Element Level Differential Privacy: The Right Granularity of Privacy

Hilal Asi,[1] **John Duchi,** [1,2] **Omid Javidbakht** [2]

[1] Stanford University
[2] Apple
asi@stanford.edu, jduchi@stanford.edu, omid_j@apple.com

Abstract

Differential Privacy (DP) provides strong guarantees on the risk of compromising a users data in statistical learning applications, though these strong protections make learning challenging and may be too stringent for some use cases. To address this, we propose element level differential privacy, which extends differential privacy to provide protection against leaking information about any particular "element" a user has, allowing better utility and more robust results than classical DP. By carefully choosing these "elements," it is possible to provide privacy protections at a desired granularity. We provide definitions, associated privacy guarantees, and analysis to identify the tradeoffs with the new definition; we also develop several private estimation and learning methodologies, providing careful examples for item frequency and M-estimation (empirical risk minimization) with concomitant privacy and utility analysis. We complement our theoretical and methodological advances with several real-world applications, estimating histograms and fitting several large-scale prediction models, including deep networks.

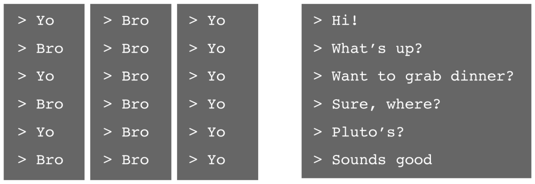
Introduction

The substantial growth in data collection across many domains has led to commensurate attention to and work on privacy risks in both academic [Dwork et al. 2006b, Dwork and Roth 2014] and industrial settings [Erlingsson, Pihur, and Korolova 2014, Apple Differential Privacy Team 2017, Bhowmick et al. 2018]. *Differential privacy* [Dwork et al. 2006b] and its variants [Dwork et al. 2006a, Bun and Steinke 2016, Mironov 2017]—where a randomized algorithm returns similar outputs for similar input samples—is now the standard privacy methodology, as it gives provable protection against strong adversarial attacks on privacy. Indeed, given the output of a differentially private analysis on a sample $S = \{X_1, \ldots, X_n\}$, it is challenging to identify whether a particular individual x belongs to S even for an attacker knowing the entire sample except for a single observation. These strong guarantees motivate work on private data analyses, including in statistical estimation [Smith 2011, Duchi, Jordan, and Wainwright 2018], machine learning [Chaudhuri, Monteleoni, and Sarwate 2011], game theory [McSherry and Talwar 2007], and networks and graphs [Kasiviswanathan et al. 2013, Kearns et al. 2016].

Yet developing private algorithms that achieve reasonable utility is challenging, as the strong protections differential privacy provides necessarily degrade statistical utility. On the theoretical side, the relative sample size necessary for private algorithms to achieve similar utility to that of non-private algorithms grows with problem dimension and inversely with the privacy parameter ε [Barber and Duchi 2014, Steinke and Ullman 2017, Duchi, Jordan, and Wainwright 2018, Duchi and Rogers 2019]. On a practical level, this challenge may lead privacy applications to instantiate a large privacy parameter ε to obtain acceptable statistical performance—for example, Abadi et al. [2016] remarkably are able to fit neural networks with differential privacy at all, though they require a value of $\varepsilon = 8$ even for a weaker form of "event level" privacy to achieve performance approaching non-private algorithms—but privacy guarantees for large values are unclear [Dwork and Roth 2014].

We argue that standard differential privacy's strong protections are not always necessary to provide sufficient protection for a system's users. For example, an individual phone user sends multiple text messages, or takes several cell-phone photos, each a single datum. In such cases, it may be satisfying from a privacy perspective not to protect whether a user participates in a dataset—versions of differential privacy protect against discovering this participation, though whether one has a phone is likely not very sensitive—but to protect so that no one knows any particular *thing* a user has done, e.g., whether the user has *ever* typed a given word or taken a photo of a mountain. Concretely, consider estimating the frequency of different word use in email messages. Differential privacy prevents an attacker from (accurately) distinguishing a user who sends hundreds of emails daily from one who has never typed a word in his or her lifetime, a protection that may be too strong. More nuanced tradeoffs can arise if we wish to prevent an attacker from knowing, for example, whether a user has ever typed a given word.

To address these challenges, we propose *element-level differential privacy*, which aims to provide protection for what we—at the risk of some hubristic excess—might term reasonable attacks. The motivation for our definition is that in many statistical estimation and learning problems, an individual may contribute many datapoints; in a problem of learning from mobile devices, a typical cell-phone contains many individual photos and hundreds of distinct text messages, for example, and it is these data that are private. The key to differential privacy and its descendant definitions is the notion of *neighboring datasets* [Dwork and Roth 2014] or samples, where privacy guarantees certify that an adversary given the output of a private mechanism M cannot reliably distinguish between its applications $\mathsf{M}(x)$ and $\mathsf{M}(x')$ on neighboring samples x and x'. In differential privacy, two samples are neighboring if they differ in at most a single observation. As Chatzikokolakis et al. [2013] note, it is thus natural to quantify a distance between users or samples x, x' to redefine neighboring, and mechanisms then provide privacy for nearby users under this distance [Chatzikokolakis et al. 2013].



```
> Yo
> Bro
> Yo
> Bro
> Yo
> Bro
```

```
> Bro
> Bro
> Bro
> Bro
> Bro
> Bro
```

```
> Yo
> Yo
> Yo
> Yo
> Yo
> Yo
```

```
> Hi!
> What's up?
> Want to grab dinner?
> Sure, where?
> Pluto's?
> Sounds good
```

Figure 1. Example histories of four different users' text messages (each column represents a user's conversation). The left three columns reflect a conversation of the first author with his friends. The rightmost is a conversation between the second and third authors. In the standard differential privacy definition, each user is distance 1 from each other user. In contrast, element-level privacy (with the histogram distance function described in the introduction) identifies the three left transcripts as neighboring—at distance 2—irrespective of the number of times each uses the word `yo` or `bro`, while the right conversation is distant.

