Generative Adversarial Privacy

Abstract

Providing provable privacy guarantees while preserving the utility of published datasets is a well-known challenge. We present a data-driven framework called generative adversarial privacy (GAP). Inspired by recent advancements in generative adversarial networks (GANs), GAP allows the data holder to learn the privatization mechanism directly from the data. Under GAP, finding the optimal privacy mechanism is formulated as a constrained minimax game between a privatizer and an adversary. We show that for appropriately chosen adversarial loss functions, GAP provides privacy guarantees against strong information-theoretic adversaries. We evaluate the performance of GAP on multi-dimensional Gaussian mixture models and the 11k Hands dataset.

1. Introduction

The use of machine learning algorithms for data analytics has recently seen unprecedented success for a variety of problems of practical relevance such as image classification, natural language processing, and prediction of consumer behavior, electricity use, political preferences, to name a few. The success of these algorithms hinges on the availability of large datasets, which are often crowd-sourced and contain private information. This, in turn, has led to privacy concerns and a growing body of research focused on developing privacy-guaranteed learning techniques. Moving towards randomization-based methods, in recent years, two distinct approaches with provable statistical privacy guarantees have emerged: (a) context-free approaches that assume worst-case dataset statistics and adversaries; (b) context-aware approaches that explicitly model the dataset statistics and adversary’s capabilities.

Context-free privacy. One of the most popular context-free notions of privacy is differential privacy (DP) (Dwork & Roth, 2014). DP, quantified by a leakage parameter ε, restricts distinguishability between any two “neighboring” datasets from the published data. DP provides strong, context-free theoretical guarantees against worst-case adversaries. However, training machine learning models on randomized data with DP guarantees often leads to a significantly reduced utility and comes with a tremendous increase in sample complexity (Fienberg et al., 2010; Wang et al., 2015; Yu et al., 2014; Karwa & Slavković, 2016; Duchi et al., 2016; Kairouz et al., 2016b) in the desired leakage regimes. For example, learning population level histograms under local DP suffers from a stupendous increase in sample complexity by a factor proportional to the size of the dictionary (Ye & Barg, 2017; Kairouz et al., 2016a; Duchi et al., 2016).

Context-aware privacy. Context-aware privacy notions fall under the domain of information theoretic (IT) privacy; see, for example, (Rebollo-Monedero et al., 2010; Calmon & Fawaz, 2012; Sankar et al., 2013; Salamatian et al., 2015; Basciftci et al., 2016). IT privacy has been predominantly quantified by mutual information (MI) which models how well an adversary, with access to the released data, can refine its belief about the private features of the data. More generally, one can also consider other measures that capture a range of adversarial capabilities including guessing (hard-decision) —all such measures exploit knowledge of the dataset alphabet and statistics.

In contrast to context-free, context-aware approaches achieve better privacy-utility tradeoffs by incorporating the statistics of the dataset and placing reasonable restrictions on the capabilities of the adversary. However, an inherent challenge in taking a context-aware privacy approach is that it requires having access to priors, such as joint distributions of public and private variables. Such information is hardly ever present in practice. Furthermore, in a data-driven setup, using information theoretic quantities (such as MI) as privacy metrics involves minimizing an empirical information theoretic loss function, a task that has been shown to be challenging in practice (Alemi et al., 2017).

Generative adversarial privacy. Given the challenges of existing privacy approaches, we take a fundamentally new approach towards enabling private data publishing with guarantees on both privacy and utility. Instead of adopting worst-case, context-free notions of data privacy, we introduce a novel context-aware model of privacy that allows the designer to cleverly add noise where it matters. We overcome the issue of statistical knowledge by taking a data-driven approach; specifically, we leverage recent advancements in generative adversarial networks (GANs) (Goodfellow et al., 2014; Mirza & Osindero, 2014) to introduce a framework for
context-aware privacy that we call generative adversarial privacy (GAP) (see Figure 1). Under GAP, the parameters of a generative model, representing the privatization mechanism, are learned from the data itself.

Our contributions. We list our main contributions below.

1. We introduce GAP as a minimax game-theoretic formulation (see Figure 1) to design privacy mechanisms matched to an adversarial model.

2. We show that our framework captures a rich class of statistical adversaries. This allows us to compare data-driven approaches directly against strong inferential adversaries (e.g., a maximum a posteriori (MAP) probability maximizing adversary with access to dataset statistics). This is in sharp contrast with most recent works on data-driven privacy (surveyed in Section 5) where privacy guarantees are only offered in the context of computational adversaries.

3. We make precise connections between data-driven privacy methods and the minimax game-theoretic GAP formulation; this implies that when: (i) the neural networks used in the data-driven approach have sufficient capacity, (ii) the learning rate is sufficiently small, and (iii) the training data is sufficiently large, the learned privacy scheme converges to the game-theoretically optimal one.

4. To showcase the power of our data-driven framework, we investigate a simple, albeit canonical, binary Gaussian Mixture Model (GMM) where \( Y \) is binary and \( X \) is a conditionally multi-dimensional Gaussian vector. We derive and compare the performance of game-theoretically optimal privatization mechanisms with those that are directly learned in a data-driven fashion to show that the gap between theory and practice is negligible.

5. Finally, we demonstrate the performance of GAP on a meaningful, widely used dataset identified as the 11k hands dataset for which we identify the public (images of hands) and sensitive (gender, race) features.

Paper outline. The remainder of our paper is organized as follows. We formally present our GAP model in Section 2. In Section 3, we present results for Gaussian mixture dataset models. In Section 4, we showcase the performance of GAP on the 11k Hands dataset. We review recent related works in Section 5 and conclude in Section 6. All proofs and algorithms are deferred to the supplementary material section.

2. Generative Adversarial Privacy Model

We consider a dataset \( D \) which contains both public and private variables for \( n \) individuals. We represent the public variables by a random variable \( X \in \mathcal{X} \), and the private variables (which are typically correlated with the public variables) by a random variable \( Y \in \mathcal{Y} \). Each dataset entry contains a pair of public and private variables denoted by \((X,Y)\). Instances of \( X \) and \( Y \) are denoted by \( x \) and \( y \), respectively. We assume that each entry pair \((X,Y)\) is distributed according to \( P(X,Y) \), and is independent from other entry pairs in the dataset. Since the dataset entries are independent and identically distributed (i.i.d), we restrict our attention to privatization schemes that only depend on \( X \).

