Journal of Privacy and Confidentiality
Vol. 8 (1) 2018
Issue in honor of S. Fienberg
Submitted Nov 26, 2018
Published December 2018

THE FIENBERG PROBLEM: HOW TO ALLOW HUMAN INTERACTIVE DATA ANALYSIS IN THE AGE OF DIFFERENTIAL PRIVACY

CYNTHIA DWORK AND JONATHAN ULLMAN

John A. Paulson School of Engineering and Applied Sciences, Harvard University
e-mail address: dwork@seas.harvard.edu

College of Computer and Information Science, Northeastern University
e-mail address: jullman@ccs.neu.edu

Abstract. Differential Privacy is a popular technology for privacy-preserving analysis of large datasets. Differential Privacy is powerful, but it requires that the analyst interact with data only through a special interface; in particular, the analyst does not see raw data, an uncomfortable situation for anyone trained in classical statistical data analysis. In this note we discuss the (overly) simple problem of allowing a *trusted* analyst to choose an "interesting" statistic for popular release (the actual computation of the chosen statistic will be carried out in a differentially private way).

1. Introduction

Differential Privacy is a definition of privacy tailored to the statistical analysis of large datasets, together with a collection of algorithmic techniques for satisfying the definition, as well as a body of negative results and lower bounds, showing the limits of the technology as well as, in some cases, *any* even minimally protective technology.

A natural desideratum, if not definition, proposed by Tore Dalenius [2], states that anything that can be learned about a respondent by interacting with a statistical database should be learnable without access to the statistical database. This captures the intuition of the mathematically rigorous cryptographic property known as *semantic security* of a cryptosystem [7]. In this setting, there are three parties: a sender, a receiver, and an eavesdropper. The sender, wishing to communicate a plaintext message m to the receiver, creates a ciphertext c. The requirements are that the receiver can quickly recover m from c ("decrypt"), but the eavesdropper should learn nothing about m that she does not already know. Semantic security (against a passive eavesdropper) formalizes this requirement, saying that anything computable from c in polynomial time can be computed without access to c in polynomial time.[1] The difference between the eavesdropper and the receiver is that the former has a *decryption key*. With enough time, the eavesdropper could determine the key,

Key words and phrases: Differential privacy; cyber-physical systems, privacy-preserving statistical analysis.
[1]The system is equipped with a *security parameter* κ, and "polynomial time" means polynomial in κ.

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.v0.i0.999

for example, by brute force enumeration of all possible keys; hence the requirement that the eavesdropper be restricted to polynomial time computations.

Private data analysis differs from the secure communication problem in that the legitimate receiver (data analyst) and the adversary/eavesdropper (evil data analyst) may be the same party. Since we *want* to allow the data analyst to learn something about the database, we certainly don't want to require that the analyst not know more about the members of the dataset than he knew before the interaction – learning about the population represented by the people in the dataset is the whole point of statistical data analysis. Thus, instead of a "before versus after" indistinguishabilty definition as proposed by Dalenius, we shift to a definition based on the inability to determine the presence or absence in the dataset of any single individual. This "in versus out" indistinguishability captures the intuition that a statistical database should not reveal more information about an individual than could have been determined had this individual opted out of the data collection process.

In consequence, the analyst can no longer be given access to *raw data*. This is intuitively obvious if there is a possibility that the analyst is malicious. But when we trust the analyst we may be comfortable drawing a line between what the analyst learns and what the public learns from statistics and conclusions drawn from the dataset that the trusted analyst chooses to publish. The question we ask here is *how to give unfettered data access to a trusted and well intentioned analyst, and permit him to safely publish his findings?* This is the Fienberg Problem.

I first met Steve during a talk I was giving at Carnegie Mellon in 2003 describing very early thoughts on a cryptography-flavored approach to privacy in public databases. Some of these ideas arose during Adam Smith's internship with me at Microsoft. Steve was critical ("Your utility is going to be in the toilet"), but I think he was intrigued by the cryptographic approach, since after the talk he proposed that we have a workshop ("Your bring your guys and I'll bring mine"). This occurred during the summer of 2005 in the hillside town of Bertinoro, Italy. The workshop almost broke down on the second day: the statisticians thought the cryptographers, with their talk of "the adversary" and its arbitrary auxiliary information, were completely paranoid, while the cryptographers were frustrated by the absence of a formal notion of privacy and a measure of its loss in the statistical work. Fortunately, there is little to do in Bertinoro at night, other than to drink grappa in the piazza, and this eased the tension considerably. Later in the workshop Steve proposed to Alan Karr and me that we found a journal and, to paraphrase Gertrude Stein, we have and this is it.

