells C' neighboring C:*

$$C' \subseteq B(q, c_{\text{secure}} r) \tag{4}$$

where $c_{\text{secure}} \leq (2/\beta)+1$. A neighboring cell is a cube of the same side-length which shares at least one vertex with C.

Remark 3. The value of the constant β can be made much larger, at least 1/15. Obtaining this bound requires more careful versions of the proofs below.

A detailed proof is in the full version. Roughly, to prove Part 1, we need to estimate the probability that a point chosen from a ball lies is a cube. To this end, we approximate sampling from a ball by sampling from an appropriately chosen spherical Gaussian. This allows us to analyze behavior one coordinate at a time. Our (rather involved) analysis only holds for radii $r \leq \beta\sqrt{d}$. It is possible that a different analysis would yield a better bound on β and hence on β.

Part 2 is much simpler; it follows from the fact that the diameter of the cube $[-1,1]^d$ is $2\sqrt{d}$.

We can now prove Proposition 2, which states that releasing the histogram preserves privacy, with the adversary's success probability bounded by $2^{-\Omega(d)}$. We first give a proof for ball adversaries, and then observe that (almost) the same proof works for point adversaries too.

Proof (of Proposition 2). Ball adversaries: Assume there is a ball adversary who chooses the best possible pair (q,r) based on SDB.

First, suppose that $r \leq \beta\sqrt{d}$ (the constant is from Lemma 2). In that case, we will actually show that 3-isolation (not 121-isolation!) is possible only with very small probability. Our proof relies on Part 1 of the lemma. We can write the mass of $B(q,r)$ under f as a weighted sum of the volume of its intersections with all the possible cubes of C:

$$\Pr_f[B(q,r)] = \sum_C \frac{n_C}{n} \cdot \mathsf{Vol}(B(q,r)\cap C)$$

We can bound each of these intersections as an exponentially small fraction of the mass of $B(q,3r)$:

$$\Pr_f[B(q,r)] \leq \sum_C \frac{n_C}{n} \cdot 2^{-\Omega(d)} \cdot \mathsf{Vol}(B(q,3r)\cap C) = 2^{-\Omega(d)} \cdot \Pr_f[B(q,3r)]$$

Now the mass of $B(q,3r)$ is at most 1, which means that the ratio in Eqn. (2) is at most $2^{-\Omega(d)}$, and so $\varepsilon/(2t + 8\log(1/\varepsilon)) \leq 2^{-\Omega(d)}$. This is satisfied by $\varepsilon = 2^{-\Omega(d)}$ (for essentially the same constant in the Ω-notation), and so in this case, 3-isolating the point q is possible only with probability $2^{-\Omega(d)}$.

Now consider the case where $r > \beta\sqrt{d}$. If $B(q, r)$ doesn't intersect any cells C, then we are done since the ball captures a point with probability 0. If there is a cell C which intersects $B(q, r)$, then, by Part 2 of Lemma 2, $B(q, c_{\text{secure}} r)$ (for $c_{\text{secure}} \leq (2/\beta) + 1$) contains all the cells C' which are neighbors to C, in particular all of $[-1, 1]^d$. (Recall that our points are initially uniform in $[-1, 1]^d$, and we consider a subdivision which splits the cube into 2^d axis-parallel subcubes of equal size). The adversary succeeds with probability zero at c_{secure}-isolation, since all n points will be within distance $c_{\text{secure}} r$.

Point adversaries: Suppose the adversary outputs a particular point q, and let r be the smallest radius such that $\Pr_f[B(q, r)] = \varepsilon$. By the previous discussion, $B(q, c_{\text{secure}} r)$ contains mass at least $(2t + 8\log(1/\varepsilon))/n$. Thus, with probability at least $1 - 2\varepsilon$, there is no point inside $B(q, r)$ and there are t points inside $B(q, c_{\text{secure}} r)$ (by the proof of Lemma 1). The ratio between the distances to the t-th nearest point and to the nearest point to q is then at most $c_{\text{secure}} r/r = c_{\text{secure}}$. The point adversary succeeds at c_{secure}-isolating a point with probability at most 2ε.

