Approximate, Adapt, Anonymize (3A): a Framework for Privacy Preserving Training Data Release for Machine Learning

Tamas Madl*, Weijie Xu, Olivia Choudhury, Matthew Howard*

Abstract

The availability of large amounts of informative data is crucial for successful machine learning. However, in domains with sensitive information, the release of high-utility data which protects the privacy of individuals has proven challenging. Despite progress in differential privacy and generative modeling for privacy-preserving data release in the literature, only a few approaches optimize for machine learning utility: most approaches only take into account statistical metrics on the data itself and fail to explicitly preserve the loss metrics of machine learning models that are to be subsequently trained on the generated data. In this paper, we introduce a data release framework, 3A (Approximate, Adapt, Anonymize), to maximize data utility for machine learning, while preserving differential privacy. The framework aims to 1) learn an approximation of the underlying data distribution, 2) adapt it such that loss metrics of machine learning models are preserved as closely as possible, and 3) anonymize by using a noise addition mechanism to ensure differential privacy. We also describe a specific implementation of this framework that leverages mixture models to approximate, kernel-inducing points to adapt, and Gaussian differential privacy to anonymize a dataset, in order to ensure that the resulting data is both privacy-preserving and high utility. We present experimental evidence showing minimal discrepancy between performance metrics of models trained on real versus privatized datasets, when evaluated on held-out real data. We also compare our results with several privacy-preserving synthetic data generation models (such as differentially private generative adversarial networks), and report significant increases in classification performance metrics compared to state-of-the-art models. These favorable comparisons show that the presented framework is a promising direction of research, increasing the utility of low-risk synthetic data release for machine learning.

Introduction

Training and using machine learning models in domains with sensitive and personally identifiable information such as healthcare presents significant legal, ethical and trust challenges; slowing the progress of this technology as well as its potential positive impact. Data synthesis has been proposed as a potential mechanism that can be a legally and ethically appropriate solution to the sharing and processing of sensitive data. It is possible to synthesize a dataset that allows accurate machine learning (ML) model training without compromising privacy, depending on privacy definition and requirements. In the case of the GDPR, for example, data that do not allow singling out may be excepted from the regulation (Cohen and Nissim 2020). In this work, we will rely on differential privacy (Dwork 2006), which unlike weaker privacy criteria such as k-anonymity has been shown to protect against singling out individuals (Cohen and Nissim 2020).

A privacy-preserving synthetic data generation framework maximizing utility for machine learning

For the purposes of privacy-preserving ML by means of differentially private (DP) data release, we are interested in approaches which 1) approximate the true data distribution, 2) preserve utility for machine learning (ML models trained on the data release perform similarly to models trained on true data), and 3) preserve privacy in accordance to DP.

More formally, we propose studying a class of data generation algorithms M which, given an original dataset $D = (X_i, Y_i)_{i=1}^n$ with n data points X_i and labels Y_i, produce a synthetic dataset $\tilde{D} = M(D)$, such that they

1. **Approximate** the underlying data distribution: estimate a parametric density $p_\theta(x)$ by optimizing a log-likelihood objective $L_1\left(p_\theta, D\right) := x \sim DE\left[-\log p_\theta(x)\right]$
2. **Adapt** the approximated data distribution such that the loss of a classifier f trained on data sampled from it is close to the loss of a classifier $\tilde{f}$ on the original data, under loss l:
   $\tilde{D} \sim p_\theta(x) L_2(D, \tilde{D}, f, \tilde{f}) = \left| E_{(x,y)\sim D}(\ell(f(x), y)) - E_{(x,y)\sim \tilde{D}}(\ell(\tilde{f}(x), y)) \right|$
   The overall optimization procedure needs to trade off the importance of L_1 the objective encouraging faithfully preserving the data distribution, and of L_2, the objective encouraging matching classifier loss: $L = \alpha L_1 + (1-\alpha) L_2$.
3. **Anonymize** by ensuring (ϵ, δ) differential privacy of the overall data publishing algorithm, such that the participation of a single data point is unlikely to be distinguish-

*These authors are from AWS. Others are from Amazon. Please contact mdlt@amazon.nl if you have any question.

able. That is, ensure the data publishing algorithm is differentially private (Dwork 2006).