We define \( \hat{Y} = h(g(X)) \) to be the adversary’s inference of the private variable \( Y \) from \( X \) using a decision rule \( h \). We allow for hard decision rules under which \( h(g(X)) \) is a direct estimate of \( Y \) and soft decision rules under which \( h(g(X)) = P_h(\cdot|g(X)) \) is a distribution over \( \mathcal{Y} \). To quantify the adversary’s performance, we use a loss function \( \ell(h(g(X = x)), Y = y) \) defined for every public-private pair \((x,y)\). Thus, the expected loss of the adversary with respect to \((\text{w.r.t.})\) \( X \) and \( Y \) is

\[
L(h, g) \triangleq \mathbb{E}[\ell(h(g(X)), Y)]
\]

where the expectation is taken over \( P(X,Y) \) and the randomness in \( g \) and \( h \).

Intuitively, the privatizer would like to minimize the adversary’s ability to learn \( Y \) reliably from the published data. This can be trivially done by releasing an \( X \) independent of \( X \). However, such an approach provides no utility for data analysts who want to learn non-private variables from \( X \). To overcome this issue, we capture the loss incurred by privatizing the original data via a distortion function \( d(\hat{x}, x) \), which measures how far the original data \( X = x \) is from the privatized data \( \hat{X} = \hat{x} \). Ensuring statistical utility in turn requires constraining the average distortion \( \mathbb{E}[d(g(X), X)] \) where the expectation is taken over \( P(X,Y) \) and the randomness in \( g \).

The data holder would like to find a privacy mechanism \( g \) that is both privacy preserving (in the sense that it is difficult for the adversary to learn \( Y \) from \( \hat{X} \)) and utility preserving (in the sense that it does not distort the original data too much). In contrast, for a fixed choice of privacy mechanism \( g \), the adversary would like to find a (potentially randomized) function \( h \) that minimizes its expected loss, which is equivalent to maximizing the negative of the expected loss. This leads to a constrained minimax game between the
privatizer and the adversary given by
\[
\min_{g(\cdot)} \max_{h(\cdot)} -L(h, g) \tag{2}
\]
\[
\text{s.t. } \mathbb{E}[d(g(X), X)] \leq D,
\]
where the constant \( D \geq 0 \) determines the allowable distortion for the privatizer and the expectation is taken over \( P(X,Y) \) and the randomness in \( g \) and \( h \).

**Theorem 1.** Under the class of hard decision rules, when \( \ell(h(g(x), y)) \) is the 0-1 loss function, the GAP minimax problem in (2) simplifies to
\[
\min_{g(\cdot)} \max_{y \in Y} P(y, g(X)) \tag{3}
\]
\[
\text{s.t. } \mathbb{E}[d(g(X), X)] \leq D,
\]
indicating that maximizing the probability of correctly guessing \( Y \) is the optimal adversarial strategy for any privatizer, i.e., the adversary uses the MAP decision rule. On the other hand, for a soft-decision decoding adversary (i.e., \( h = P_h(y|x) \) is a distribution over \( Y \)) under log-loss function \( \ell(h(g(X), y)) = \log \frac{1}{P_h(y|x)} \), the optimal adversarial strategy \( h^* \) is the posterior belief of \( Y \) given \( g(X) \) and the GAP minimax problem in (2) simplifies to
\[
\min_{g(\cdot)} I(g(X); Y) \tag{4}
\]
\[
\text{s.t. } \mathbb{E}[d(g(X), X)] \leq D,
\]
where \( I(g(X); Y) \) is the mutual information (MI) between \( g(X) \) and \( Y \).

The above theorem shows that GAP can recover MI privacy (under a log loss) and MAP privacy (under a 0-1 loss). The proof of Theorem 1 and a discussion about other loss functions are presented in Section A of the supplementary material.

### 2.1. Data-driven GAP

Thus far, we have focused on a setting where the data holder has access to \( P(X,Y) \). When \( P(X,Y) \) is known, the data holder can simply solve the constrained minimax optimization problem in (2) (game-theoretic version of GAP) to obtain a privatization mechanism that would perform best against a chosen type of adversary. In the absence of \( P(X,Y) \), we propose a data-driven version of GAP that allows the data holder to learn privatization mechanisms directly from a dataset \( D = \{(x_{(i)}, y_{(i)})\}_{i=1}^n \). Under the data-driven version of GAP, we represent the privacy mechanism via a generative model \( g(X; \theta_p) \) parameterized by \( \theta_p \). This generative model takes \( X \) as input and outputs \( \hat{X} \). In the training phase, the data holder learns the optimal parameters \( \theta_p \) by competing against a computational adversary: a classifier modeled by a neural network \( h(g(X; \theta_p); \theta_a) \) parameterized by \( \theta_a \). In the evaluation phase, the performance of the learned privacy mechanism can be tested under a strong adversary that is computationally unbounded and has access to dataset statistics. We follow this procedure in the next section.

We note that in theory, the functions \( h \) and \( g \) can be arbitrary. However, in practice, we need to restrict them to a rich hypothesis class. Figure 2 shows an example of the GAP model in which the privatizer and adversary are modeled as multi-layer neural networks. For a fixed \( h \) and \( g \), we can quantify the adversary’s empirical loss using cross entropy loss
\[
L_n(\theta_p, \theta_a) = \frac{1}{n} \sum_{i=1}^n y_{(i)} \log h(g(x_{(i)}; \theta_p); \theta_a) \tag{5}
\]
\[
+ (1 - y_{(i)}) \log(1 - h(g(x_{(i)}; \theta_p); \theta_a)),
\]
where \((x_{(i)}, y_{(i)})\) is the \( i \)-th row of \( D \). The optimal parameters for the privatizer and adversary are the solutions to
\[
\min_{\theta_p} \max_{\theta_a} -L_n(\theta_p, \theta_a) \tag{6}
\]
\[
\text{s.t. } \mathbb{E}_D[d(g(X; \theta_p), X)] \leq D,
\]
where the expectation is over \( D \) and the randomness in \( g \).