Element-level privacy takes this idea and defines distances based on the *elements*, which we describe in the sequel, that an individual user's data x contains; here, two users are neighboring if they differ in one or fewer elements. Consider estimating frequency of word use in text (SMS) messages. Then a possible distance function between two users is the *number* of words that have different counts per user, i.e., we represent each user as a vector $x \in \mathbb{N}^d$ of per-word counts (how many times the user used each word in a dictionary of size d), and the distance between users is the Hamming distance $d(x, x') = \sum_{j=1}^{d} 1\{x_j \neq x'_j\}$ between their histograms (see Figure 1). Element-level differential privacy then makes it challenging for an attacker to discover any particular word a user utters. We present more concrete examples to compare and contrast element-level and classical differential privacy.

As we note above, there is substantial work on privacy broadly, with a line of work investigating appropriate notions of distance and what distinctions between individuals and data should be protected. We highlight a few works in this direction here. Andrés et al. [2013] develop distance-based notions of privacy to release information to geo-location services, where privacy protections may degrade with distance to a user (e.g., it is acceptable to release that a user is in Paris, but perhaps not at 28 Rue Vieille du Temple). In the context of large-scale web or mobile applications, there are differences between *event-level* privacy [Dwork et al. 2010, Erlingsson, Pihur, and Korolova 2014, Abadi et al. 2016], which protects each individual action a user takes, though a user contributing multiple data items (e.g. sending multiple text messages) suffers linear degradation in privacy guarantees, and *user-level* privacy [McMahan et al. 2018], where all users are neighboring, no matter how many data contributions they make or how diverse their data. The former (event-level) provides limited privacy guarantees, while the latter (user-level) may be too strong for practical use. In this context, *element-level* privacy attempts to provide privacy at the right granularity for the application at hand: in a way we formalize shortly, one identifies the elements to be protected, then guarantees that no matter how much data corresponding to a particular element a user contributes, the output of the privacy mechanism changes little.

In the remainder of the paper, we carefully define element-level differential privacy, using standard tools to show that it inherits many of the desiderata important for satisfactory privacy definitions (composition, group privacy, privacy to post-processing, side-information resilience, and amplification by subsampling). As one of our major goals is to provide practicable procedures for estimation and learning with privacy protections, we present several methodological contributions. In particular, we demonstrate histogram estimators and tools for estimation of frequent elements, highlighting the advantages element-level privacy can provide, and we show how to apply element-level privacy to fit large scale machine learning models and compute M-estimators using stochastic-gradient-type methods. Along the way, we demonstrate a new asymptotic normality result for stochastic approximation procedures applied to fixed finite datasets, which may be of interest beyond privacy. We complement these with experimental evidence on several real-world machine-learning tasks.

Element-level privacy

As we allude in the introduction, our main goal in this paper is to provide a new definition of privacy, simultaneously developing its properties while demonstrating new procedures that obey its strictures. To that end, we begin by defining element-level privacy, contrasting it with prior notions.

Privacy definitions

The key to each of these definitions of privacy is a *distance* on the space of samples. In particular, let $d_{\mathsf{sample}} : \mathcal{X}^n \times \mathcal{X}^n \to \mathbb{R}_+$ be a distance on $\mathcal{X}^n$, and let M be a randomized mapping from $\mathcal{X}^n$ to some (measurable) space $\mathcal{Z}$. In standard differential privacy, this distance is the (order-invariant) Hamming metric: letting Π_n be the collection of all permutations of n elements, for samples $S = (x_1, \ldots, x_n), S' = (x'_1, \ldots, x'_n) \in \mathcal{X}^n$ we have

$$d_{\mathsf{sample}}(S, S') = d_{\mathsf{ham}}(S, S') := \min_{\pi \in \Pi_n} \sum_{i=1}^{n} 1\{x_i \neq x'_{\pi(i)}\}.$$

As Chatzikokolakis et al. [2013] note, focusing on the case of differential privacy, we may take any distance on the samples to provide analogues of differential privacy; such alternative distances are important, for example, for graph-based notions of differential privacy [Kasiviswanathan et al. 2013], location services [Andrés et al. 2013], or event-level streams [Dwork et al. 2010, Erlingsson, Pihur, and Korolova 2014].

We thus make the following definitions, which generalize those in prior work by treating distance between two samples as a first-class object.

Definition 0.1 (Dwork et al. [Dwork et al. 2006b,a]). *Let $\varepsilon, \delta \geq 0$. The randomized mechanism $\mathsf{M} : \mathcal{X}^n \to \mathcal{Z}$ is* (ε, δ)-differentially private *for the distance d_{sample} if for any pair of samples S, S' with $d_{\mathsf{sample}}(S, S') \leq 1$ and any measurable subset $A \subset \mathcal{Z}$,*

$$\mathbb{P}(\mathsf{M}(S) \in A) \leq e^{\varepsilon} \mathbb{P}(\mathsf{M}(S') \in A) + \delta.$$

Rather than exhaustively discussing alternative privacy definitions, we note that each variant of differential privacy (Rényi privacy [Mironov 2017] or concentrated differential privacy [Dwork and Rothblum 2016, Bun and Steinke 2016]) similarly rely on sample distances, saying that a mechanism $\mathsf{M}(\cdot)$ is private if its output distribution changes little (under an appropriate metric) when its input sample changes.