Many instantiations of this general framework are possible. In this paper, we evaluate ClustMix, a simple algorithm instantiating these 3 steps. We will choose 1) a Gaussian Mixture Model as the density estimator (Bishop 2006), 2) the Kernel Inducing Point meta-learning algorithm (Nguyen, Chen, and Lee 2020) as the loss approximator (with an objective allowing a tradeoff between preserving density fidelity vs. preserving classifier fidelity), and 3) an improved version of Random Mixing to ensure privacy (Lee et al. 2018, 2019) - preserving combinations of data points, rather than individual data points, to facilitate a 'safety in numbers' approach to avoiding reidentification. Our main contributions are the flexible privacy-preserving data generation framework described above, and the introduction of cluster-based instead of random mixing for preserving differential privacy, which, together, allow significant accuracy increases over previously published methods (see Table 1).

The idea to create new training examples by taking convex combinations of existing data points has been successfully leveraged in machine learning, e.g. for data augmentation (Zhang et al. 2017a; Inoue 2018), learning with redundancy in distributed settings (Karakus et al. 2017), and more recently also for private machine learning (Lee et al. 2018). Lee et al. (Lee et al. 2019) leverage random mixtures (convex combinations of a randomly sampled subset of a dataset) and additive Gaussian noise to design a differentially private (DP) data release mechanism.

However, most of these methods ignore data geometry, sampling at random instead of explicitly attempting to preserve the original data distribution. This may lead to lower downstream utility for machine learning, as low-density regions around decision boundaries may not be preserved. Mixtures of random samples may also fail to preserve certain data distributions, such as skewed and multimode continuous variables.

In our approach, instead of random sampling, we sample from the immediate neighborhood of cluster centroids in order to preserve data distribution. Focusing on preserving cluster structure and mixing similar data points instead of random data points allows noisy mixtures to more closely approximate the original data distribution, and lose less utility than competing methods despite stronger DP guarantee.

Related Work

Privacy-preserving data publishing for machine learning tasks is an important problem, and several approaches have been proposed to address it. Mechanisms for publishing such data can operate in an interactive or non-interactive setting (Zhu et al. 2017). In an interactive setting, the data publishing method receives queries from users and responds with noisy outputs to preserve data privacy. The performance of interactive data publishing methods depend on several constraints, such as type of query, maximum number of queries, accuracy, and computational efficiency. Malicious users may still be able to deduce an output much closer to the original answer, thereby exposing the sensitivity of the dataset. In a non-interactive setting, the publishing method releases the data all at once in a privacy-preserving manner.

Among the state-of-the-art non-interactive approaches, synthetic data publishing is the most widely-used technique. This can be achieved by either anonymizing the dataset (Mohammed et al. 2011) or generating new samples based on the original data distribution (Zhang et al. 2017b; Blum, Ligett, and Roth 2013; Kasiviswanathan et al. 2011; Hardt, Ligett, and McSherry 2012). As noted in (Lee et al. 2019), computational complexity of most of the existing methods is exponential in the dimensionality of the dataset. Hence, they are not applicable to deep learning applications that commonly deal with high-dimensional data.

An alternate method is local perturbation that perturbs every data point with additive noise (Agrawal and Srikant 2000; Mishra and Sandler 2006). A relevant approach is random projection, which extracts lower-dimensional features via random projection and then perturbs them with some noise (Kenthapadi et al. 2012; Xu et al. 2017). In (Lee et al. 2019), the authors propose a new data publishing algorithm called Differentially Private Mix (DPMix), which generates a dataset by mixing ℓ randomly chosen data points and then perturbing them with an additive noise.

Generative Adversarial Network (GAN) (Goodfellow et al. 2014) is a well-known method for generating synthetic data from real data. As GANs do not provide any privacy guarantees by default, several methods have been proposed to modify it. PATE-GAN (Jordon, Yoon, and Van Der Schaar 2018) adapts the training procedure of the discriminator to be differentially private by using a modified version of the Private Aggregation of Teacher Ensembles (PATE) framework (Papernot et al. 2016, 2018). The Differentially Private GAN (DP-GAN) (Xie et al. 2018) aims to preserve privacy during training, by adding noise to the gradient of the Wasserstein distance. More recently, Differentially Private Mean Embeddings (DP-MERF) with Random Features have been suggested and shown to provide substantial utility improvements. DP-MERF leverages random Fourier feature representations to approximate the maximum mean discrepancy (MMD) objective in terms of two finite-dimensional mean embeddings, detaching the approximated embedding of the true data from that of the synthetic data. The former is the only term that is data dependent and requires privatization only once (and can then be re-used repeatedly to train a generator model).