The minimax optimization in (6) is a two-player non-cooperative game between the privatizer and the adversary with strategies \( \theta_p \) and \( \theta_a \), respectively. In practice, we can learn the equilibrium of the game using an iterative algorithm presented in Algorithm 1 in Section B of the supplementary material. We first maximize the negative of the adversary’s loss function in the inner loop to compute the parameters of \( h \) for a fixed \( g \). Then, we minimize the privatizer’s loss function, which is modeled as the negative of the adversary’s loss function, to compute the parameters of \( g \) for a fixed \( h \). To avoid over-fitting and ensure convergence, we alternate between training the adversary for \( k \) epochs and training the privatizer for one epoch. This results in the adversary moving towards its optimal solution for small perturbations of the privatizer (Goodfellow et al., 2014). Observe that the hard constraint in (6) makes our minimax problem different from what is extensively studied in the machine learning community. The algorithmic approach and optimization techniques that we use to solve the constrained optimization in (6) are detailed in the supplementary material.

### 2.2. Our Focus

Our GAP framework is very general and can be used to capture many notions of privacy via appropriately chosen
loss functions that model adversarial capabilities. However, in the next section, we focus on a setting where $Y$ is binary and $X$ is conditionally multi-dimensional Gaussian. We consider a 0-1 loss which gives rise to a MAP adversary and show that privacy schemes that are learned in a data-driven fashion achieve an optimal performance when tested under a MAP adversary who has access to $P(X, Y)$ and knows the learned privacy scheme. For Section 4, we focus on the log-loss and evaluate the performance of GAP on the 11k Hands dataset.

3. GAP for Gaussian Mixture Models

In this section, we focus on a setting where $Y \in \{0, 1\}$ and $X$ is an $m$-dimensional Gaussian mixture random vector whose mean is dependent on $Y$. Let $P(Y = 1) = q$. Let $X|Y = 0 \sim \mathcal{N}(-\mu, \Sigma)$ and $X|Y = 1 \sim \mathcal{N}(\mu, \Sigma)$, where $\mu = (\mu_1, \ldots, \mu_m)$, and without loss of generality, we assume that $X|Y = 0$ and $X|Y = 1$ have the same covariance $\Sigma$.

We consider a MAP adversary who has access to $P(X, Y)$ and the privacy mechanism. The privatizer’s goal is to privatize $X$ in a way that minimizes the adversary’s probability of correctly inferring $Y$ from $X$. In order to have a tractable model for the privatizer, we mainly focus on linear (precisely affine) GAP mechanisms $X = g(X) = X + Z + \beta$, where $Z$ is an independently generated noise vector. This linear GAP mechanism enables controlling both the mean and covariance of the privatized data. To quantify utility of the privatized data, we use the $\ell_2$ distance between $X$ and $\hat{X}$ as a distortion measure to obtain a distortion constraint $E_{X, \hat{X}}\|X - \hat{X}\|^2 \leq D$.

3.1. Game-Theoretical Approach

Consider the setup where both the privatizer and the adversary have access to $P(X, Y)$. Further, let $Z$ be a zero-mean multi-dimensional Gaussian random vector. Although other distributions can be considered, we choose additive Gaussian noise for tractability reasons.

Without loss of generality, we assume that $\beta = (\beta_1, \ldots, \beta_m)$ is a constant parameter vector and $Z \sim \mathcal{N}(0, \Sigma_p)$. Following similar analysis in (Gallager, 2013), we can show that the adversary’s probability of detection is given by

$$P_d^{(C)} = q Q \left( \frac{\alpha}{2} + \frac{1}{\alpha} \ln \left( \frac{1-q}{q} \right) \right) + (1-q) Q \left( -\frac{\alpha}{2} - \frac{1}{\alpha} \ln \left( \frac{1-q}{q} \right) \right),$$

(7)

where $\alpha = \sqrt{(2\mu)^T (\Sigma + \Sigma_p)^{-1} 2\mu}$. Furthermore, since $E_{X, \hat{X}}[d(\hat{X}, X)] = E_{X, \hat{X}}\|X - \hat{X}\|^2 = E\|Z + \beta\|^2 = \|\beta\|^2 + \text{tr}(\Sigma_p)$, the distortion constraint implies that $\|\beta\|^2 + \text{tr}(\Sigma_p) \leq D$. To make the problem more tractable, we assume both $X$ and $Z$ are independent multi-dimensional Gaussian random vectors with diagonal covariance matrices. In this case, the optimal privacy mechanism is given by the solution of

$$\min_{\beta, \Sigma_p} \quad (2\mu)^T (\Sigma + \Sigma_p)^{-1} 2\mu$$

$$\text{s.t.} \quad \|\beta\|^2 + \text{tr}(\Sigma_p) \leq D.$$

Theorem 2. Consider GAP mechanisms given by $g(X) = X + Z + \beta$, where $X$ and $Z$ are multi-dimensional Gaussian random vectors with diagonal covariance matrices $\Sigma$ and $\Sigma_p$. Let $\{\sigma_1^2, \ldots, \sigma_m^2\}$ and $\{\sigma_1^2, \ldots, \sigma_m^2\}$ be the diagonal entries of $\Sigma$ and $\Sigma_p$, respectively. The parameters of the minimax optimal privacy mechanism are

$$\beta_i^* = 0, \quad \sigma_i^* = \left( \frac{|\mu_i|}{\sqrt{\lambda_i^*}} - \frac{\sigma_i^2}{\sqrt{\lambda_i^*}} \right)^+, \forall i = \{1, 2, \ldots, m\},$$

where $\lambda_i^*$ is chosen such that $\sum_{i=1}^m \left( \frac{|\mu_i|}{\sqrt{\lambda_i^*}} - \frac{\sigma_i^2}{\sqrt{\lambda_i^*}} \right)^+ = D$. For this optimal mechanism, the accuracy of the MAP adversary is given by (7) with $\alpha = 2 \sqrt{\sum_{i=1}^m \frac{\mu_i^2}{\sigma_i^2} + \left( \frac{|\mu_i|}{\sqrt{\lambda_i^*}} - \frac{\sigma_i^2}{\sqrt{\lambda_i^*}} \right)^+}$.

The proof of Theorem 2 is provided in the supplementary material C. We observe that the when $\sigma_i^2$ is greater than some threshold $\frac{|\mu_i|}{\sqrt{\lambda_i^*}}$, no noise is added to the data on this dimension due to the high variance. When $\sigma_i^2$ is smaller than $\frac{|\mu_i|}{\sqrt{\lambda_i^*}}$, the amount of noise added to this dimension is proportional to $|\mu_i|$: this is intuitive since a large $|\mu_i|$ indicates the two conditionally Gaussian distributions are further away on this dimension, and thus, distinguishable. Thus, more noise needs to be added in order to reduce the MAP adversary’s inference accuracy.