Element-level privacy definition

The standard distance in each privacy definition is the Hamming distance between samples S, S'; this is satisfying, as it limits any inferences that can be made about an individual [Dwork et al. 2006b]. In some scenarios, this definition makes learning challenging (or, depending on the task and desired privacy guarantee, essentially impossible) [Duchi, Jordan, and Wainwright 2018]. It is thus natural to consider more fine-grained distance notions to allow utility while providing sufficient privacy. For our purposes, it is useful to consider a scenario frequent in large-scale learning applications, such as federated learning (e.g. [Abadi et al. 2016]), where individual users contribute multiple data items rather than a single item. In such cases, we protect a user so that no one knows any particular *thing* the user has done. For example, a student with a phone sends many text messages, but may wish that his parents and teachers never know whether he has ever sent a curse word, irrespective of the number of times he may or may not have sent one.

To formalize this, we introduce *element-level privacy*. A sample or dataset S consists of n user's data (or data units) $S = \{x^{(u)}\}_{u=1}^{n}$, while each user u maintains local data of size $m(u)$, where the size may depend on the user $x^{(u)} = \{x_1^{(u)}, \ldots, x_{m(u)}^{(u)}\}$. For example, individual u's data may consist of the $m(u)$ photos she has taken. External to the users are K *clusters* $\{c_1, \ldots, c_K\}$ partitioning $\mathcal{X}$, where we view the cluster centroids as the *elements* to be made private, and each datapoint $x_i^{(u)}$ belongs to precisely one cluster c_k (i.e. has a nearest element); we denote this by $x_i^{(u)} \in c_k$. The distance between two users' local data $x = \{x_1, \ldots, x_n\}$ and $x' = \{x'_1, \ldots, x'_m\}$ is then the number of clusters $c_1, \ldots, c_K$ with different memberships for the two users' data, that is,

$$\begin{aligned} d_{\mathsf{user}}(x, x') &= d_{\mathsf{user}}(\{x_1, \ldots, x_n\}, \{x'_1, \ldots, x'_m\}) \\ &:= \sum_{k=1}^{K} 1\left\{\{x_i : x_i \in c_k\} \neq \{x'_i : x'_i \in c_k\}\right\}, \end{aligned} \tag{1}$$

where $\{x_i : x_i \in c_k\}$ are implicitly multi-sets. Then two users' data x, x' are *element-neighbors* if $d_{\mathsf{user}}(x, x') \leq 1$; this is equivalent to allowing users to differ arbitrarily on one element of their data. With this distance definition, we can then define the element-level sample distance by

$$d_{\mathsf{element}}(S, S') := \min_{\pi \in \Pi_n} \sum_{u=1}^{n} d_{\mathsf{user}}(x^{(u)}, x'^{(\pi(u))}). \tag{2}$$

Two samples S, S' of size n are *element-neighbors* if each of the units within the sample is identical except for (at most) one unit $x \in S, x' \in S'$, where $d_{\mathsf{user}}(x, x') \leq 1$. The definition of element level privacy is now immediate: we take the sample distance d_{sample} in any privacy definition (e.g. 0.1) to be d_{element}.

Definition 0.2. *A mechanism M satisfies* element-level *differential privacy if it satisfies Definition 0.1 with distance $d_{\mathsf{sample}} = d_{\mathsf{element}}$.*

Element-level differential privacy guarantees that the releases of a mechanism trained on users' sensitive data does not leak any particular "element" the user has, that is, whether a user has data belonging to any one of the clusters $c_1, \ldots, c_K$, no matter how many data point belong to one of the clusters. It is useful to compare this definition to two frequent definitions of privacy for large-scale learning systems. The first is *event-level privacy* [Erlingsson, Pihur, and Korolova 2014], which applies privacy commensurate with each individual *event* a user performs, for example, whenever a user visits any website. This definition may be too weak: consider a user who sends 50 text-messages consisting of the phrase "Hello!" Then event-level privacy (say with Def. 0.1) guarantees a likelihood ratio bound of $e^{50\varepsilon}$ versus an otherwise identical user who never uses the phrase "Hello!" In the case of element-level privacy, however, the distance between these users is at most 1 regardless of how many times either says "Hello!" The second common definition is *user-level privacy*, which corresponds to the standard definitions with Hamming distance; by taking a single cluster $c_1 = \mathcal{X}$ in the definitions (1)–(2) of element level distances, one recovers user-level privacy, but as we shall see, the additional flexibility of element-level privacy allows more utility.

To get a feel for Definition 0.2, it is instructive to consider two (somewhat stylized) examples.

Example 1 (Word frequency estimation)**:** Consider the problem of estimating frequent words used in text (SMS) messages. Ignoring punctuation, we treat each word as a cluster, so that for a dictionary of size d, a user u's data $x^{(u)} = \{x_1^{(u)}, \ldots, x_d^{(u)}\}$ consists of the counts $x_j^{(u)} \in \mathbb{N}$ of the times user u typed word j, a histogram of word counts. In Figure 1, for example, the leftmost column has histogram with count 3 for the word "yo," 3 for "bro," and 0 for all other words. The distance between two user data x, x' is then $d_{\mathsf{user}}(x, x') = \sum_{j=1}^d 1\{x_j \neq x'_j\}$, the number of distinct counts. In this case, two users are neighboring when their word use is identical except that one may use a word j arbitrarily more or less than the other. ◇