Cynthia Dwork

DOI: 10.29012/jpc.702

We implicitly trust Steve and know that his intentions are good. Were he to promise never to publish his findings, not to let them influence any future act, including his prior beliefs when accessing future data sets, there would be no concern. But Fienberg takes publicly observable acts based on his view of the data – he is a complex "function" operating on the data set and producing statements and statistics. We therefore have two flavors of concerns. A simple version of the most easily addressed concern is that Fienberg knows the statistics he wishes to publish before he accesses the data. In this case he can publish differentially private estimates of these statistics. This will ensure that he does not accidentally publish a set of statistics that would, taken together, compromise the privacy of any individual. This also lets us reason about the privacy risks that occur when Fienberg's published statistics interact with differentially private computations of statistics published by others based on the same or intersecting datasets.

The second concern is a form of adaptivity: the decision about *which statistics* Fienberg chooses to publish can reveal sensitive information in the dataset. For example, perhaps the data of a particular outlier strikes Fienberg as odd and leads him to examine and release a specific three-way marginal, but if the sample had not contained this outlier this marginal would remain unexamined. Or perhaps seeing several members of a small minority S causes him to analyze the sample as two distinct populations, but seeing significantly fewer members of S would not have such a result.

The obvious antidote is to make the entire investigation of the dataset differentially private, which also protects against the threats to statistical validity arising in exploratory data analysis [3, 4, 1], but our goal in this work is to permit Fienberg access to the raw data, while ensuring that the choice of statistics is not disclosive. If we can make the process of choosing the statistics differentially private (while allowing Fienberg access to the raw data), then we can release these privately chosen statistics using privacy-preserving algorithms. We therefore focus on the *choice* of which statistic to release.

Fienberg is the best-case scenario, and the formulation of the problem was indeed inspired by the (lexicographically) first author's conversations with Steve in the early days of differential privacy. Importantly, despite widespread adoption of differential privacy in certain settings, for the foreseeable future the vast majority of social science research even on industrial scale datasets will likely be carried out under nondisclosure agreements, with selected researchers granted access to raw data. This reflects both the relative youth of the field of differentially private data analysis – in a nutshell, there are things we do not yet know how to do – and lack of training among the social scientists and other data analysts in the use of existing differentially private methods.

A Precise Formulation of the Problem. A data analyst will be given access to a dataset x of individuals drawn from an underlying universe $\mathcal{U}$. The analyst will be given a protocol Π to follow regarding releasing information learned from x, and is trusted to scrupulously follow this protocol. Formally, we think of the protocol as having two inputs: the data analyst $\mathcal{F} \in \mathcal{A}$, which is an arbitrary program that can interact with data, and the dataset $x \in \mathcal{U}^*$ itself. The protocol does not need to "know" anything about the analyst program, which may be arbitrarily complex; that is, the protocol has only "black box" access to the analyst.

The analyst may approach the data set with extensive background knowledge, which we model as the analyst's initial *state*. The protocol may partition the dataset into mutually disjoint smaller pieces, and require the analyst to engage with each piece afresh. This corresponds to reverting to the initial state between interactions, as if the analyst had cloned itself at the start of the protocol and the clones do not communicate with each other. While this assumption is not realistic—an analyst first exploring one subset of the data cannot truly be expected to forget everything he or she has seen before exploring a different subset of the data—it may be possible to replace clones with a small team of analysts, who are trusted not to confer during the execution of the protocol.

We formalize our privacy requirement via differential privacy (Definition 2.1). A brief review of the terminology and key results is provided in the next section, but the intuition is that the protocol will be randomized, so a dataset gives rise to a distribution over possible outputs, and the distributions on similar datasets will be similar. This similarity "hides" the participation of any individual or small group of individuals.

2. Technical Background on Differential Privacy

We begin by formally defining differential privacy. To aid in the definition, we say that two datasets x, x' are *adjacent* if they differ on a single element so that $|x \Delta x'| = 1$. We remark that it is sometimes convenient for adjacent sets to be of the same cardinality, in which case they agree on all elements except 1. We will use this same-cardinality notion of adjacency in Section 4.