Because $\beta > 1/60$, the constant c_{secure} is at most 121, and so the adversary fails to 121-isolate any point in the database. □

Remark 4. The proof technique of this section is very powerful and extends in a number of natural ways. For example, it holds even if the adversary knows an arbitrary number of the points in the real database, or (with a worse isolation constant), if the adversary knows a constant fraction of the attributes of a database point. The analysis holds if the underlying distribution is a mixture of (sufficiently separated) hypercubes.

Recent work also indicates that histogram sanitization, at least to a limited depth of recursion, can be constructed for "round" distributions such as the sphere or the ball [6]. Together, these techniques yield privacy for sufficiently separated mixtures of round and square distributions.

5 "Round" Perturbation Sanitizations

Perturbation via additive noise is a common technique in the disclosure control literature. In this section, we present a variant on this technique in which the magnitude of the noise added to a point depends on the local density of the database near the point. We consider three perturbation sanitizers that are very similar when the dimension d is large. In these sanitizers, $d' = d$ and $n' = n$ (that is, the sanitized database consists of points in the same space as the real database, and the numbers of points in the real and sanitized databases are identical). As before, let $B(p, r)$ denote the ball of radius r around p, let $S(p, r)$ denote the corresponding sphere, or the surface of $B(p, r)$. Let $\mathcal{N}(\mu, \sigma^2)$ denote a d-dimensional Gaussian with mean μ and variance σ^2 in every dimension. For $x \in \mathbb{R}^d$, the t-radius τ_x is the minimum radius such that $B(x, \tau_x)$ contains t RDB points (x need not be an RDB point.)

1. **The Ball Sanitizer:** For $x \in RDB$, $\text{BSan}(x, RDB) \in_R B(x, \tau_x)$.

2. **The Sphere Sanitizer:** For $x \in RDB$, $\mathrm{SSan}(x, RDB) \in_R S(x, \tau_x)$.
3. **The Gaussian Sanitizer:** For $x \in RDB$, $\mathrm{GSan}(x, RDB) \in_R \mathcal{N}(x, \tau_x^2/d)$.

We will refer to these as *round*, or *spherical* sanitizers, because of the shape of the noise distribution. The intuition for these sanitizations is three-fold: we are blending a point in with a crowd of size t, so privacy should be preserved; points in dense regions should be perturbed much less than points in sparse regions, and so the sanitization should allow one to recover a lot of useful information about the database, especially information about clusters and local density; we are added noise with mean zero, so data means should be preserved.

Round sanitizations have been studied before, typically with independent, identically distributed noise added to each point in the database. This approach implicitly assumes that the density of the data is more or less uniform in space for the entire data set. Even with data drawn i.i.d. from a uniform distribution on a fixed region, this need not be the case. Indeed, Roque [28] showed that (in low dimensions) re-identification software defeats this i.i.d. spherical noise, though the standard packages fail if the noise is not spherical (say, drawn from from a mixture of Gaussians). Kargupta et al. [24] argue that independent additive perturbation may have limited application to preserving privacy insofar as certain informative features of the data set (e.g.: the principal components) are largely unaffected by such perturbations. Their argument assumes (critically) that the sanitizer applies a fixed distribution to each element, and ultimately describes how to reconstruct the covariance matrix of the *attributes*. In this work, we apply data-driven distributions to the elements, and prove privacy guarantees for the *individuals*. Moreover, we conjecture it is possible to exploit what Kargupta et al. perceive as a weakness (reconstructing the covariance matrix, which we can do), while provably maintaining privacy (which we conjecture)[9]. Finally, the data-dependent noise distribution provides more potential functionality than a fixed noise distribution [4, 1, 16], at the cost of a more difficult analysis.