Example 2 (Website visit counts)**:** Consider estimating the frequency of popular websites (URLs) that users visit. In this case, a natural set of elements are domains (the first part of a website name), while specific URLs belong to a single domain. For example, https://en.wikipedia.org/wiki/Apple_Inc. and https://en.wikipedia.org/wiki/NeXT belong to the domain (cluster) `wikipedia.org`, while http://web.stanford.edu/~jduchi/ and http://web.stanford.edu/~asi/ belong to `stanford.edu`. Then a user's data consists of all URLs he or she visits, while the distance between users is the number of domains in which they visit distinct URLs. The intuition here is that any mechanism satisfying Definition 0.2 limits release of whether a user ever even visits a website in a particular domain, for example, `wikipedia.org`, `stanford.edu`, or `youtube.com`. In contrast, standard differential privacy would protect whether a user has ever used the internet. ◇

As these examples attempt to clarify, the important facet of element-level DP is that it protects a data provider from anyone ever knowing any particular thing they have done, regardless of how many times they have done it: visiting a domain, using a word, or other desired protected element.

Properties of element-level differential privacy

By replacing the standard Hamming distance in the definition of differential privacy with the element-based distance (2), any element-level differentially private mechanism inherits the typical properties private mechanisms enjoy, including privacy to post-processing, group privacy, composition, and amplification of privacy by subsampling (see the book [Dwork and Roth 2014] for a discussion of these desiderata). Almost all of these inheritances are immediate and we present them in the full version of the paper.

Element-level private methods

One of our major goals is to demonstrate the methodological possibilities of mechanisms satisfying element-level privacy, both to give some sense of the way to design mechanisms satisfying the definition and to understand the potential utility benefits—in terms of more accurate estimation—element-level privacy allows over user-level notions of privacy. To that end, we present two examples in this section of increasing sophistication: estimating multinomial frequencies, and stochastic optimization (statistical learning).

We begin by attempting to give a somewhat general picture, connecting to the classical Laplace mechanisms and sensitivity analyses of Dwork et al. [2006b]; we specialize in the coming sections. Suppose each user contributes a batch $x = (x_1, \ldots, x_m)$ of data, and we wish to compute the average $\frac{1}{n}\sum_{u=1}^n f(x^{(u)})$ of a function $f : \mathcal{X}^m \to \mathbb{R}$ on $S = \{x^{(u)}\}_{u=1}^n$. Standard mechanisms add noise that scales with the *global sensitivity* of the function f, that is, $\mathsf{gs}(f) := \sup_{x \in \mathcal{X}^m, x' \in \mathcal{X}^m} |f(x) - f(x')|$, and the Gaussian mechanism for (ε, δ)-differentially private release is

$$\mathsf{M}(S) := \frac{1}{n}\sum_{u=1}^n f(x^{(u)}) + \mathsf{gs}(f) \cdot \mathsf{N}\left(0, \frac{2\log\frac{1}{\delta}}{\varepsilon^2}\right).$$

In contrast, given a partition $\{c_1, \ldots, c_K\}$ of $\mathcal{X}$ and corresponding user distance d_{user} (recall Eq. (1)), the analogous recipe here is to add noise scaling with the *element sensitivity* of f,

$$\mathsf{es}(f) := \sup_{x, x' \in \mathcal{X}^m} \{|f(x) - f(x')| \text{ s.t. } d_{\mathsf{user}}(x, x') \leq 1\}, \tag{3}$$

which satisfies $\mathsf{es}(f) \leq \mathsf{gs}(f)$. Then the standard Gaussian mechanism becomes

$$\mathsf{M}(S) := \frac{1}{n}\sum_{u=1}^n f(x^{(u)}) + \mathsf{es}(f) \cdot \mathsf{N}\left(0, \frac{2\log(1/\delta)}{\varepsilon^2}\right) \tag{4}$$

and guarantee (ε, δ)-element-level differential privacy. We see utility gains whenever $\mathsf{es}(f) \ll \mathsf{gs}(f)$, which we expect when the number K of elements is large, providing finer granularity privacy.

Histogram estimation

We now turn to the problem of estimating item frequencies—histogram estimation—one of the original motivations for differential privacy [Dwork et al. 2006b, Ex. 3]. We have $X^{(u)} \overset{\text{iid}}{\sim} \mathsf{Multinomial}(m, p)$ for some $m \in \mathbb{N}$ and $p \in \mathbb{R}^d_+$ with $p^T \mathbf{1} = 1$. We elaborate this setting somewhat to allow more substantial elements, as in Example 2, by assuming there are K *clusters* $\{c_1, \ldots, c_K\}$ partitioning $[d]$. For shorthand, for $v \in \mathbb{R}^d$ we let $v_{c_k} = [v_j]_{j \in c_k} \in \mathbb{R}^{|c_k|}$, and we denote the probability of an item in c_k by $P(c_k) = \mathbf{1}^T p_{c_k} = \sum_{j \in c_k} p_j$.

We consider a normal noise addition mechanism (4), but our first step is to design a function insensitive to changes within the partition $\{c_1, \ldots, c_K\}$ of $[d]$, reducing the element sensitivity. To that end, we consider a mechanism that first projects each cluster c_k of counts into an ℓ_2-ball, then adds Gaussian noise. For $v \in \mathbb{R}^d$, we define the projection

$$\pi_{\rho,\{c_k\}}(v) := \operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ \|x - v\|_2^2 : \|x_{c_k}\|_2 \leq \rho \right\}$$

The mechanism is then

$$\mathsf{M}(S, \rho, \{c_k\}) := \frac{1}{n} \sum_{u=1}^{n} \pi_{\rho,\{c_k\}}(X^{(u)}) + \mathsf{N}\left(0, \frac{\rho^2 \sigma^2}{n^2} I_d\right). \tag{5}$$

The privacy properties of mechanism (5) are immediate and we provide the statement in the appendix. We now turn to an investigation of the error of the mechanism (5), providing the following proposition (whose proof we give in the appendix).