Definition 2.1 (Differential Privacy [6])**.** A protocol $\Pi : \mathcal{A} \times \mathcal{U}^* \to \text{Range}$ is (ε, δ)-*differentially private* if for all adjacent datasets $x, x' \in \mathcal{U}^*$, all analysts $\mathcal{F} \in \mathcal{A}$, and all $S \subseteq \text{Range}$,

$$\Pr[\Pi(\mathcal{F}, x) \in S] \leq e^{\varepsilon} \Pr[\Pi(\mathcal{F}, x') \in S] + \delta.$$

We now give a brief recap of three of the fundamental mechanisms in differential privacy: the *Laplace mechanism*, the *exponential mechanism*, and *propose-test-release*.

Definition 2.2 (Sensitivity)**.** For $\Delta \geq 0$, a function $f : \mathcal{U}^* \to \mathbb{R}$ is Δ-*sensitive* if for all adjacent datasets $x, x' \in \mathcal{U}^*$, $|f(x) - f(x')| \leq \Delta$.

Definition 2.3 (Laplace Mechanism [6])**.** Given a Δ-sensitive function $f : \mathcal{U}^* \to \mathbb{R}$, a privacy parameter $\varepsilon > 0$ and a dataset $x \in \mathcal{U}^*$, the *Laplace mechanism* $\mathcal{LM}_{f,\varepsilon}(x)$ is defined to be the random variable $f(x) + \text{Lap}(\frac{\Delta}{\varepsilon})$. Here $\text{Lap}(\sigma)$ is the Laplace distribution with mean 0 and scale parameter σ.

Theorem 2.4 ([6])**.** *For every function f as above, $\varepsilon > 0$, $\mathcal{LM}_{f,\varepsilon}$ is $(\varepsilon, 0)$-differentially private. Moreover, for every $x \in \mathcal{U}^*$,*

$$\forall \beta > 0 \quad \Pr\left[|f(x) - \mathcal{LM}_{f,\varepsilon}(x)| > \frac{\sqrt{2}\ln(1/\beta)}{\varepsilon}\right] \leq \beta$$

Definition 2.5 (Exponential Mechanism [8])**.** Let $u : \mathcal{U}^* \times \text{Range} \to \mathbb{R}$ be a utility function such that for every $r \in \text{Range}$, $u(\cdot, t)$ is 1-sensitive and let $\varepsilon > 0$ be a privacy parameter and $x \in \mathcal{U}^*$ be a dataset. The *exponential mechanism* $\mathcal{EM}_{u,\varepsilon}(x)$ is the probability distribution given by

$$\Pr[\mathcal{EM}_{u,\varepsilon}(x) = r] = \frac{\exp\left(\frac{\varepsilon}{2} \cdot u(x, r)\right)}{\sum_{r' \in \text{Range}} \exp\left(\frac{\varepsilon}{2} \cdot u(x, r')\right)}.$$

Theorem 2.6 ([8])**.** *For every utility function u, $\varepsilon > 0$, $\mathcal{EM}_{u,\varepsilon}$ is $(\varepsilon, 0)$-differentially private. Moreover, for every $x \in \mathcal{U}^*$,*

$$\forall \beta > 0 \quad \Pr\left[\max_{r \in \text{Range}} u(x, r) - u(x, \mathcal{EM}_{u,\varepsilon}(x)) > \frac{2\ln(|\text{Range}|/\beta)}{\varepsilon}\right] \leq 1 - \beta$$

That is, the exponential mechanism outputs an approximate maximizer of the utility function $u(x, \cdot)$.

Note that the theorem pits an exponential in the utility against the cardinality of the range, so we can expect very little if the range is larger than exponential in the maximum possible utility. Equivalently, to use the exponential mechanism the utility must be at least logarithmic in the size of the range.