5.1 Results for Round Sanitizers

We have obtained several results on the privacy and utility of round sanitizations. Our most powerful result is concerned with learning a mixture of Gaussians from the sanitized data. This result is of independent interest, and is described in the Section 5.2. We first summarize our results.

Utility. The task of extracting information from a database whose points have been spherically perturbed is essentially one of learning from noisy data. Standard techniques do *not* apply here, since the noise distribution actually depends on the data. Nonetheless, we prove two results using the intuition that round perturbations preserve expectations (on average) and that our particular strategy is suited to clustering.

[9] Specifically, given recent results in constructing histograms for round distributions [6], we conjecture it will be possible to obtain cross-training results for mixtures of Gaussians, analogous to our cross-training results for the hypercube described in Section 6 below.

1. When the data are drawn uniformly from a mixture of Gaussians $\mathcal{D}$, there is an efficient algorithm that learns $\mathcal{D}$ from the Gaussian sanitization. Learning mixtures of Gaussians has already been heavily investigated [3, 8, 33], however existing analyses do not apply in our setting. The algorithm and its analysis are sketched in Section 5.2.
2. For any distribution, suppose we are given an algorithm to find k clusters, each of cardinality at least t, minimizing the maximum diameter of a cluster, and assume the data are sanitized with either BSan or SSan. Then running the algorithm on the sanitized data does a good job of clustering the original data. More precisely, any algorithm that approximates the optimal clustering of the sanitized database to within a constant factor gives an algorithm that approximates the optimal clustering of the real database to within a constant factor, and the maximum diameter of a cluster exceeds the maximum diameter of an optimal k-clustering on the RDB by at most a factor of 3.

Privacy. The intuition for privacy is significantly harder to turn into a complete proof than is the one for utility. We analyze two special cases, and give a lower bound showing that high-dimensionality is necessary for the privacy of this type of sanitization. The proofs of the results below appear in the full version of the paper.

1. The database consists of only two points, x and y, which are sanitized with respect to each other, and the underlying distribution is the unit sphere in d dimensions. That is, $t = 2$ and each of x and y is perturbed using SSan with perturbation radius $\|x - y\|$. The adversary is given $\|x - y\|$, and the sanitizations x' and y'. We show that the probability of 4-isolation is exponentially small in d, with overwhelming probability over the choice of x and y. The proof is by symmetry: we construct many pairwise-distant "decoy" pairs x', y' which are equiprobable in the adversary's view.
2. The real database consists of n sanitized points, drawn from the d-dimensional unit sphere. The adversary is given all but one point in the clear, together with a sanitization of the final point using SSan. The adversary's goal is to 4-isolate the last point. Intuitively, privacy holds because the hidden point can lie in any direction from its sanitization, while any point q produced by the adversary can only isolate points lying in an exponentially small fraction of these directions. The result is proved for $t = 2$.
3. Sanitization cannot be made arbitrarily safe: for any distribution, if sanitization is done using BSan, then there is a polynomial time adversary $\mathcal{I}$ requiring no auxiliary information, such that the probability that the adversary succeeds is $\Omega(\exp(-d)/\log(n/t))$.

Remark 5. The second result above highlights the delicacy of proving privacy for this type of sanitization. Contrary to intuition, it is *not* the case that seeing the sanitizations of the remaining $n-1$ points, rather than their exact values, gives less information. The reason is that the sanitization of y, implicitly contains information about the t-neighborhood of y. This sort of dependency is notoriously hard to deal with in cryptography, e.g. in the *selective decommitment problem.* We have not yet proved or disproved the viability of the above-mentioned sanitizations; instead we circumvent the difficulty via cross-training.

5.2 Learning Mixtures of Gaussians

In this section we look at an algorithm for mining sanitized data. We address the well-studied problem of learning a mixture of Gaussians, with the twist that the samples have been sanitized using one of the round sanitizers discussed above. The distribution that results from the sanitization is no longer a mixture of Gaussians (samples are not even independent!) and traditional algorithms for learning mixtures of Gaussians do not apply. Nonetheless, we will see that the core properties that make Gaussian mixtures learnable remain intact, and prove that the mixtures can be read from an optimal low rank approximation.