Proposition 1. *Let* $m \geq 3, t \geq 0$, *and assume that for cluster probabilites* $P(c) = \sum_{j \in c} p_j$ *we have* $\rho \geq \min\{3mP(c) + 3 \log m + t, m\}$ *for each* $c \in \{c_k\}$. *Then there exists* $q \in \mathbb{R}^d_+$ *with* $\mathbf{1}^T q_c \leq P(c)$ *for each* $c \in \{c_k\}$ *and a numerical constants* $C_1 < \infty$,

$$\mathbb{E}\left[|\mathsf{M}_j(S, \rho, \{c_k\}) - mp_j|^2 \right] \leq C_1 \left[\frac{q_j^2}{2^{2t}} + \frac{mp_j}{n} + \frac{\sigma^2 \rho^2}{n^2} \right].$$

If $\rho \geq m$, *the preceding inequalities hold with* $t = \infty$.

Let us compare standard mechanism's errors with the element-level mechanism's errors, focusing on the squared error. For the user-level case, we have global sensitivity $\rho = m$, and the proposition shows that the mean-squared error for each coordinate scales as $\max\{\frac{mp_j}{n}, \frac{\sigma^2 m^2}{n^2}\}$. For element-level privacy, if we take $t = \log n$ in the definition of ρ, we obtain mean-squared error scaling as

$$\max_{c \in \{c_k\}} \left\{ \frac{mp_j}{n}, \frac{\sigma^2}{n^2} \left[m^2 P(c)^2 + \log^2 m + \log^2 n \right] \right\}.$$

Thus, whenever the individual contribution sizes m are large while probabilities of elements $P(c)$ are small, element-level mechanisms allow much more accurate estimation of frequencies than standard private noise addition. Of course, the best choice of the projection threshold ρ for element-level privacy requires some knowledge of the rough probabilities of each cluster, as otherwise, it is impossible to choose ρ appropriately; a two-stage estimator (to give rough upper bounds on the element probabilities $P(c)$) makes this feasible.

Statistical learning and risk minimization

Our final application is a fairly careful investigation of statistical learning problems in the context of element-level differential privacy and realistic federated learning problems, where individuals contribute more than a single data point (e.g. because they send many text messages). The typical statistical learning or generic M-estimation problem [Hastie, Tibshirani, and Friedman 2009, van der Vaart 1998] is as follows: for a sample space $\mathcal{X}$ and parameter space Θ, we have a loss $\ell : \Theta \times \mathcal{X} \to \mathbb{R}_+$, where $\ell(\theta; x)$ measures the loss of a parameter θ on observation x, and we wish to minimize the average loss over a population P. In standard empirical risk minimization or M-estimation, one receives $X^{(u)} \stackrel{\text{iid}}{\sim} P$, then chooses $\widehat{\theta}_n$ to minimize the empirical average $\frac{1}{n} \sum_{u=1}^{n} \ell(\theta; X^{(u)})$.

In our context of element privacy, we modify this slightly. Individuals (users) contribute batches of data $x \subset \mathcal{X}$, where the users are drawn from an underlying population P. We assume that there is a prespecified partition $\{c_1, \ldots, c_K\}$ of $\mathcal{X}$, so that the element of protection is whether a user with data $x = \{x_1, \ldots, x_m\}$ has any individual datum $x_i \in c_k$. Then the *element-level loss* for a data batch $x \in 2^{\mathcal{X}}$ averages losses within each element,

$$\ell_{\mathsf{el}}(\theta; x) := \sum_{k=1}^{K} 1\left\{x \cap c_k \neq \emptyset\right\} \frac{\sum_{x_i \in c_k} \ell(\theta; x_i)}{\operatorname{card}\{x_i \in c_k\}}, \tag{6}$$

that is, the sum of average losses in the non-empty elements in x. The idea of the averaging (6) is to make the loss insensitive to modification of data belonging to any single c_k. For an underlying population distribution P, we then wish to solve the risk minimization problem

$$\operatorname*{minimize}_{\theta \in \Theta} L_{\mathsf{el}}(\theta) := \mathbb{E}[\ell_{\mathsf{el}}(\theta; X)] = \int \ell_{\mathsf{el}}(\theta; x) dP(x). \tag{7}$$

Given a sample $S = \{X^{(u)}\}_{u=1}^n \sim P$, we approximate the risk (7) with $L^n_{\mathsf{el}}(\theta) := \frac{1}{n} \sum_{u=1}^{n} \ell_{\mathsf{el}}(\theta; X^{(u)})$, which we attempt to minimize as a proxy for (7). To describe our algorithms and their properties, however, we require a brief digression to provide a general analysis of stochastic approximation procedures under noise, giving an asymptotic convergence result that may be interesting independent of its privacy implications.

A private stochastic gradient method We now turn to the appropriate variant of the projected gradient method for privacy. The key from an element-level privacy perspective is to apply a projected gradient update on each of a user's elements, then average them together. Algorithm 1 captures this.