3.2. Data-driven Approach

For the data-driven linear GAP mechanism, we assume the privatizer only has access to the dataset $D$ with $n$ data samples but not the actual distribution of $(X, Y)$. Computing the optimal privacy mechanism becomes a learning problem.

In the training phase, the data holder learns the parameter of the GAP mechanism by competing against a computational adversary modeled by a multi-layer neural network. When convergence is reached, we evaluate the performance of the learned mechanism by comparing with the one obtained from the game-theoretic approach. To quantify the performance of the learned GAP mechanism, we compute the accuracy of inferring $Y$ under a strong MAP adversary that has access to both the joint distribution of $(X, Y)$ and the privacy mechanism.

Since the private variable $Y$ is binary, we measure the training loss of the adversary network by the empirical log-loss
We use synthetic datasets to evaluate the performance of GAP. For a fixed privatizer parameter \( \theta_p \), the adversary learns the optimal \( \theta_a^\ast \) by maximizing (9). For a fixed \( \theta_a \), the privatizer learns the optimal \( \theta_p^\ast \) by minimizing 

\[
L_n(\theta_p, \theta_a) = -\frac{1}{n} \sum_{i=1}^{n} y(i) \log h(g(x(i); \theta_p); \theta_a) + (1 - y(i)) \log (1 - h(g(x(i); \theta_p); \theta_a)).
\]

For a fixed privatizer parameter \( \theta_p \), the adversary learns the optimal \( \theta_a^\ast \) by maximizing (9). For a fixed \( \theta_a \), the privatizer learns the optimal \( \theta_p^\ast \) by minimizing 

\[
L_n(\theta_p, \theta_a) = -\frac{1}{n} \sum_{i=1}^{n} y(i) \log h(g(x(i); \theta_p); \theta_a) + (1 - y(i)) \log (1 - h(g(x(i); \theta_p); \theta_a)).
\]

The random noise \( Z \) is drawn from a \( n \)-dimensional independent zero-mean standard Gaussian distribution with covariance \( \Sigma_1 \).

To incorporate the distortion constraint into the learning process, we add a penalty term to the objective of the privatizer. Thus, the training loss function of the privatizer is given by

\[
L(\theta_p, \theta_a) = L_n(\theta_p, \theta_a) + \rho \max\{0, -\frac{1}{n} \sum_{i=1}^{n} d(g(x(i); \theta_p), x(i)) - D\},
\]

where \( \rho \) is a penalty coefficient which increases with the number of iterations. The added penalty consists of a penalty parameter \( \rho \) multiplied by a measure of violation of the constraint. This measure of violation is non-zero when the constraint is violated. Otherwise, it is zero.

3.3. Illustration of Results

We use synthetic datasets to evaluate the performance of the learned GAP mechanisms. Each dataset contains \( 20K \) training samples and \( 2K \) test samples. Each data entry is sampled from an independent multi-dimensional Gaussian mixture model. We consider two categories of synthetic datasets with \( P(Y = 1) \) equal to 0.75 and 0.5, respectively. Both the privatizer and the adversary in the GAP framework are trained on Tensorflow (Abadi et al., 2016) using Adam optimizer with a learning rate of 0.005 and a minibatch size of 1000. The distortion constraint is enforced by the penalty method as detailed in supplement B (see (10)).

Figure 4 illustrates the performance of the learned GAP mechanism against a strong theoretical MAP adversary for \( q = 0.75 \). The illustration for \( q = 0.5 \) is included in the supplementary material. It can be seen that the inference accuracy of the MAP adversary reduces as the distortion increases and asymptotically approaches (as expected) the prior on the private variable. This is because noise adding mechanisms cannot further reduce the accuracy of the MAP adversary than the prior on \( Y \). We also observe that the privacy mechanism obtained via the data-driven approach performs very well when pitted against the MAP adversary (maximum accuracy difference around 0.3% compared to the theoretical approach). In other words, for the Gaussian mixture data model with binary private variable, the data-driven version of GAP can learn privacy mechanisms that perform as well as the mechanisms computed under the theoretical version of GAP, which assumes that the privatizer has access to the underlying distribution of the dataset.

4. GAP for the 11K Hands Dataset

We consider the 11k Hands Dataset (Afifi, 2017) which consists of high-dimensional hand images with metadata. Our goal is to learn a GAP mechanism that restricts inferences on specific metadata variables denoted as private with limited distortion of the original images.
4.1. Privatizer model

The architecture of the privatizer is presented in Figure 5. Due to the high dimension of image data, we use an autoencoder to extract important features from the original image for reasons of both computational complexity and the sufficiently strong correlation between the meta data and a set of features representative of the image. Thus, we apply the GAP mechanism to the extracted features and reconstruct the original data using the computed decoder.

**Autoencoder.** We use a convolution autoencoder (CAE) (Masci et al., 2011) to compress and reconstruct the original image. The encoder network is comprised of an initial convolution layer ($5 \times 5$ filter size, 32 kernels, stride 2, padding 1), followed by three convolution layers with a filter size of $3 \times 3$ and stride 2. Each convolution layer consists of 64, 128, and 256 kernels, respectively. Between each convolution layer, we apply batch normalization and ReLU activation.

The decoder network consists of four deconvolution blocks with 256, 128, 64, and 32 deconvolution filters whose sizes are $3 \times 3, 4 \times 4, 6 \times 6,$ and $8 \times 8,$ respectively. Each block has a pair of stacked operations (batch normalization and ReLU activation).

**Privatization network.** The privatization network takes a low-dimensional random noise $\varepsilon$ and generate high-dimensional noise using multi-layer neural networks. The privatization network is comprised of an initial deconvolution layer with 64 filters (kernel size 13, stride 2, padding 7), followed by batch normalization, leaky ReLU activation, and another deconvolution layer with 128 filters (kernel size 9, stride 2, padding 6).