We assume there are k mean vectors μ_i, each with an associated mixing weight w_i. Let $w_{\min}$ denote the minimum of the mixing weights. Each point in the data set is independently produced by selecting a μ_i with probability proportional to the w_i, and applying independent, normally distributed noise to each coordinate. We assume that the Gaussians are spherical, in that every Gaussian has an associated variance that is used for each of the coordinates. Let σ_1^2 denote the maximum such variance. We assume that the sanitization process amounts to applying an additive perturbation established by choosing a point uniformly at random from the unit sphere, which is then scaled in length by a random variable at most $2\sigma_2\sqrt{d}$, where σ_2 may *depend on the sampled data and* t. Notice that this is sufficiently general to capture all the perturbation based sanitizations described above – SSan, BSan, and GSan – the latter two using a random variable for the scaling factor.

For the purposes of analysis, we assume that we have access to two data sets $\overline{A}$ and $\overline{B}$ that have been independently sanitized. Each is assumed to result from the same underlying set of means, and to be independent of the other. We use $\overline{A}_u$ to denote the sanitized vector associated with u, $\widehat{A}_u$ the unsanitized vector associated with u, and A_u the original mean vector associated with u. We form the matrices $\overline{A}$, $\widehat{A}$, and A by collecting these columns for each point in the data set. Let w_u denote the mixing weight associated with the Gaussian with mean A_u. The matrices $\overline{B}$, $\widehat{B}$, and B and their associated columns are analogous, though they represent an independently sanitized disjoint data set. While this setting is not difficult for the sanitization process to accommodate, the motivation is simply for the clarity of analysis and it is unclear whether disjoint data sets are necessary in practice.

The main linear algebraic tool that we use is a matrix's *optimal rank k projection.* For every matrix M, this is a projection matrix P_M, such that for all rank k matrices D, we have $\|M - P_M M\|_2 \leq \|M - D\|_2$. Computing the optimal projection is not difficult; in most cases it is an $O(dn\log(dn))$ operation. We also make use of the *single linkage* clustering algorithm [30]. For our purposes, given a collection of points, single linkage repeatedly inserts an edge between the closest pair of non-adjacent points until the resulting graph has k connected components. Our actual algorithm can be stated succinctly:

$\mathsf{Cluster}(\overline{A}, \overline{B}, k)$
1. Compute $P_{\overline{A}}$ and $P_{\overline{B}}$, and form $C = [P_{\overline{B}}\overline{A}|P_{\overline{A}}\overline{B}]$.
2. Apply single linkage to C, forming k clusters.

Cluster takes a pair of data sets, and uses the structure define by each data set to filter the noise from the points in the other. If the mean vectors μ_i have sufficient separation, all inter-cluster distances in C will exceed all intra-cluster distances in C, and single linkage will associate exactly those points drawn from the same mean.

Theorem 3. *Assume that $d < n/2$. If for each pair of means μ_i, μ_j,*

$$\|\mu_i - \mu_j\| \geq 4(\sigma_1 + \sigma_2)(16/w_{\min}^{1/2} + \sqrt{k\log(kn/\delta)})$$

then with probability $1 - 2(e^{-nw_{\min}/8} + 2^{-\log^6 n/2} + \delta)$, Cluster *partitions the columns of $\overline{A}$ and $\overline{B}$ according to the underlying Gaussian mixture.*

Proof. The proof is conducted by bounding $\|\mu_u - C_u\|$ for each u. Assume, without loss of generality, that $\mu_u = A_u$ and $C_u = P_{\overline{B}}\overline{A}_u$. Notice that

$$\|\mu_u - C_u\| = \|A_u - P_{\overline{B}}\overline{A}_u\| \leq \|A_u - P_{\overline{B}}A_u\| + \|P_{\overline{B}}A_u - P_{\overline{B}}\overline{A}_u\|$$

In Lemmas 3 and 4 below, we bound these two terms, so that their sum is at most 1/4 the assumed separation of the mean vectors. With such separation, all inter-cluster distances are at least 1/2 the mean separation, and all intra-cluster distances are at most 1/2 the mean separation.