Because Algorithm 1 divides its updates into the clusters c_k before computing projections (clipping them to a particular radius) and updates, its combination with appropriate noise immediately yields several privacy properties. The most important result for us is to apply Alg. 1 in a stochastic-gradient-type scheme, which

Algorithm 1: Element-level projected gradient update $\mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, x)$

```
Require: Projection parameter $\rho$, stepsize $\alpha$, initial model $\theta_0$, partition of $\mathcal{X}$ into $\mathcal{C} = \{c_1, \ldots, c_K\}$, and user data $x = \{x_1, \ldots, x_m\}$
  for each $k \in \{1, \ldots, K\}$ such that $x \cap c_k \neq \emptyset$
    Set $\mathcal{B} = \{x_i : x_i \in c_k\}$
    $\theta_k^+ \leftarrow \mathsf{proj}_\Theta(\theta_0 - \alpha \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \nabla \ell(\theta_0; x))$
    $\Delta_k \leftarrow (\theta_k^+ - \theta_0)/\alpha$ and $[\Delta_k]_\rho \leftarrow \Delta_k \min\{1, \frac{\rho}{\|\Delta_k\|_2}\}$
  return $\sum_k [\Delta_k]_\rho$
```

allows us to both leverage the moments-accountant and convergence guarantees of stochastic gradient-type methods. Following the subsampling, for $q \in (0, 1)$ let $B_u \overset{\text{iid}}{\sim} \mathsf{Bernoulli}(q)$ or B_u be uniform on $\sum_u B_u = qn$, and for a sample $S = \{x^{(u)}\}_{u=1}^n$ define the subsampled mechanism

$$\mathsf{M}_q(S; \theta_0) \coloneqq \left(\sum_{u=1}^{n} B_u \cdot \mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, x^{(u)}) \right) + \mathsf{N}(0, \rho^2 \sigma^2 I).$$

For any sequence of stepsizes, we may define the *private stochastic approximation method*

$$\theta_{k+1} \coloneqq \theta_k - \alpha_k \frac{1}{qn} \mathsf{M}_q(S; \theta_k). \tag{8}$$

We consider the privacy of the iteration (8) both in the standard (user-level) private scenario and under element-level privacy. It is immediate that the update $\mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, \cdot)$ in Alg. 1 has element sensitivity at most 2ρ, where neighboring data x, x' guarantee $\|\mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, x) - \mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, x')\|_2 \leq 2\rho$. For standard privacy, we consider the global sensitivity of the update: assuming the upper bound $\mathrm{card}(x) \leq M$ on the cardinality of user data, we have $\|\mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, x) - \mathsf{sgd\text{-}el}^{\ell}_{\alpha,\rho}(\theta_0, x')\|_2 \leq 2(K \wedge M)\rho$ for any two sets $x, x' \subset \mathcal{X}$. We immediately obtain the following two corollaries on the privacy of the private stochastic gradient update (8).

Applications of element-level private stochastic approximation We now provide a generic convergence result with a brief application to generalized linear model estimation; our coming experiments evidence the utility of our definitions and mechanisms. We first recall the element-level population risk (7), which averages a standard loss ℓ into the element-level loss ℓ_{el}. The following result shows that the private stochastic iteration guarantees both asymptotic normality, and privacy. This result requires certain assumptions (such as Lipschitz conditions) which we provide in the appendix.

Proposition 2. *Under certain conditions (see Assumption A.1 in full version), define* $\overline{\theta}_k^n = \frac{1}{k} \sum_{i=1}^k \theta_i^n$*, where the number of iterations* $k = k(n)$ *satisfies* $\lim_n k(n)/n = \gamma$. *Let* $\Sigma_{\mathsf{el}} = \mathrm{Cov}(\nabla \ell_{\mathsf{el}}(\theta^\star; X))$ *and* $\Sigma = \nabla^2 L_{\mathsf{el}}(\theta^\star)^{-1} \left(\Sigma_{\mathsf{el}} + \frac{1}{\gamma} \left(\frac{1}{m} \Sigma_{\mathsf{el}} + \frac{\rho^2 \sigma^2}{m^2} I \right) \right) \nabla^2 L_{\mathsf{el}}(\theta^\star)^{-1}$. *Then*

$$\sqrt{n}(\overline{\theta}_k^n - \theta^\star) \xrightarrow{d} \mathsf{N}(0, \Sigma).$$

Fix $\delta > 0$ *and let* $\varepsilon(\tau) = \inf_\alpha \{ \frac{\gamma m^2}{n \tau^2} + \frac{\gamma m^2}{n \tau^2} \alpha + \frac{\log \delta^{-1}}{\alpha} \mid \alpha \leq \tau^2 \log \frac{n}{m} \}$ *for shorthand. Then*

(i) If $\sigma^2 \geq 2$*, then the collection* $\{\theta_i^n\}_{i=1}^k$ *is* $(O(1) \cdot \varepsilon(\sigma), \delta)$*-element-level differentially private.*

(ii) Assume each user data x *has cardinality at most* $\mathrm{card}(x) \leq M$*. If* $\sigma^2 \geq (K \wedge M)^2 \tau^2$*, where* $\tau^2 \geq 2$*, then* $\{\theta_i^n\}_{i=1}^k$ *is* $(O(\varepsilon(\tau)), \delta)$*-differentially private.*

As in the preceding examples, we see roughly the same tradeoffs between user-level (standard) and element-level privacy: for a given level ε, it is possible to provide the less-stringent element-level privacy with noise a factor $K \wedge M$ less than that for user-level privacy.

Experiments

To demonstrate the behavior of element-level private mechanisms, we present a series of experimental results in crowdsourced (federated) learning and stochastic optimization. We perform both simulations, where we may control all aspects of the experiments, and real-world experiments. Our theoretical results and intuition suggest that as the number of elements we consider grows—meaning that the elements provide a finer partition of the input space $\mathcal{X}$—we should observe performance improvements. In large-scale estimation, such as federated learning [McMahan et al. 2017], user data is rarely i.i.d. For example, some users take many photos of their children, others of dogs, others of hikes with friends; thus, a user may provide data only relating to a few elements. Motivated by this potential variability, for datasets with no pre-existing users, we diversify our experiments by constructing pseudo-users and assigning them varying numbers of elements.