4.2. Adversary model

In our data-drive GAP, we model the adversary using state-of-the-art convolutional neural networks (CNNs). This architecture outperforms most of other models for image classification. Furthermore trained CNNs have been shown to be very effective in generic feature extraction (known as transfer learning features) for accurate image classification (Sharif Razavian et al., 2014). Many known CNNs, including AlexNet (Jia et al., 2014), VGG-Net (Simonyan & Zisserman, 2014), and ResNet (He et al., 2016), have demonstrated the success of transfer learning with initial weights pretrained by Imagenet on various image classification problems.

**Transfer learning & fine tuning.** For the adversary network, we apply the VGG-16 network with batch normalization on convolution layers by freezing the parameters of all the convolution layers and modifying the classification layers. Specifically, after using VGG-16 network that maps the input down to 1000-dimensional features, we append a leaky ReLu activation followed by a linear layer that halves 1000 neurons and output two neurons with the sigmoid activation. These two neurons represent the logits for the binary classification. Then we fine tune the parameters in the classification layers.

4.3. Loss functions

We use standard cross entropy loss to model the loss of the adversary for classification. For the privatizer’s loss function, we use the negative of the adversary’s loss and incorporate the distortion between the original and privatized images using the penalty method presented in (10). We model the distortion by the mean squared difference of pixels between original and privatized image per sample. To avoid the difficulties of vanilla GAN style training (Salimans et al., 2016), we modify the privatizer’s objective by adding maximizing the distance between the adversary’s output logits and one-hot encoded version of the label distribution. This can be considered as a heuristic that follows the idea of Wasserstein (Arjovsky et al., 2017) distance.

4.4. Experiment setup

We use the 11k Hands dataset to train and evaluate the GAP mechanism learned in a data-driven fashion. This dataset consists of 11076 hand images from 190 subjects of different ages under the same lighting conditions, labeled with the subject ID, gender, age, skin color, aspect of hand (i.e., left/right and dorsal/ventral), accessories, nail polish, and irregularities. In our experiment, we consider gender as the private information that a data holder wants to protect. The original image size is $1600 \times 1200.$ We apply center cropping and resizing to reduce the image size down to $224 \times 224$ so that we can utilize the state-of-the-art CNN architecture, such as VGG-16 network, to model the adversary network. We train and test the GAP model with images on the dorsal side.

We divide the dataset up to 10 batches. Each batch contains 2000 training samples and 1000 testing samples that are randomly drawn from the dorsal side dataset. Initially, we train the convolution autoencoder to achieve a low reconstruction error (0.015 mean squared error per pixel). As shown in Figure 6, the reconstructed images of the autoencoder output are almost identical to the original ones. Then we train the adversary neural network which achieves around 93% accuracy on gender classification. Using the reconstructed images from the output of the autoencoder, the adversary network can still achieve close to 90% accuracy.
In our experiment, we first train the autoencoder and adversary network to achieve good reconstruction and classification results. After we have pre-trained the autoencoder and adversary neural network, we start training the privatization network and the adversary network iteratively till convergence. Due to the difficulties in training the privatization network, we fix the autoencoder and train the privatization network first. Then we fix the privatization network and train the autoencoder for a few epochs.

4.5. Illustration of results

Figure 7 illustrates the privatized images for different values of distortion. It can be seen that the adversary’s accuracy of classifying private label (gender) decreases progressively as the distortion increases. When the distortion is small (0.013), the adversary’s classification accuracy is already reduced to 80% by using the learned GAP mechanism. When we increase the distortion to 0.34, the classification accuracy further decreases to 50%.

We also observe that the images get increasingly more mottled as the distortion increases. An ideal privatizer should restrict noise addition to a subset of features that are highly correlated with the private gender variable; however, it appears more noise is being added to more features thus mottling the entire image instead of only a specific subset of the features. This may also be due to the distortion constraint requiring the image to be as close to a hand as possible.

5. Related Work

In the context of publishing datasets with privacy and utility guarantees, a number of similar approaches have been recently considered. We briefly review them and clarify how our work is different. In (Xu et al., 2017), the authors consider linear privatizer and adversary models by adding noise in directions that are orthogonal to the public features in the hope that the “spaces” of the public and private features are orthogonal (or nearly orthogonal). This allows the privatizer to achieve full privacy without sacrificing utility. However, this work is restrictive in the sense that it requires the public and private features to be nearly orthogonal. Furthermore, this work provides no rigorous quantification of privacy and only investigates a limited class of linear adversaries and privatizers.

DP-based obfuscators for data publishing have been considered in (Hamm, 2016; Liu et al., 1987). The author in (Hamm, 2016) considers a deterministic, compressive mapping of the input data with differentially private noise added either before or after the mapping. The mapping rule is determined by a data-driven methodology to design minimax filters that allow non-malicious entities to learn some public features from the filtered data, while preventing malicious entities from learning other private features. The approach in (Liu et al., 1987) relies on using deep auto-encoders to determine the relevant feature space to add differentially private noise to, eliminating the need to add noise to the original data. After noise adding, the original signal is reconstructed. These novel approaches leverage minimax filters and deep auto-encoders to incorporate a notion of context-aware privacy and achieve better privacy-utility tradeoffs while using DP to enforce privacy. However, DP will still incur an insurmountable utility loss since it assumes worst-case dataset statistics. Our approach captures a broader class of randomization-based mechanisms via a generative model which allows the privatizer to tailor the noise to the statistics of the dataset.
Our work is also closely related to adversarial neural cryptography (Abadi & Andersen, 2016), learning censored representations (Edwards & Storkey, 2015), privacy preserving image sharing (Raval et al., 2017), and privacy-preserving adversarial networks (Tripathy et al., 2017) in which adversarial learning is used to learn how to protect communications by encryption or hide/remove sensitive information. Similar to these problems, our model includes a minimax formulation and uses adversarial neural networks to learn privatization schemes. However, in (Edwards & Storkey, 2015; Raval et al., 2017), the authors use non-generative auto-encoders to remove sensitive information. Instead, we use a GANs-like approach to learn privatization schemes that prevent an adversary from inferring the private data. We also go beyond these works by studying a game-theoretic setting with constrained optimization, which provides a specific privacy guarantee for a fixed distortion. We also compare the performance of the privatization schemes learned in an adversarial fashion with the game-theoretically optimal ones.