In the Results section, we evaluate against these methods, and add a non-DP method for completeness's sake - ADS-GAN (Anonymization Through Data Synthesis Using Generative Adversarial Networks) (Yoon, Drumright, and Van Der Schaar 2020). In addition to training a generator model that minimizes discrepancy to the real data (in this case using Wasserstein distance), the objective function of ADS-GAN also penalizes an identifiability term directly. ADS-GAN formulates identifiability as the percentage of synthetic data points appearing in closer proximity to a real data point than the next-nearest real neighbor of that real data point (in other words, synthetic instances which may put real instances at risk of distance-to-closest-record attacks).

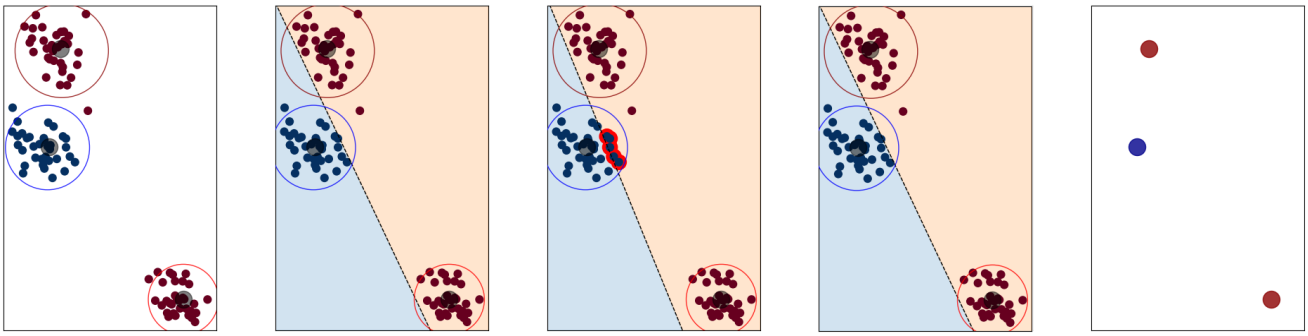

Figure 1: Illustration of the 3A (Approximate, Adapt, Anonymize) privacy-preserving data generation framework applied in a simple linearly separable setting. Panel 1 (*Approximate*): density estimation (in our case, Gaussian Mixture Models). Panel 2: baseline classifier trained on full data (dashed black line). Panel 3: classifier trained on sample from the density estimator (dashed black line), which misclassifies several data points close to the blue cluster (marked with thick red borders). Panel 4 (*Adapt*): adapted model which avoids these misclassifications (in our case, using Kernel Inducing Points (Nguyen, Chen, and Lee 2020) and trading off density vs. classifier fidelity). Panel 5 (*Anonymize*): differentially private dataset facilitating the same accuracy as the baseline classifier and minimizing identifiability risk by taking convex combinations of nearby data points ('safety in numbers') and adding noise according to Gaussian Differential Privacy (Dong, Roth, and Su 2019)

Methods

We describe a simple instantiation of the 3A framework, dubbed ClustMix. We will choose 1) a Gaussian Mixture Model as the density estimator (Bishop 2006), 2) the Kernel Inducing Point meta-learning algorithm (Nguyen, Chen, and Lee 2020) as the loss approximator (with an objective allowing a tradeoff between preserving density fidelity vs. preserving classifier fidelity), and 3) an improved version of Random Mixing to ensure privacy (Lee et al. 2018, 2019).

ClustMix is presented in Algorithm 2. The mixing component leverages the same intuition from previous research in data augmentation (Zhang et al. 2017a; Inoue 2018) and vicinal risk minimization (Chapelle et al. 2001) that convex combinations of samples from the vicinity of existing training examples tend to follow the original data distribution.