The Propose-Test-Release formalism, defined next, is inspired by the situation in which the worst-case sensitivity of the function f to be privately released is much larger than the *local* sensitivity on a "typical" dataset. For example, consider datasets in $\{0, 1\}^n$ drawn from

an underlying distribution $\mathcal{D}$ where $\Pr_{z\sim\mathcal{D}}[z=0]=3/4$. If, for example, the function f is the median, then we expect that with overwhelming probability $f(x)=f(x')$ for $x\sim\mathcal{D}^n$. The intuition behind Propose-Test-Release is that we can test in a differentially private way whether the dataset looks "typical"; if so, we can add little or no noise. If the dataset looks atypical the Propose-Test-Release algorithm can report this fact, and the analyst can choose a different function to compute or different algorithm for privately releasing the median. Here we give a specific instantiation of the framework.

Definition 2.7 (Propose-Test-Release [5])**.** Given a function $f:\mathcal{U}^*\to\text{Range}$, and privacy parameters $\varepsilon,\delta>0$, and dataset $x\in\mathcal{U}^*$, the *propose-test-release mechanism* $\mathcal{PTR}_{f,\varepsilon,\delta}(x)$ is defined as follows:

(1) Let d be the minimum distance between x and the nearest dataset x' such that $f(x)\neq f(x')$
(2) Let $\hat{d}=d+\text{Lap}(\frac{1}{\varepsilon})$
(3) If $\hat{d}>\frac{\ln(1/\delta)}{\varepsilon}$, output $f(x)$, otherwise output $\perp$ (which represents "no output").

Theorem 2.8 ([5])**.** *For every function f, $\varepsilon,\delta>0$, $\mathcal{PTR}_{f,\varepsilon,\delta}$ is (ε,δ)-differentially private. Moreover, if d (as defined above) is larger than $\frac{\ln(1/\delta\beta)}{\varepsilon}$, then*

$$\Pr\left[\mathcal{PTR}_{f,\varepsilon,\delta}(x)=f(x)\right]\geq 1-\beta.$$

2.1. **Sample and Aggregate.** Unfortunatley, we do not know how Fienberg works, nor do we have a bound on his sensitivity. We want him to be able to interact with the dataset in as natural a fashion as possible, with no concern for privacy. Ultimately, however, we assume he chooses to release a particular statistic from a fixed set $\mathcal{T}$ of statistics. The set $\mathcal{T}$ contains programs for computing the statistics of interest, and may be completely arbitrary. For example, it may contain a collection of regression programs, programs for computing various contingency tables, and programs for model selection. $\mathcal{F}$ is free to explore to his heart's content before choosing an element of $\mathcal{T}$ for eventual release. Responsibility for ensuring statistical validity of his selection, protecting against overfitting, false discovery control, *etc.*, rest entirely with $\mathcal{F}$.

Formally, the private analysis will operate in two stages, the first of which is the heart of the Fienberg problem. In the first stage, $\mathcal{F}$ chooses a program in $\mathcal{T}$. This is the heart of the Fienberg problem. In the second stage we will run a differentially private algorithm on the dataset x to obtain an estimate of the selected statistic. This stage is well studied and not the focus of the Fienberg problem—we focus only on the selection of the program.

Sample and Aggregate is an elegant tool permitting us to obtain differentially private approximations to functions f that are too hard to analyze [9]. In *Sample and Aggregate*, the n data items are partitioned randomly into a number k of slices. We discuss the choice of k later. The function f is run independently on each of the k slices, and the results are aggregated. The aggregation must be done in a differentially private fashion. In a little more detail, for a function $f:\mathcal{D}\to\text{Range}$, the aggregation algorithm has the form $\text{Agg}:\text{Range}^k\to\text{Range}$ and is (ε,δ)-differentially private for some $\varepsilon,\delta>0$.[2]

[2]Observe that the "dataset" given to Agg is not the original dataset $x\in\mathcal{U}^*$, but a "dataset" $y\in\mathcal{T}^k$ consisting of the output of f on each of the k slices. When we say that Agg is differentially private we mean that it respects the privacy of y (which will, in turn, imply the privacy of *Sample and Aggregate* with respect to the original dataset x).

The intuition behind *Sample and Aggregate* is deliciously simple. Privacy is particularly easy: any given datum x_i in the original dataset appears in exactly one slice, and therefore affects exactly one input to the aggregator. Since the aggregator is differentially private changing any one input to the aggregator can have little impact on the probabiltiy distribution on the outputs. To see why we have hope for utility, if a signal in the data is strong, then the signal will persist in (most of) the slices, so many of the inputs to the aggregation function will be identical or at least similar. An appropriately chosen aggregator should be able to detect this and produce a correspondingly similar output.