We use conditional generative models to represent privatization schemes. Generative models have recently received a lot of attention in the machine learning community (Goodfellow et al., 2014; Mirza & Osindero, 2014). Ultimately, deep generative models hold the promise of discovering and efficiently internalizing the statistics of the target signal to be generated. State-of-the-art generative models are trained in an adversarial fashion: the generated signal is fed into a discriminator which attempts to distinguish whether the data is real (i.e., sampled from the true underlying distribution) or synthetic (i.e., generated from a low dimensional noise sequence). Training generative models in an adversarial fashion has proven to be successful in computer vision and enabled several exciting applications. Analogous to how the generator is trained in GANs, we train the privatizer in an adversarial fashion by making it compete with an attacker.

6. Conclusion

We have introduced a novel generative adversarial privacy framework for designing data-driven context-aware privacy mechanisms with verifiable guarantees. GAP allows the data holder to learn the privatization mechanism directly from the dataset (to be published) without requiring access to the dataset statistics. Under GAP, finding the optimal privacy mechanism is formulated as a game between two players: a privatizer and an adversary. We showed that for appropriately chosen loss functions, GAP can provide guarantees against strong information-theoretic adversaries, such as a guessing MAP and belief-refining MI adversaries. We have also validated the performance of GAP on Gaussian mixture models and the 11k Hands dataset.

There are several fundamental questions that we seek to address. An immediate one is to develop techniques to rigorously benchmark data-driven results for large datasets against computable theoretical guarantees. One approach is to exploit recent developments in computing information-theoretic functionals (such as mutual information) in a data-driven manner; specifically, for the 11k Hands dataset, our current focus is on evaluating (from the data) mutual information between X and Y prior to privatization as well as that between g(X) and Y for the g() learned in a data-driven GAP fashion in Section 4 —from Fano’s inequality, mutual information lower bounds the accuracy of correctly guessing the sensitive variable from both the original and published datasets, and thereby, provides a benchmark for the performance of GAP. Such a comparison will allow us to provide provable privacy guarantees against computationally unbounded adversaries that have access to dataset statistics. More broadly, it will be interesting to investigate the robustness and convergence speed of the privacy mechanisms learned in a data-driven fashion. Finally, it will be also interesting to compare our approach to a context-free notion of privacy such as DP.
References


A. Proof of Theorem 1

Our minimax formulation places no restrictions on the adversary. Indeed, different loss functions and decision rules lead to different adversarial models. In what follows, we will discuss a variety of loss functions under hard and soft decision rules, and show how our GAP framework can recover several popular information theoretic privacy notions.

**Hard Decision Rules.** When the adversary adopts a hard decision rule, \( h(g(X)) \) is an estimate of \( Y \). Under this setting, we can choose \( \ell(h(g(X)), Y) \) in a variety of ways. For instance, if \( Y \) is continuous, the adversary can attempt to minimize the difference between the estimated and true private variable values. This can be achieved by considering a squared loss function

\[
\ell(h(g(X)), Y) = (h(g(X)) - Y)^2,
\]

which is known as the \( \ell_2 \) loss. In this case, one can verify that the adversary’s optimal decision rule is \( h^* = \mathbb{E}[Y|g(X)] \), which is the conditional mean of \( Y \) given \( g(X) \). Furthermore, under the adversary’s optimal decision rule, the minimax problem in (2) simplifies to

\[
\min_{g(\cdot)} -\text{mmse}(Y|g(X)) = \max_{Y} \mathbb{E}[Y^2|g(X)],
\]

subject to the distortion constraint. Here \( \text{mmse}(Y|g(X)) \) is the resulting minimum mean square error (MMSE) under \( h^* = \mathbb{E}[Y|g(X)] \). Thus, under the \( \ell_2 \) loss, GAP provides privacy guarantees against an MMSE adversary. On the other hand, when \( Y \) is discrete (e.g., age, gender, political affiliation, etc), the adversary can attempt to maximize its classification accuracy. This is achieved by considering a 0-1 loss function (Nguyen & Sanner, 2013) given by

\[
\ell(h(g(X)), Y) = \begin{cases} 
0 & \text{if } h(g(X)) = Y \\
1 & \text{otherwise}
\end{cases}.
\]

In this case, one can verify that the adversary’s optimal decision rule is the maximum a posteriori probability (MAP) decision rule: \( h^* = \arg\max_{y \in Y} P(y|g(X)) \), with ties broken uniformly at random. Moreover, under the MAP decision rule, the minimax problem in (2) reduces to

\[
\min_{g(\cdot)} - (1 - \max_{y \in Y} P(y, g(X))) = \max_{Y} \min_{g(\cdot)} P(y, g(X)) - 1,
\]

subject to the distortion constraint. Thus, under a 0-1 loss function, the GAP formulation provides privacy guarantees against a MAP adversary.

**Soft Decision Rules.** Instead of a hard decision rule, we can also consider a broader class of soft decision rules where \( h(g(X)) \) is a distribution over \( Y \); i.e., \( h(g(X)) = P_h(y|g(X)) \) for \( y \in Y \). In this context, we can analyze the performance under a log-loss

\[
\ell(h(g(X)), y) = \log \frac{1}{P_h(y|g(X))}.
\]

In this case, the objective of the adversary simplifies to

\[
\max_{h(\cdot)} -\mathbb{E}[\log \frac{1}{P_h(y|g(X))}] = -H(Y|g(X)),
\]

and that the maximization is attained at \( P_h^*(y|g(X)) = P(y|g(X)) \). Therefore, the optimal adversarial decision rule is determined by the true conditional distribution \( P(y|g(X)) \), which we assume is known to the data holder in the game-theoretic setting. Thus, under the log-loss function, the minimax optimization problem in (2) reduces to

\[
\min_{g(\cdot)} -H(Y|g(X)) = \max_{g(\cdot)} I(g(X); Y) - H(Y),
\]

subject to the distortion constraint. Thus, under the log-loss in (14), GAP is equivalent to using MI as the privacy metric (Calmon & Fawaz, 2012).