Approximation through Gaussian Mixture Models

We consider modeling the data distribution from the probabilistic perspective, approximating the probability density of the data by means of a finite mixture model (McLachlan, Lee, and Rathnayake 2019). The joint density over the training samples factorizes as

$$p(X, Z \mid \theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[p\left(z_n = k\right) p\left(x_n \mid \theta_k\right) \right]^{I[z_n=k]} \quad (1)$$

In ClustMix, before fitting the Gaussian Mixture Model, we randomly sliced data into n clusters and fit the model for each randomly sliced cluster. To be specific, for feature, we uniformly sample n data points as cutoff points. We apply the following algorithm to each small region. This makes sure each sample can only contribute to the creation of one cluster and creation of one synthetic sample. We also make the assumption of isotropic Gaussian in order to reduce computational complexity, but note that the algorithm can be made more general by relaxing this constraint.

Due to the privacy-utility tradeoff associated with the size of mixtures - too small mixtures incur privacy risk, as shown by (Lee et al. 2019). The exact minimum mixture size $l^{min}(\epsilon, \delta, \sigma_{max})$ for given privacy parameters (ϵ, δ) can be obtained from Eq. 7 by means of optimization

Thus, our data synthesis Algorithm 2 creates mixtures from clusters obtained from a Gaussian Mixture Model (GMM) adapted for size-constrained clustering, with the constraint that all cluster sizes have to be greater than or equal $l^{min}(\epsilon, \delta, \sigma_{max})$. We can reformulate Eq. 1 by factorizing to explicitly control the distribution over cluster sizes, as first proposed by (Jitta and Klami 2018):

$$p(X, Z \mid \theta) = \prod_{k=1}^{K} \left[p\left(s_k\right) \prod_{n=1}^{N} p\left(x_n \mid \theta_k\right)^{I[z_n=k]} \right], \quad (2)$$

Where s_k refers to the number of samples in cluster k. The assignment of data points to clusters can be obtained by finding the maximum of the joint log likelihood. X is the features and Z is latent distributions.

$$\log p(Z \mid X, \theta) = \log p(X \mid \theta, Z) + \log p(Z)$$
$$= \sum_n \log p\left(x_n \mid \theta_{z_n}\right) + \sum_k \log p\left(s_k\right) \quad (3)$$

$p(s)$ can be chosen to take the form of a step function, such that the probability is zero for all $s < l^{min}$ that are smaller than our minimum cluster size l^{min}, and uniformly nonzero otherwise. The optimum assignment based on Eq. 3 can then be found by expectation maximization, and yields cluster parameters and cluster assignments that approximate the data distribution given the size constraint (all clusters must contain at least l^{min} data points). We skip details for reasons of space (different solution approaches can be found in (Jitta and Klami 2018) and (Bishop 2006)).

Below, we will use $GMMConstrained(\mathcal{X}, n, l^{min})$ to denote a function taking a dataset $\mathcal{X}$ and a minimum cluster

size l^{min} as inputs, and producing n clusters (subsets of $\mathcal{X}$) as its output.

Adapt through Kernel Inducing Points

The Kernel Inducing Points (KIP) meta-learning algorithm (Nguyen, Chen, and Lee 2020) allows obtaining a smaller or distorted version of an original dataset which nevertheless maintains similar model performance as a model trained on the original data, and is thus uniquely well suited for the second step of our framework. It leverages kernel ridge regression, allowing a computationally efficient convex first order optimization approach to adaptation.

Let $D = (X_t, Y_t)_{i=1}^{n}$ be the original dataset, and $\tilde{D} = (\tilde{X_s}, \tilde{Y_s})_{i=1}^{s} \sim p_\theta(x)$ be a dataset obtained from the approximated model. In our case, ClustMix uses the cluster centroids as the smallest number of most representative samples (since the amount of noise that needs to be added to protect privacy rapidly grows with sample size - see next section).

Given a kernel K, KIP defines a kernel ridge regression loss function as follows:

$$L_{KRR}\left(X_s, y_s\right) = \frac{1}{2}\left\| y_t - K_{X_t X_s}\left(K_{X_s X_s} + \lambda I\right)^{-1} y_s \right\|_2^2 \tag{4}$$

where $\lambda > 0$ is a fixed regularization parameter. The KIP algorithm then minimizes Eq. 4 with respect to the dataset Xs. The optimization procedure is initialized with $\tilde{X}_s$, the synthetic dataset obtained from the approximated model through extracting centroids.