Our approach will be simple: view Fienberg as an arbitrary algorithm and apply *Sample and Aggregate*. In this approach, analyzed in Section 3, Fienberg must be energetic—he must be willing to explore all k slices. In addition, the proof of privacy of *Sample and Aggregate* relies on the fact that the computations on different slices are independent (because they are working on mutually disjoint subsets of the data). Thus, Fienberg must also be *forgetful*, as he cannot permit his investigation of slice i to affect his computation in slice j. Forgetfulness is implausible in practice, no matter how well intentioned Fienberg may be, but we postpone consideration of this point until Section 4. Fortunately, the privacy argument does not require that the same algorithm be applied to each slice, so we may have Fienberg investigate the first slice and various former students or others who think similarly investigate the remaining slices. For purposes of utility, we would only require that similarly trained analysts looking at slices of the same dataset are likely to choose the same statistic for eventually release, so that many copies of the same statistic are fed into the aggregation step.

3. Energetic Fienberg

In this section we are agnostic as to whether *Sample and Aggregate* is implemented with a single energetic and forgetful Fienberg, clones of Fienberg, or Fienberg and $k-1$ Friends. We require only that the parties not share information with each other during the execution of *Sample and Aggregate*, so that their behavior be independent—specifically, the behavior of clone i on slice i is independent of all the other slices. It is fine for the parties to agree on a shared strategy prior to seeing the data. Since we do not need to understand the algorithm $\mathcal{F}$, we need only discuss the aggregation method.

3.1. Few Possible Outcomes.

If the set $\mathcal{T} = \text{Range}(\mathcal{F})$ of outcomes is not too large, then we may aggregate using the exponential mechanism (Definition 2.5). Recall that the aggregation function Agg maps $\mathcal{T}^k$ to $\mathcal{T}$. For $y \in \mathcal{T}^k$ and $t \in \mathcal{T}$, we define the utility of t for y, denoted $\text{count}(t, y) \in \{0, 1, \ldots, k\}$, as the number of occurrences of t in y. We can see that count is 1-sensitive. It is now immediate from Theorem 2.6 that

$$\forall \beta > 0 \quad \Pr\left[\max_{r \in \text{Range}} \text{count}(y, r) - \text{count}(y, \mathcal{EM}_{u,\varepsilon}(y)) > \frac{2\ln(\mathcal{T}/\beta)}{\varepsilon}\right] \leq \beta.$$

To interpret this guarantee, suppose that all k copies of Fienberg unanimously agree on an outcome t^* so that $\text{count}(y, t^*) = k$ and $\text{count}(y, t) = 0$ for all $t \neq t^*$. In this case we can guarantee that the aggregation step outputs t^* with probability at least $1 - \beta$ so long as there are $k > \frac{2\ln(|\mathcal{T}|/\beta)}{\varepsilon}$ Fienbergs. For example, if we want to guarantee $(1, 0)$-differential privacy, and have a 95% chance of success, then with 100 outcomes we require a modest party of $k = 16$ unanimous Fienbergs. This example is not overly specific to unanimous

Fienbergs—if the two Fienbergs are evenly split between two outcomes t_1, t_2, then the exponential mechanism will select *one* of these as long as there are $k = 32$ Fienbergs.

However, when the space of outcomes is enormous, requiring $k > 2 \ln |\mathcal{T}|$ Fienbergs becomes prohibitive. With a not unrealistic choice of 2^{30} outcomes we require $k \geq 48$ Fienbergs even in the best possible scenario where all Fienbergs agree and $k \geq 96$ if the Fienbergs are evenly split! In addition to requiring many energetic, forgetful Fienbergs, each of the k slices is a separate chunk of data, meaning that this approach requires the size of the dataset to be at least kn where n is the number of samples required for Fienberg to make the right selection most of the time.

3.2. **Many Possible Outcomes.** When the space of outcomes $\mathcal{T}$ is too large to apply the exponential mechanism, we can replace the exponential mechanism with the Propose-Test-Release framework (Definition 2.7) to avoid having the number of Fienbergs grow with the number of outcomes. In exchange, we now require that there is a clear favorite choice among the k Fienbergs (*i.e.* a significant plurality), and this approach will require $k \in \omega(\log n)$ in order to ensure (ε, δ)-differential privacy for a cryptographically small $\delta > 0$, *i.e.* smaller than the inverse of any polynomial in the size n of the dataset.