The 0-1 loss captures a strong guessing adversary; in contrast, log-loss or information-loss models a belief refining adversary. Next, we consider a more general \( \alpha \)-loss function that allows continuous interpolation between these extremes via

\[
\ell(h(g(X)), y) = \frac{\alpha}{\alpha - 1} \left( 1 - P_h(y|g(X))^1 - \frac{1}{\alpha} \right),
\]

for any \( \alpha > 1 \). It is easy to see that for large \( \alpha (\alpha \to \infty) \), this loss approaches that of the 0-1 (MAP) adversary. As \( \alpha \) decreases, the convexity of the loss function encourages the estimator \( \hat{Y} \) to be probabilistic, as it increasingly rewards correct
Generative Adversarial Privacy

inferences of lesser and lesser outcomes (in contrast to a hard decision rule by a MAP adversary of the most likely outcome) conditioned on the revealed data. As \( \alpha \to 1 \), (15) yields the logarithmic loss, and the optimal belief \( \hat{P}_Y \) is simply the posterior belief. Denoting \( H_\alpha^p(Y|g(X)) \) as the Arimoto conditional entropy of order \( \alpha \), one can verify that

\[
\max_{h(.)} \mathbb{E} \left[ \frac{\alpha}{\alpha - 1} \left( 1 - P_h(y|g(X))^{1 - \frac{1}{\alpha}} \right) \right] = -H_\alpha^p(Y|g(X)),
\]

which is achieved by a ‘\( \alpha \)-tilted’ conditional distribution

\[
P_h^p(y|g(X)) = \frac{P(y|g(X))^{\alpha}}{\sum_{y \in \mathcal{Y}} P(y|g(X))^{\alpha}}.
\]

Under this choice of a decision rule, the objective of the minimax optimization in (2) reduces to

\[
\min_{g(.)} -H_\alpha^p(Y|g(X)) = \min_{g(.)} I_\alpha^p(g(X); Y) - H_\alpha(Y),
\]

where \( I_\alpha^p \) is the Arimoto mutual information and \( H_\alpha \) is the Rényi entropy. Note that as \( \alpha \to 1 \), we recover the classical MI in the privacy setting and when \( \alpha \to \infty \), we recover the 0-1 loss.

B. Alternate Minimax Algorithm

In this section, we present the alternate minimax algorithm to learn the GAP mechanism from a dataset.

Algorithm 1 Alternating minimax privacy preserving algorithm

\textbf{Input:} dataset \( D \), distortion parameter \( D \), iteration number \( T \)

\textbf{Output:} Optimal privatizer parameter \( \theta_p \)

Initialize \( \theta_p^0 \) and \( \theta_a^0 \)

\textbf{for} \( t = 1, \ldots, T \) \textbf{do}

Random minibatch of \( M \) datapoints \( \{x(1), \ldots, x(M)\} \) drawn from full dataset

Generate \( \{\hat{x}(1), \ldots, \hat{x}(M)\} \) via \( \hat{x}(i) = g(x(i); \theta_p^t) \)

Update the adversary parameter \( \theta_a^{t+1} \) by stochastic gradient ascent for \( k \) epochs

\[
\theta_a^{t+1} = \theta_a^t + \alpha_t \nabla_{\theta_a} \frac{1}{M} \sum_{i=1}^M -\ell(h(\hat{x}(i); \theta_a), y(i)), \quad \alpha_t > 0
\]

Compute the descent direction \( \nabla_{\theta_p} l(\theta_p; \theta_a^{t+1}) \), where

\[
\ell(\theta_p; \theta_a^{t+1}) = -\frac{1}{M} \sum_{i=1}^M \ell(h(g(x(i); \theta_p); \theta_a^{t+1}), y(i))
\]

subject to \( \frac{1}{M} \sum_{i=1}^M d(g(x(i); \theta_p), x(i)) \leq D \)

Perform line search along \( \nabla_{\theta_p} l(\theta_p, \theta_a^{t+1}) \) and update

\[
\theta_p^{t+1} = \theta_p^t - \alpha_t \nabla_{\theta_p} \ell(\theta_p, \theta_a^{t+1})
\]

\textbf{end for}

\textbf{return} \( \theta_p^{t+1} \)

To incorporate the distortion constraint into the learning algorithm, we use the \textit{penalty method} (Lillo et al., 1993) and \textit{augmented Lagrangian method} (Eckstein & Yao, 2012) to replace the constrained optimization problem by a series of unconstrained problems whose solutions asymptotically converge to the solution of the constrained problem. Under the penalty method, the unconstrained optimization problem is formed by adding a penalty to the objective function. The added penalty consists of a penalty parameter \( \rho_t \), multiplied by a measure of violation of the constraint. The measure of violation is non-zero when the constraint is violated and is zero if the constraint is not violated. Therefore, in Algorithm 1, the constrained optimization problem of the privatizer can be approximated by a series of unconstrained optimization
problems with the loss function
\[
\ell(\theta_p, \theta_a^{t+1}) = -\frac{1}{M} \sum_{i=1}^{M} \ell(h(g(x(i); \theta_p); \theta_a^{t+1}), y(i)) + \rho_t \max\{0, \frac{1}{M} \sum_{i=1}^{M} d(g(x(i); \theta_p), x(i)) - D\},
\]
(17)
where \(\rho_t\) is a penalty coefficient which increases with the number of iterations \(t\). For convex optimization problems, the solution to the series of unconstrained problems will eventually converge to the solution of the original constrained problem (Lillo et al., 1993).

The augmented Lagrangian method is another approach to enforce equality constraints by penalizing the objective function whenever the constraints are not satisfied. Different from the penalty method, the augmented Lagrangian method combines the use of a Lagrange multiplier and a quadratic penalty term. Note that this method is designed for equality constraints. Therefore, we introduce a slack variable \(\delta\) to convert the inequality distortion constraint into an equality constraint. Using the augmented Lagrangian method, the constrained optimization problem of the privatizer can be replaced by a series of unconstrained problems with the loss function given by
\[
\ell(\theta_p, \theta_a^{t+1}, \delta) = -\frac{1}{M} \sum_{i=1}^{M} \ell(h(g(x(i); \theta_p); \theta_a^{t+1}), y(i)) + \rho_t (\frac{1}{M} \sum_{i=1}^{M} d(g(x(i); \theta_p), x(i)) + \delta - D)^2
\]
(18)
where \(\rho_t\) is a penalty coefficient which increases with the number of iterations \(t\) and \(\lambda_t\) is updated according to the rule \(\lambda_{t+1} = \lambda_t - \rho_t (\frac{1}{M} \sum_{i=1}^{M} d(g(x(i); \theta_p), x(i)) + \delta - D)\). For convex optimization problems, the solution to the series of unconstrained problems formulated by the augmented Lagrangian method also converges to the solution of the original constrained problem (Eckstein & Yao, 2012).