The kernel we use here is the Neural Tangent Kernel (Jacot, Gabriel, and Hongler 2018), mirroring the exact setup described in (Nguyen et al. 2021) ($neural_t angents$ library, Adam optimizer, z-scoring as preprocessing), with two exceptions: omitting any data augmentation, and using a fully connected architecture instead of a convolutional neural network in order to facilitate application to tabular data.

In our optimization procedure, we explicate a tradeoff between the importance of preserving data density and classifier accuracy. In addition to allowing users to make this choice explicitly (e.g. in domains where faithfully preserving density is important), this is also important for privacy, as allowing KIP to over-optimize on the loss may result in Xs regimes which are difficult to anonymize. A simple example may be a data point in Xs matching an outlier in X, thus easily exposing the identity of that outlier to simple privacy attacks. This can be avoided with a nonzero alpha in our procedure, which forces Xs staying arbitrarily close to GMM cluster centroids. Finally, a nonzero α acts as a regularizer and prevents overfitting to a decision boundary that may not generalize well.

$$\begin{aligned} L\left(X_s, y_s\right) &= \alpha . \frac{tr((X_s - \tilde{X_s})^T (X_s - \tilde{X_s}))}{tr(\tilde{X_s}^T \tilde{X_s})} \\ &+ (1-\alpha)\frac{1}{2}\left\| y_t - K_{X_t X_s}\left(K_{X_s X_s} + \lambda I\right)^{-1} y_s \right\|_2^2 \end{aligned} \tag{5}$$

The first term is the sum of normalized, α-weighted Euclidean distances between the data points in the original approximation set $\tilde{X}s$ and corresponding KIP-adapted data points Xs. Thus, $\alpha = 0$ would preserve the output of the Approximate step (in our case, GMM cluster centroids), and $\alpha = 1$ would simply yield the Xs most conducive to preserving classifier accuracy (ignoring density approximation). In our implementation, we obtain the optimal α as part of hyperparameter optimization. We apply KIP in each randomly sliced clusters.

Algorithm 1: Kernel Inducing Point (KIP)

Require: A target labeled dataset (X_t, y_t) along with a kernel or family of kernels.

1: Initialize a labeled support set (X_s, y_s).
2: **while** not converged **do**
3: Sample a random kernel. Sample a random batch $(\bar{X}_s, \bar{y}_s)$ from the support set. Sample a random batch $(\bar{X}_t, \bar{y}_t)$ from the target dataset.
4: Compute the kernel ridge-regression loss given by (4) using the sampled kernel and sampled support and target data.
5: Backpropagate through $\bar{X}_s$ (and optionally $\bar{y}_s$ and any kernel hyper-parameters) update the support set (X_s, y_s) by updating the subset $(\bar{X}_s, \bar{y}_s)$.
6: **end while**
7: **return** Learned support set (X_s, y_s)

Anonymize through Gaussian Differential Privacy

Once the data distribution is approximated and adapted, privacy-preserving synthetic data can be generated by a variety of approaches. (Lee et al. 2019) showed that a simple approach of taking linear combinations of data points yields promising results, provided that a sufficient number of instances are mixed. We follow the same approach, with the difference that we mix data points of the same cluster, rather than random data points, in order to more closely preserve the underlying data distribution. Finally, we add noise as derived from Gaussian Differential Privacy as the last ingredient of the algorithm shown below.

To obtain the accuracies reported in the Results, Algorithm 2 was re-run with different parameter choices for σ_{max}. Of the resulting datasets, the one yielding the highest accuracy against the training set when training a classifier on the synthetic data was taken as the final synthetic dataset to evaluate.

Differential privacy guarantees

We first provide some preliminaries and theorems on differential privacy for our proofs. We adopt gaussian privacy since it achieved better tradeoff between privacy and utility (Bu et al. 2019).

Definition 1. *(Adjacent Datasets) let two datasets* $S = (X_i, Y_i)_{i=1}^{n}$ *and* $S^{'} = (X_i^{'}, Y_i^{'})_{i=1}^{n}$ *be adjacent if they are identical except for one data point. We write* $S \sim S^{'}$. *(Dwork 2006)*

Definition 2. *(Differential Privacy) A data publishing algorithm* $M(S)$ *is called* (ϵ, δ) *differentially private if it satisfies,*

$$P[M(S) \in A] \leq e^{\epsilon} P[M(S^{'}) \in A] + \delta$$

M is an algorithm that publishes some statistics of the data. (Dwork 2006)

Here you can consider $M(S)$ is an array where each entry means a feature in a generated data point.