In the remainder of the section, we present two experiments on fitting large image classification models, the first on tuning a model to a new dataset based on Flickr images, and the second an investigation on training a full neural network. The full version includes more experiments for histogram estimation and simulated logistic regression.

Large-scale image classification: Flickr

We investigate element-level privacy in the context of model fitting for a large image classification task. In this experiment, we vary several parameters: the privacy level $\varepsilon \in \{1, 3, \infty\}$, the number of distinct clusters into which we partition the input space ($K = 50, 500$), and, as we discuss in the introduction to the experiments, we also vary the diversity of images of individual users, so that we provide nominal "users" with data from 5, 30, or 100 distinct clusters/elements. As in the previous experiments, we expect the following: as the number of clusters K increases, element-level private methods should improve relative to the user-level private method, and similarly, as the diversity of individual users' images increases (the number of distinct elements), we expect to see further relative improvement. This is natural: in Algorithm 1 and the update (8), the magnitude of noise addition relative to the scale of a user's contribution decreases linearly in the number of distinct elements a user provides.

To this end, we perform a model tuning experiment on the Flickr dataset [Thomee et al. 2016] using a ResNet50 network [He et al. 2016] pre-trained on ImageNet [Deng et al. 2009], with reference implementation [Paszke et al. 2017]. This tuning means we fit only the *last layer* of the network, that is, we fit a multiclass logistic regression on input features $x \in \mathbb{R}^d$, $d = 2048$, defined by the outputs of the second-to-last ResNet50 layer. We use the 100 most popular Flickr image tags as labels, which represent 89% of the chosen data, and used an "unknown" label for anything remaining, resulting in a 101 class multiclass problem. To construct the elements into which we partition the images, we chose a uniformly random subset of 100,000 Flickr images, then used KMeans++ [Arthur and Vassilvitskii 2007] to cluster them into $K = 50$ and 500 clusters. Then a given image representation x simply belongs to the nearest cluster centroid. To fit the resulting model, we use the stochastic gradient method in Algorithm 1 as applied in the update (8). We construct a nominal collection of $n = 8000$ users, assigning each $m = 100$ labeled images (x, y). We vary the image allocations, so that (depending on the experiment) each user has images from on average $k = 5, 30, 100$ distinct elements. We perform $T = 40{,}000$ updates (8) in each experiment.

We present our results in Figure 2, plotting the maximum top-5 accuracy achieved versus iteration for many parameter settings. We simultaneously present results for different privacy levels ε, number K of clusters, and diversity of clusters per user. We highlight a few of the most salient points. First, user-level privacy with $\varepsilon = 1$ is substantially worse than any other method. Second, we see roughly what we expect, in that the element-level private algorithms achieve higher accuracy as the number of clusters and per-user diversity increase. Given that true internet-scale datasets are several times larger than the 400,000 image dataset we construct, this suggests the element-level private mechanisms can provide strong utility with satisfactory privacy.

Fully training a neural network: CIFAR10

We present our final experimental results for a classification problem on the CIFAR10 dataset [Krizhevsky and Hinton 2009], showing that it is possible to privately train a neural network while providing element-level privacy. We use the relatively simple convolutional neural network model architecture in the PyTorch tutorial [Paszke et al. 2017]. To construct the cluster centroids (elements), we mimic the method we propose for Flickr: we upsample the CIFAR image (using PyTorch), pass the resulting image through the pre-trained ResNet50 network above, and then cluster the resulting 2048-dimensional vectors using KMeans++ [Arthur and Vassilvitskii 2007] to construct $K = 100$ centroids that partition the CIFAR dataset.

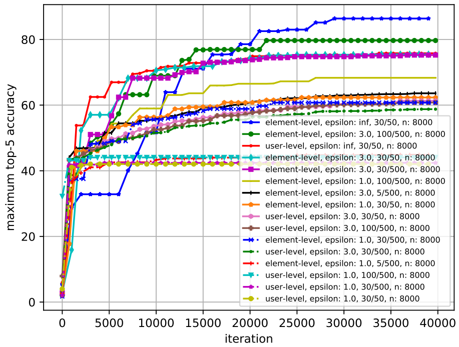


Figure 2. Training curves for the Flickr dataset. The legend ratio k/K represent the number of clusters (elements) a user has (k_u) over the total possible number of distinct clusters (K).

In Figure 3, we plot the difference in top-1 accuracy between a private method and the fully-trained (non-private) tutorial convolutional neural network [Paszke et al. 2017] against iteration, varying the privacy parameter ε and cluster diversity. As expected, we see two effects: first, as the sample size n grows, the accuracy improves; second, as the diversity of elements per user decreases, performance degrades as expected. All user-level private instantiations have accuracy more at least 15%-worse than the non-private accuracy. Conversely, the element-level-private algorithm with $\varepsilon = 3$, $n = 8000$, and high element diversity per-user (30/100 data clusters present) achieves top-1 accuracy nearly equal to non-private training.

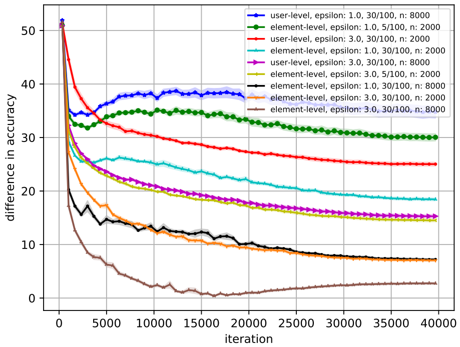


Figure 3. Difference in accuracy of a convolutional neural network model on the CIFAR10 dataset trained with privacy and without. Confidence interval are ±1.64 standard errors.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep Learning with Differential Privacy. In *23rd ACM Conference on Computer and Communications Security (ACM CCS)*, 308–318.