**C. Proof of Theorem 2**

The objective function in (8) can be written as
\[
2 \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_m \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2 + \sigma_p_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2 + \sigma_p_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_m^2 + \sigma_p_m^2} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{bmatrix} = \sum_{i=1}^{m} \frac{4\mu_i^2}{\sigma_p_i^2 + \sigma_i^2}.
\]
Thus, the optimization problem in (8) is equivalent to
\[
\begin{aligned}
\min_{\beta, \sigma_p_1^2, \ldots, \sigma_p_m^2} \sum_{i=1}^{m} \frac{\mu_i^2}{\sigma_p_i^2 + \sigma_i^2} \\
\text{s.t.} \quad ||\beta||^2 + tr(\Sigma_p) \leq D \\
\sigma_p_i^2 \geq 0 \quad \forall i \in \{1, 2, \ldots, m\}.
\end{aligned}
\]
(19)
Since a non-zero \(\beta\) does not affect the objective function but result in positive distortion, the optimal mechanism satisfies \(\beta = (0, \ldots, 0)\). Furthermore, the Lagrangian of the above optimization problem is given by
\[
L(\sigma_p_1^2, \ldots, \sigma_p_m^2, \lambda) = \sum_{i=1}^{m} \frac{\mu_i^2}{\sigma_p_i^2 + \sigma_i^2} + \lambda_0 (\sum_{i=1}^{m} \sigma_p_i^2 - D) - \sum_{i=1}^{m} \lambda_i \sigma_i^2,
\]
(20)
where \(\lambda = \{\lambda_0, \ldots, \lambda_m\}\) denotes the Lagrangian multipliers associated with the constraints. Taking the derivatives of \(L(\sigma_p_1^2, \ldots, \sigma_p_m^2, \lambda)\) with respect to \(\sigma_p_i^2, \forall i \in \{1, \ldots, m\}\) we have
\[
\frac{\partial L(\sigma_p_1^2, \ldots, \sigma_p_m^2, \lambda)}{\partial \sigma_p_i^2} = -\frac{\mu_i^2}{(\sigma_p_i^2 + \sigma_i^2)^2} + \lambda_0 - \lambda_i.
\]
(21)
Notice that the objective function in (8) is decreasing in \( \sigma_{p_i}^2 \), \( \forall i \in \{1, ..., m\} \). Thus, the optimal solution \( \sigma_{p_i}^* \) satisfies \( \sum_{i=1}^m \sigma_{p_i}^2 = D \). By the KKT conditions, we have
\[
\frac{\partial L(\sigma_{p_1}^2, ..., \sigma_{p_m}^2, \lambda)}{\partial \sigma_{p_i}^2} \bigg|_{\sigma_{p_i}^2 = \sigma_i^*} = -\frac{\mu_i^2}{(\sigma_{p_i}^* + \sigma_{p_i}^*)^2} + \lambda_0 - \lambda_i^* = 0. \tag{22}
\]
Since \( \lambda_i^*, i \in \{0, 1, ..., m\} \) is dual feasible, we have \( \lambda_i^* \geq 0, i \in \{0, 1, ..., m\} \). Therefore
\[
\lambda_0^* \geq \frac{\mu_i^2}{(\sigma_i^* + \sigma_{p_i}^*)^2}.
\]
If \( \lambda_0^* > \frac{\mu_i^2}{\sigma_i^2} \), we have \( \lambda_0^* > \frac{\mu_i^2}{(\sigma_i^* + \sigma_{p_i}^*)^2} \). This implies \( \lambda_i^* > 0 \). Thus, by complementary slackness, \( \sigma_{p_i}^* = 0 \). On the other hand, if \( \lambda_0^* < \frac{\mu_i^2}{\sigma_i^2} \), we have \( \sigma_{p_i}^* > 0 \). Furthermore, by the complementary slackness condition, \( \lambda_i^* \sigma_{p_i}^* = 0, \forall \sigma_{p_i}^* \). This implies \( \lambda_i^* = 0, \forall \sigma_{p_i}^* > 0 \). As a result, for all \( \sigma_{p_i}^* > 0 \), we have
\[
\frac{|\mu_i|}{\sqrt{\lambda_0}} = \sigma_i^2 + \sigma_{p_i}^*.
\tag{23}
\]
Therefore, \( \sigma_{p_i}^2 = \max\{\frac{|\mu_i|}{\sqrt{\lambda_0}} - \sigma_i^2, 0\} = \left(\frac{|\mu_i|}{\sqrt{\lambda_0}} - \sigma_i^2\right)^+ \) with \( \sum_{i=1}^m \sigma_{p_i}^2 = D \). Substitute this optimal solution into (7) with \( \alpha = \sqrt{(2\mu)^4 (\Sigma + \Sigma_p)^{-1} 2\mu} \), we obtain the accuracy of the MAP adversary.

**D. Performance of learned GAP mechanisms against MAP adversary for** \( q = 0.5 \)

Figure 9 compares the inference accuracy of the MAP adversary for GAP mechanisms obtained from both game-theoretical and data-drive approach under different distortion values. The synthetic dataset used in this simulation is sampled from a Gaussian mixture model with \( P(Y = 1) = 0.5 \).

![Figure 9: Performance of learned GAP mechanisms against MAP adversary for q = 0.5](image)

**D.1. Further Experiments on the 11k Hands Dataset**

We carry out an experiment that the private label are comprised of gender and skin color. The skin color consists dark, fair, median, very fair 4 categories, the gender has 2 classes, i.e. female and male. Thus we encoded them into 8 categories, in order to understand generative noise effect on more specific classes. We pre-train the adversarial that original image classification can achieve 69.3% accuracy. With incremental distortion tolerance, we observe that the overall classification accuracy drops down to 50.1%, 40.9% and 32% at the distortion level of 0.01, 0.11, and 0.34.
Figure 10: Privatized images for different distortion values. We consider the labels are both comprised by gender and skin color, e.g. (male, dark). From left to right the distortion goes from low to high. **Top-Left panel:** the raw images. **Top-right panel:** low perturbation at distortion 0.01. **Bottom-left panel:** median perturbation at distortion 0.11. **Bottom-right:** high perturbation at distortion 0.34.