Although the above result requires a uniform bound on the pairwise separation between the means of the clusters, by using a more sophisticated clustering algorithm than Single-Linkage on the low-dimensional projection of the data, we can improve the results such that the requisite separation between a pair of means depends only on the variances of the corresponding Gaussians.

Lemma 3 (Systematic Error). *Assume $d < n/2$. With probability at least $1 - (e^{-nw_{\min}/8} + 2^{-\log^6 n/2})$, for all u*

$$\|(I - P_{\overline{B}})A_u\| \leq 16(\sigma_1 + \sigma_2)/w_u^{1/2}$$

Proof. (Sketch) In previous work using cross training techniques, [26] shows that

$$\|(I - P_{\overline{B}})A_u\| \leq 4\|B - \overline{B}\|_2/nw_u$$

To continue, [26] uses a result of Furedi and Komlos [17], which places a high probability bound on the norm of zero mean random matrices with independent entries of bounded variance. While $B - \overline{B}$ does not have independent entries –

the sanitization process depends on the result of the random events producing $\widehat{B}$ – each of the matrices $B-\widehat{B}$ and $\widehat{B}-\overline{B}$ *do*. As such, we get that with probability at least $1-2^{-\log^6 n/2}$

$$\|B-\overline{B}\|_2 \;\; \leq \;\; \|B-\widehat{B}\|_2 + \|\widehat{B}-\overline{B}\|_2 \;\; \leq \;\; 4(\sigma_1+\sigma_2)\sqrt{n}$$

Lemma 4 (Random Error). *For each u,*

$$Pr[\|P_{\overline{B}}(A_u - \overline{A}_u)\| > c(\sigma_1+\sigma_2)\sqrt{k}] \leq 2ke^{-c^2}$$

6 Combining Histograms and Perturbations: Cross-Training

We are drawn to spherical sanitizations because of their apparent utility (see the discussion in Section 5.1). However, as noted in Remark 5, we have some concerns regarding the privacy offered: it is not the privacy of the perturbed points that concerns us, but the privacy of the points in the t-neighborhood of the perturbed points (since the sanitization radius itself leaks information about these points). In this section, we combine a histogram-based sanitization with a spherical sanitization to obtain a provably private spherical sanitization for $n = 2^{o(d)}$ points, (again in the absence of auxiliary information).

We randomly divide the dataset into two sets — A and B[10]. We construct a recursive histogram on B (as in Section 4). We then sanitize points in A using only their position in the histogram on B. We release the sanitizations of points in A, along with the exact count, for every cell in the histogram, of points in A and B lying in that cell. We also assume that the adversary knows for every sanitized point $v' \in$ SDB, the cell in the histogram that its pre-image $v \in A$ lies in (this only helps the adversary).

For a point $v \in A$, let C be the cell containing v in the recursive histogram constructed for B. Let $P(C)$ be the parent cell of C. $P(C)$ has twice the side-length of C, and contains at least t points. Consider the following sanitization procedure:

> **Cross-Training Round Sanitization:** Let ρ_v be the side-length of the cube C. Select a point N_v at random from a spherical Gaussian which has variance ρ_v^2 in each coordinate. Output $v' = v + N_v$.

As shown above, this procedure protects the privacy of points in B since the information released about these points depends only on the recursive histogram of the set B. In this section we prove that it also protects the privacy of points in A, under the assumption that from the adversary's point of view, the *a priori* distribution of points in the database is uniform when restricted to the cell C.