The test will examine the difference in the popularity of the top two options. That is, we can think of each clone operating on a slice as casting a vote for some $t \in \mathcal{T}$ and we are looking at the difference between the counts of the most popular and next most popular options in this tally, breaking ties with a fair coin. Henceforth, let $t_{\max}$ be the most popular and t_{next} be the next most popular. We allow the algorithm to break ties arbitrarily, but the algorithm is most interesting when there is a clear favorite $t_{\max}$. In this case, Propose-Test-Release specializes to the following differentially private aggregator:

> If $\text{count}(y, t_{\max}) - \text{count}(y, t_{\text{next}}) + \text{Lap}(\frac{2}{\varepsilon}) \geq \tau = \frac{2 \ln(1/\delta)}{\varepsilon}$ then output $t_{\max}$.[3]
> Otherwise, output $\perp$.

Privacy and utility follow from Theorem 2.8. To build intuition, we briefly discuss both.

Privacy. At first, privacy seems obvious: the presence or absence of any individual from the dataset can affect at most one count, by at most one. We now explain why this algorithm cannot achieve $(\varepsilon, 0)$-differential privacy for any ε. To make the math simple we will use an extreme scenario.

Suppose Fienberg would never *think* of producing $\hat{t} \in \mathcal{T}$ unless he sees the data of a particular outlier $\mathbf{o}$. Suppose further that this is if and only if, meaning that if $\mathbf{o}$ is in the dataset then Fienberg will necessarily produce $\hat{t}$. This is immediately a red flag for pure differential privacy (the $\delta = 0$ case), as it says there is a potential outcome $\hat{t}$ that could occur on x and that cannot occur on x'. It remains only to complete the scenario to show that using *Sample and Aggregate* $\hat{t}$ can indeed occur. For this, we assume that the datasets are such that there is no particularly interesting signal, so when run on x' the counts for the different choices are all in $\{0, 1\}$. In this case $t_{\max}$ and t_{next} are tied for most popular, and $t_{\max} \neq \hat{t}$, so on this dataset there is zero probability of selecting $\hat{t}$. On x, however, the probability of producing $\hat{t}$ is not zero, as one of the inputs to the aggregation function is $\hat{t}$

[3]Note that the noise added is $\text{Lap}(\frac{2}{\varepsilon})$ because the function $u(y, t_{\max}) - u(y, t_{\text{next}})$ is *twice* the distance between y and the nearest y' such that $t_{\max}$ changes, whereas in Definition 2.7 we used the distance itself.

(because **o** appears in one of the slices). This example violates $(\varepsilon, 0)$-differential privacy for any finite value of ε.

However, in the case where $\text{count}(t_{\max}) = \text{count}(t_{\text{next}})$, the aggregator produces an output $t \in \mathcal{T}$ only if the draw from $\text{Lap}(\frac{2}{\varepsilon}) \geq \tau$, which has probability at most $\exp(-\frac{\varepsilon\tau}{2})$. For fixed $\varepsilon > 0$, this probability is negligible in n as long as $\tau \in \omega(\log n)$; this is how we get (ε, δ)-differential privacy.

Utility and Sample Complexity. Using Thereom 2.8 we immediately obtain the following guarantee: for every $\beta > 0$, if $t_{\max} - t_{\text{next}} \geq \frac{2\ln(1/\delta\beta)}{\varepsilon}$, then the probability of returning $t_{\max}$ is at least $1 - \beta$. To get a feel for what this condition means, suppose we want to have a 95% chance of making the correct decision in the end, and suppose that each slice is large enough that Fienberg would select $t_{\max}$ on each slice at least 3/4 of the time. Then *on average* we can say that Fienberg indeed selects $t_{\max}$ on at least $3k/4$ slices, meaning $t_{\max} - t_{\text{next}} \geq k/2$. Thus to make the correct selection and achieve $(1, \delta)$-differential privacy, we would suffice to have $k \approx 4\ln(20/\delta)$ slices. Note that this holds for an *arbitrarily large (possibly infinite)* set of choices $\mathcal{T}$ so long as there is a single favorite choice.