Andrés, M.; Bordenabe, N.; Chatzikokolakis, K.; and Palamidessi, C. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 901–914. Association for Computing Machinery.

Apple Differential Privacy Team. 2017. Learning with Privacy at Scale. Available at https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html.

Arthur, D.; and Vassilvitskii, S. 2007. k-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1027–1035.

Asi, H.; and Duchi, J. C. 2019. The importance of better models in stochastic optimization. *Proceedings of the National Academy of Sciences*, 116(46): 22924–22930.

Balle, B.; Barthe, G.; and Gaboardi, M. 2018. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *Advances in Neural Information Processing Systems 31*, 6277–6287.

Barber, R. F.; and Duchi, J. C. 2014. Privacy and Statistical Risk: Formalisms and Minimax Bounds. *arXiv:1412.4451 [math.ST]*.

Baumgartner, J. 2017. Reddit Comments.

Bhaskar, R.; Laxman, S.; Smith, A.; and Thakurta, A. 2010. Discovering frequent patterns in sensitive data. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Bhowmick, A.; Duchi, J.; Freudiger, J.; Kapoor, G.; and Rogers, R. 2018. Protection Against Reconstruction and Its Applications in Private Federated Learning. *arXiv:1812.00984 [stat.ML]*.

Bun, M.; and Steinke, T. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography Conference (TCC)*, 635–658.

Chatzikokolakis, K.; Andrés, M.; Bordenabe, N.; and Palamidessi, C. 2013. Broadening the Scope of Differential Privacy Using Metrics. In *The 13th Privacy Enhancing Technologies Symposium*, 82–102.

Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12: 1069–1109.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. ImageNet: a large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dong, J.; Roth, A.; and Su, W. J. 2019. Gaussian Differential Privacy. *arXiv:arXiv:1905.02383 [cs.LG]*.

Duchi, J. C. 2019. Information Theory and Statistics. Lecture Notes for Statistics 311/EE 377, Stanford University. Accessed May 2019.

Duchi, J. C.; Jordan, M. I.; and Wainwright, M. J. 2018. Minimax Optimal Procedures for Locally Private Estimation (with discussion). *Journal of the American Statistical Association*, 113(521): 182–215.

Duchi, J. C.; and Rogers, R. 2019. Lower Bounds for Locally Private Estimation via Communication Complexity. In *Proceedings of the Thirty Second Annual Conference on Computational Learning Theory*.

Duchi, J. C.; and Ruan, F. 2018. Stochastic Methods for Composite and Weakly Convex Optimization Problems. *SIAM Journal on Optimization*, 28(4): 3229–3259.

Duchi, J. C.; and Ruan, F. 2021. Asymptotic optimality in stochastic optimization. *The Annals of Statistics*, 49(1): 21–48.

Dwork, C. 2008. Differential privacy: a survey of results. In *Theory and Applications of Models of Computation*, volume 4978 of *Lecture Notes in Computer Science*, 1–19. Springer.

Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; and Naor, M. 2006a. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology (EUROCRYPT 2006)*.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006b. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference*, 265–284.

Dwork, C.; Naor, M.; Pitassi, T.; and Rothblum, G. 2010. Differential privacy under continual observation. In *Proceedings of the Forty-Second Annual ACM Symposium on the Theory of Computing*, 715–724. Association for Computing Machinery.

Dwork, C.; and Roth, A. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3 & 4): 211–407.

Dwork, C.; and Rothblum, G. 2016. Concentrated Differential Privacy. *arXiv:1603.01887 [cs.DS]*.

Erlingsson, U.; Pihur, V.; and Korolova, A. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS)*.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer, second edition.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Kasiviswanathan, S. P.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2013. Analyzing graphs with node differential privacy. In Sahai, A., ed., *Theory of Cryptog-*

raphy, volume 7785 of *Lecture Notes in Computer Science*, 457–476. Springer.

Kearns, M.; Roth, A.; Wu, Z. S.; and Yaroslavtsev, G. 2016. Private algorithms for the protected in social network search. *Proceedings of the National Academy of Sciences*, 113(4): 913–918.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

McCullagh, P.; and Nelder, J. 1989. *Generalized Linear Models*. London: Chapman and Hall.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2018. Learning Differentially Private Recurrent Language Models. In *Proceedings of the Sixth International Conference on Learning Representations*.

McSherry, F.; and Talwar, K. 2007. Mechanism design via differential privacy. In *48th Annual Symposium on Foundations of Computer Science*.

Mironov, I. 2017. Rényi Differential Privacy. In *30th IEEE Computer Security Foundations Symposium (CSF)*, 263–275.

Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *Neural Information Processing Systems (NIPS) Workshop on Automatic Differentiation*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of Empirical Methods for Natural Language Processing*.

Polyak, B. T.; and Juditsky, A. B. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4): 838–855.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *Annals of Mathematical Statistics*, 22: 400–407.

Smith, A. 2011. Privacy-preserving Statistical Estimation with Optimal Convergence Rates. In *Proceedings of the Forty-Third Annual ACM Symposium on the Theory of Computing*, 813–822. ACM.

Steinke, T.; and Ullman, J. 2017. Between Pure and Approximate Differential Privacy. *Journal of Privacy and Confidentiality*, 7(2): 3–22.

Thomee, B.; Shamma, D.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L. 2016. Yahoo Flickr Creative Commons 100M: The New Data in Multimedia Research. *Communications of the ACM*, 2(59): 64–73.

van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.