Consider a particular point $v \in A$. Suppose the side-length of the cell C containing v is ρ_v. Lemma 5 below shows that with probability $1-2^{-\Omega(d)}$ over

[10] In fact, our proof only requires that A contain at most $2^{o(d)}$ points.

the choice of RDB and the coins of the sanitization algorithm, the following occurs: for any point q which the adversary might produce, the distance between q and v will be $\Omega(\rho_v\sqrt{d})$. Since the t-radius of v is $O(\rho_v\sqrt{d})$, this implies that adversary c-isolates v with probability at most $2^{-\Omega(d)}$ (for some constant c).

The result is quite useful. If A contains $2^{o(d)}$ points, then a union bound shows that with probability at least $1-2^{-(\Omega(d)-o(d))}$, the sanitization is "good": that is, the adversary can succeed at isolating some point with probability at most $2^{-\Omega(d)}$.

Below, we state the main lemma of this section; the proof, omitted for lack of space, is in the full version.

Lemma 5. *Suppose that v is uniformly distributed in the cube C, and q is the output of the adversary on input $v' = v + N_v$. There is a constant $\alpha < 9$ such that with probability $1-2^{-\Omega(d)}$, the ball $B(q,(\alpha+1)\|v-q\|)$ contains the entire parent cell $P(C)$. The probability is over choice of the real database* RDB *and the coins of the sanitization algorithm.*

7 Future Work

Isolation in few dimensions. Many have raised the case in which the adversary, studying the sanitized data, chooses a small set of attributes and outputs values that uniquely identify a point in the RDB (no other point in the RDB agrees well with the given point on this particular set of attributes). This may not be a privacy breach as we have defined it, since the adversary may have very bad luck at guessing the remaining attribute values, and therefore the point q that the adversary produces may not be particularly close to any point in the RDB. However, as M. Sudan has pointed out, the adversary may know the difference between the attributes on which it is guessing and the ones it has learned from the sanitized data.

We are uncertain exactly what it means for the adversary to "know" this difference. Our notion of privacy breach essentially says we don't care about such things: after all, releasing a histogram cell count of 1 says there is a unique individual in a certain subcube, but we prove that the advesary cannot isolate this individual. However, the question is provocative.

The attack corresponds imprecisely to identification of a short key for a population unique (see the discussion in Section 1.2). Alternatively, the adversary may know a key to a population unique and the worry is that the sanitization may permit the learning of additional attribute values. On the one hand, we note that our definition of a perfect sanitization precludes either of these possibilities: roughly speaking, if it were possible to learn the key to a population unique then there is a choice for the auxiliary information that would permit the remaining attribute values to be learned, which would constitute an isolation. On the other hand, we have already noted that perfect sanitizers cannot exist [14], and our privacy results have been proved, for the most part, without permitting the adversary auxiliary information.

With this in mind, one may extend the definition of isolation to allow the adversary to approximate a real point in only a few attributes. Note however, that as the number of attributes estimated by the adversary decreases, the notion of isolation must become more and more stringent. This corresponds to an increase in the parameter c in our definition of c-isolation.

This suggests the following extended definition. The adversary, upon receiving the SDB, outputs a k-dimensional axis-parallel hyperplane H ($k \leq d$), and a point q in this hyperplane. Let $\Pi_H(y)$ denote the projection of an RDB point y onto the hyperplane H. Let y be the RDB point which is closest to q under the projection Π_H. For a given function $\phi(k)$, we say that q **(ϕ, c, t)-isolates** y in H iff $\Pi_H(y)$ is $(\phi(k)c, t)$-isolated by q in the projected space H. Recursive histogram sanitizations are safe with respect to $(\phi, O(1), 2^{o(d)})$-isolation for $\phi(k) = 2^{d/k}$.

We believe that understanding these issues is the most important conceptual challenge arising from this work.