Observe that each slice only needs to be large enough for Fienberg to make the right selection *most* of the time. Thus, each slice need not be quite large enough for Fienberg to make the right decision with the *high confidence* we would want for the entire analysis. Thus each slice might be smaller than what we would want in isolation, and the increase in sample complexity might be smaller than a factor of k.

4. Lazy Fienberg?

In addition to being sample-hungry, the two previous methods make high demands on Fienberg's energy and ability to forget (or work with colleagues). Is there some way to make things easier for Fienberg? We offer an approach to reduce the demands on Fienberg's energy level based on one of the bedrock principles of computer science: *verifying can be easier than computing.*

As we noted, in this section it is more convenient to use the "same cardinality" notion of adjacency. Thus, our adjacent sets x, x' will be of the same cardinality, say, n, and will agree on $n - 1$ elements.

In our new approach, we only apply the Fienberg program $\mathcal{F}$ to one slice of data. On each of k remaining slices of the data, we apply a simpler *verifier* $\mathcal{V} : \mathcal{U}^* \times \mathcal{T} \to \{0, 1\}$. Each verifier takes in a slice of data and an outcome in $\mathcal{T}$, and either accepts (outputs 1) or rejects that outcome (outputs 0). Each slice of data x_j may use a different verifier $\mathcal{V}_j$, which may represent k students or assistants of Fienberg's. Given these checkers, we can use the following variant of *Sample and Aggregate* that we call the *Lazy-Fienberg Protocol*.

(1) With probability $\frac{1}{(k+1)\varepsilon}$, output $\perp$ and terminate.[4]
(2) Randomly split the data into $k + 1$ slices $x_0, x_1, \ldots, x_k \in \mathcal{U}^*$.
(3) Run Fienberg on the first slice to obtain $t \leftarrow \mathcal{F}(x_0)$, and let count $\leftarrow \sum_{j=1}^{k} \mathcal{V}_j(x_j, t)$ be the number of verifiers who accept Fienberg's choice.

[4]If the algorithm terminates and outputs $\perp$, then no statistic is selected. Because of the first step, $\perp$ is returned with probability $p \geq \frac{1}{(k+1)\varepsilon}$. We could then run the algorithm t times on a separate datasets, which reduces the probability of failure exponentially to p^t while requiring a factor of $(k + 1)t$ analysts in total.

(4) If $\text{count} + \text{Lap}(\frac{1}{\varepsilon}) > \tau = \frac{2k}{3} + \frac{\ln(1/\delta)}{\varepsilon}$ then output t. Otherwise, output $\perp$.

Intuitively, this algorithm checks that a significant fraction of the verifiers agree on Fienberg's choice of outcome. This approach as stated is *not* differentially private for the simple reasons that the verifiers could accept *any* outcome Fienberg comes up with, in which case the algorithm outputs $\mathcal{F}(x_0)$ with probability ≈ 1! If, as above, Fienberg has a special output $\hat{t}$ that he thinks of if and only if the dataset contains the outlier $\mathbf{o}$, this is a disaster.

To make this approach private, we crucially assume that the verifiers are *picky*: for every dataset x, $\mathcal{V}_j(x,t)$ accepts only if $\mathcal{F}(x) = t$. One way to make the checker picky is to have it run Fienberg himself, however in this case the verifier is no less energetic than Fienberg himself. However, we conjecture that there are picky verifiers that can be much *lazier* than Fienberg.

We now discuss why the Lazy-Fienberg Protocol is private when we have a sufficiently large number of picky verifiers. First, assuming that the number k of verifiers is sufficiently large, the probability that $\text{Lap}(\frac{1}{\varepsilon}) > \frac{k}{6}$ is negligibly small. Thus, except with very small probability, the algorithm will output $\perp$ unless $> \frac{k}{2}$ of the verifiers accept, *i.e.*, unless there is a strict majority of the verifiers accept. This means that, for any partitioning, there is at most one possible strict majority choice t^*. Thus, once we fix the slices $x_1, \ldots, x_k$, there are only two possibilities: either Fienberg, operating on x_0 selects t^* or he selects something other than t^*, causing the algorithm to output $\perp$ with high probability.

Consider adjacent x and x' and an arbitrary sequence of coins ω used in the random partitioning. Let $i \in \{0, 1, \ldots, k\}$ denote the unique slice for which $x_i \neq x'_i$. Note that $\Pr_\omega[i = 0] = 1/(k+1)$.