Histogram Sanitizations of Round Distributions and of Mixtures. An immediate focus of future work will be to investigate histogram sanitizations in the context of the "round" (spherical, ball, and Gaussian) distributions. (Chawla *et al.* [6] prove privacy of a first-level histogram for balls and spheres, in which the distribution is partitioned into $exp(d)$ regions, but as of this writing the results only extend to a constant number of recursive steps). Together with a cross-training result for round distributions, such a result would nicely complement our learning algorithm for mixtures of Gaussians.

The results extend immediately to the case in which the underlying distribution is a mixture of sufficiently separated "nice" distributions such as hypercubes, balls, and spheres.

Utility. Another pressing direction is to further explore utility, in particular, a method for assessing the validity of results obtained by applying a given statistical test to sanitized data. The statistics literature on imputed data (e.g., [27] should be helpful in this endeavor.

Changes over Time. An important aspect of any sanitization technique is to consider its application in an online setting, where the database changes over time. We feel that sanitizations of points should not be recomputed independently as the database changes, because an adversary collecting information over time may be able to gather enough to filter out the noise. However, in situations such as when one of the t-nearest neighbors of a point dies, one may be forced to recompute the sanitization. We believe that in such a case the new sanitization should be conditioned on the previous one appropriately, so as to prevent leakage of extra information. A related open area is to extend the definitions and techniques to multiple databases.

Real-Life Data. Then, there are the more obvious questions: how to cope with discrete data, or even non-numerical data. In general, to draw a connection to real life data, we will need to scale different attributes appropriately, so that the data are well-rounded. This requires some formal treatment.

Impossibility Results. M. Naor has suggested studying impossibility results, for example, searching for utilities that cannot be obtained while maintaining privacy. Initial investigations, already mentioned in the paper, have been fruitful. This is a subject of work in progress [14].

Acknowledgements.

We have benefited enormously from numerous conversations with many people, and are particularly grateful to Dimitris Achlioptas, Gagan Aggarwal, Jin-Yi Cai, Petros Drineas, John Dunagan, Amos Fiat, Michael Freedman, Russel Impagliazzo, Michael Isard, Anna Karlin, Moni Naor, Helen Nissenbaum, Kobbi Nissim, Anna Redz, Werner Steutzle, Madhu Sudan, and Luca Trevisan.

Not knowing of his terminal illness, but feeling his interest in research diminish, Larry Stockmeyer withdrew from this paper in April, 2004. We will always think of him as our co-author, and we dedicate this work to his memory.

References

1. D. Agrawal and C. Aggarwal, On the Design and Quantification of Privacy Preserving Data Mining Algorithms, *Proceedings of the 20th Symposium on Principles of Database Systems*, 2001.
2. N. R. Adam and J. C. Wortmann, Security-Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Surveys* 21(4): 515-556 (1989).
3. S. Arora and R. Kannan. Learning mixtures of arbitrary Gaussians. *ACM STOC*, 2001.
4. R. Agrawal and R. Srikant, Privacy-preserving data mining, *Proc. of the ACM SIGMOD Conference on Management of Data*, pp. 439–450, 2000.
5. Beck, L., A security machanism for statistical database, *ACM Transactions on Database Systems (TODS)*, 5(3), p.316-3338, 1980.
6. Chawla S., Dwork, C., McSherry, F., Talwar, K., On the Utility of Privacy-Preserving Histograms, *in preparation*, November, 2004.
7. Cox, L. H., New Results in Disclosure Avoidance for Tabulations, *International Statistical Institute Proceedings of the 46th Session*, Tokyo, 1987, pp. 83-84.
8. S. Dasgupta, Learning mixtures of Gaussians, *IEEE FOCS*, 1999.
9. Denning, D., Secure statistical databases with random sample queries, *ACM Transactions on Database Systems (TODS)*, 5(3), p.291-315, 1980.
10. P. Diaconis and B. Sturmfels, Algebraic Algorithms for Sampling from Conditional Distributions, *Annals of Statistics 26*(1), pp. 363–397, 1998
11. I. Dinur and K. Nissim, Revealing information while preserving privacy, *Proceedings of the Symposium on Principles of Database Systems*, pp. 202-210, 2003.
12. A. Dobra and S.E. Fienberg, and M. Trottini, Assessing the risk of disclosure of confidential categorical data, *Bayesian Statistics 7*, pp. 125–14, Oxford University Press, 2000.
13. C. Dwork, A Cryptography-Flavored Approach to Privacy in Public Databases, lecture at Aladdin Workshop on Privacy in DATA, March, 2003; http://www.aladdin.cs.cmu.edu/workshops/privacy/slides/pdf/dwork.pdf)

14. C. Dwork, M. Naor, et al., Impossibility Results for Privacy-Preserving Data Sanitization, *in preparation*, 2004.
15. C. Dwork and K. Nissim, Privacy-Preserving Datamining on Vertically Partitioned Databases, *Proc. CRYPTO 2004*.
16. A. V. Evfimievski, J. Gehrke and R. Srikant, Limiting privacy breaches in privacy preserving data mining, *Proceedings of the Symposium on Principles of Database Systems*, pp. 211-222, 2003.
17. Füredi, Zoltán and Komlós, János, The eigenvalues of random symmetric matrices, *Combinatorica*, 1:3, 1981, pages 233–241.
18. W. Gasarch, A Survey on Private Information Retrieval. *BEATCS Computational Complexity Column*, 82, pp. 72-107, Feb 2004.
19. R. Gavison, Privacy and the Limits of the Law, in Deborah G. Johnson and Helen Nissenbaum, editors, *Computers, Ethics, and Social Values*, pp. 332–351. Prentice Hall, 1995.
20. O. Goldreich, *The Foundations of Cryptography - Volume 2*. Cambridge University Press, 2004.
21. S. Goldwasser and S. Micali, Probabilistic Encryption, *JCSS 28*(2), pp. 270–299, 1984.
22. D. Gusfield, A Graph Theoretic Approach to Statistical Data Security, *SIAM Journal on Computing 17*(3), pp. 552–571, 1988
23. P. Indyk and R. Motwani. Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality. *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, 1998.
24. H. Kargupta, S. Datta, Q. Wang, K. Sivakumar. On the Privacy Preserving Properties of Random Data Perturbation Techniques. *Proceedings of the Third ICDM IEEE International Conference on Data Mining*, 2003.
25. J. M. Kleinberg, C. H. Papadimitriou, and P. Raghavan, Auditing Boolean Attributes, *J. Comput. Syst. Sci. 66*(1), pp. 244–253, 2003
26. F. McSherry, Spectral Partitioning of Random Graphs, *Proc. 42nd FOCS*, pp. 529 – 537, 2001
27. T.E. Raghunathank, J.P. Reiter, and D.B. Rubin, Multiple Imputation for Statistical Disclosure Limitation, *J. Official Statistics 19*(1), 2003, pp. 1–16.
28. G. Roque. Application and Analysis of the Mixture-of-Normals Approach to Masking Census Public-use Microdata. Manuscript, 2003.
29. D. B. Rubin, Discussion: Statistical Disclosure Limitation, *Journal of Official Statistics 9*(2), 1993, pp. 461–468.
30. Sibson, R, SLINK: an optimally efficient algorithm for the single-link cluster method, In *the Computer Journal* Vol. 16, No. 1, 1973, pages 30–34.
31. L. Sweeney, k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10 (5), 2002; 557-570.
32. L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10* (5), 2002; 571-588.
33. S. Vempala and G. Wang, A spectral algorithm for learning mixtures of distributions, *IEEE FOCS*, 2002.
34. W. E. Winkler. Masking and Re-identification Methods for Public-Use Microdata: Overview and Research Problems. In *Proc. Privacy in Statistical Databases* 2004, Springer LNCS 3050.
35. W. E. Winkler. Re-identification Methods for Masked Microdata. In *Proc. Privacy in Statistical Databases* 2004, Springer LNCS 3050.