Assuming $i \neq 0$ (the more likely case), the change occurs in one of the verifiers' slices, potentially causing this verifier to change its vote. In this case the addition of noise $\text{Lap}(\frac{1}{\varepsilon})$ "hides" this change.

If instead $i = 0$, which occurs with probability only $\frac{1}{k+1}$, the change occurs in Fienberg's slice, in which case the probability Fienberg outputs t^* can change arbitrarily, and by pickiness there would be no strict majority supporting his changed output, resulting in $\perp$ (with high probability). However, the algorithm already outputs $\perp$ with probability at least $\frac{1}{\varepsilon(k+1)}$, we can ensure a multiplicative change in probability of outputting $\perp$ of at most $1 + \varepsilon \approx e^\varepsilon$. Formalizing this analysis yields the following result.

Theorem 4.1. *There is an absolute constant $C > 0$ such that if there are $k \geq \frac{C \log(1/\delta)}{\varepsilon}$ picky verifiers then the Lazy-Fienberg Protocol satisfies $(C\varepsilon, C\delta)$-differential privacy.*

We now discuss why this approach provides utility. If the verifiers do indeed accept the choice that Fienberg would have made on their slice, and given a random slice, at least a $p \approx 1$ fraction of the Fienbergs agree on some choice t^*, then the algorithm will output t^* so long as four events occur. These events are: (1) the algorithm does not output $\perp$ in the first step, (2) $\mathcal{F}(x_0) = t^*$, which happens with probability p by assumption, (3) count $> 5k/6$, which occurs with probability at least $1 - \exp(-\Omega(k))$ by standard concentration of measure arguments, and (4) the noise value from $\text{Lap}(\frac{1}{\varepsilon})$ is at most $\frac{k}{6}$, which also occurs with probability at least $1 - \exp(\frac{\varepsilon k}{6})$. Thus the Lazy-Fienberg Protocol outputs the correct choice with probability at least $p - \frac{1}{k\varepsilon} - \exp(-\Omega(\varepsilon k))$, which is close to the probability that Fienberg himself selects t^*.

5. Conclusions

Although we have described the "Fienberg" problem in a playful fashion, the importance of the problem cannot be overstated. Absent a solution to the Fienberg problem, we see no way of arguing formally about the privacy risks that are incurred by multiple studies of the same or overlapping datasets. Efforts to open internet-scale corporate datasets to social science research (e.g. Social Science One [10]) may rely on user agreements and trust, but without rigorous guarantees we cannot understand how the publications may interact. If, however, our trusted analysts obtain their published results using a protocol that solves the Fienberg problem, these publications will be differentially private, and we can use standard composition theorems for differential privacy to understand and control the cumulative privacy loss.

A number of specific technical questions remain after our initial steps. Foremost among these would be to find an alternative that does not require the increase in sample complexity and human effort implicit in using the *Sample and Aggregate* paradigm. Since *Sample and Aggregate* is the only known approach to handling arbitrary functions, such an approach would be of interest even beyond the Fienberg problem.

Finally, we remark also that a solution to the Fienberg problem addresses two very different kinds of difficulties: first, there may not (yet) exist sufficiently good differentially private algorithms for the analysis tasks at hand; second, even if the algorithms exist, not all data analysts are skilled in the techniques of privacy-preservation.

There are some interesting smaller questions raised by the Lazy Fienberg problem. How realistic is the assumption of "pickiness," and is it necessary in any sense? Under what conditions can verification $\mathcal{V}(x,t)$ be simpler than running $\mathcal{F}(x)$?

References

[1] R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the 48th Annual ACM Symposium on the Theory of Computing*, STOC '16, pages 1046–1059, Cambridge, MA, 2016. ACM.

[2] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[3] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Preserving statistical validity in adaptive data analysis. *arXiv preprint arXiv:1411.2664*, 2014.

[4] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th Annual ACM Symposium on the Theory of Computing*, STOC '15, pages 1046–1059. ACM, 2015.

[5] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, STOC '09, pages 371–380. ACM, 2009.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[7] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[8] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, FOCS '07, pages 94–103, 2007.

[9] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 30th annual ACM Symposium on Theory of Computing, STOC*, pages 75–84, 2007.

[10] Social Science One. Socialscienceone. `https://socialscience.one/`, 2018.