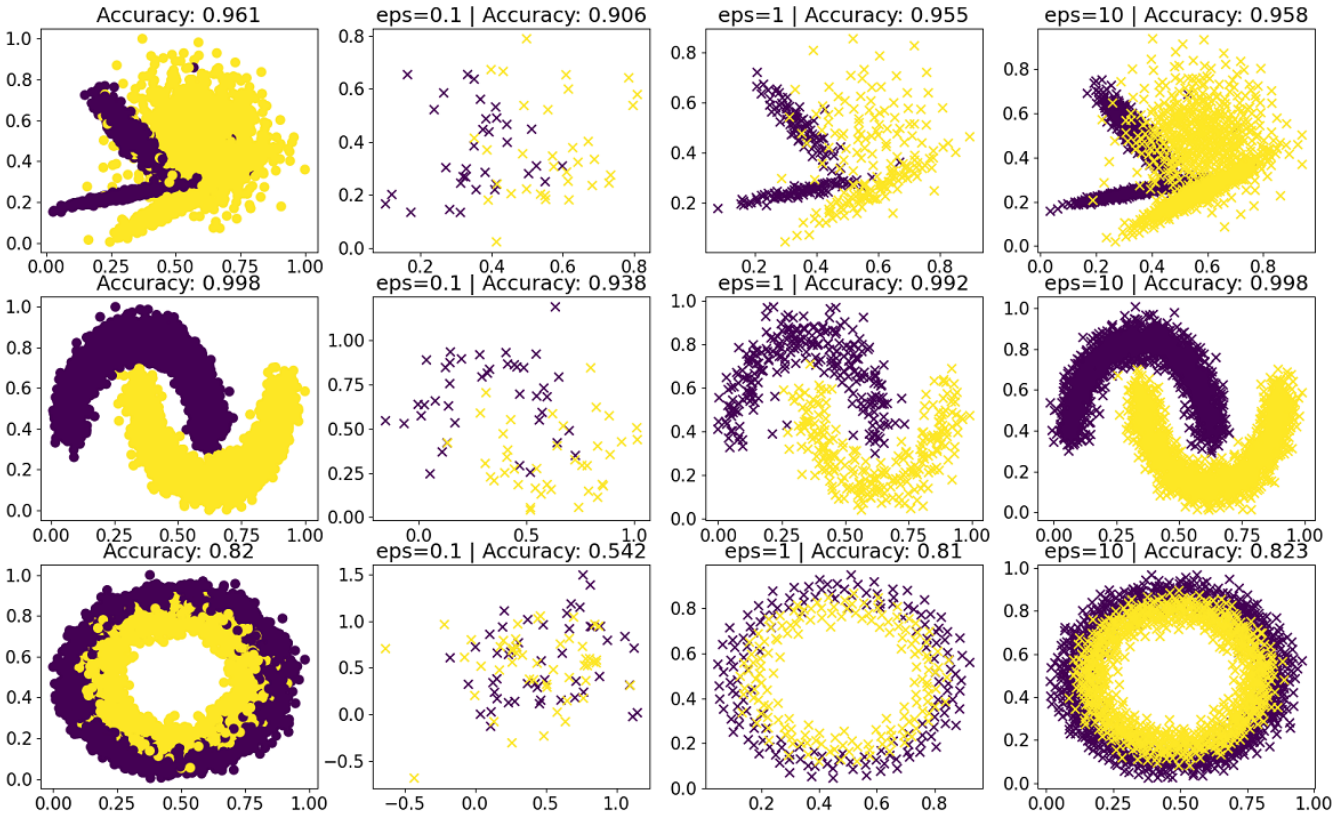

Figure 2: Example synthetic data points generated with different epsilon parameters, on three toy datasets.

Algorithm 2 ClustMix Synthetic Data Generator

Inputs:
- T instances in dataset $\mathcal{X} = \{x_i\}_{i=1}^{T}, x_i \in R^{D \times T} \mid 0 \leq x_i \leq 1$ with D features, number of classes C, labels y_i for each instance, privacy parameters ϵ_0 and δ_0
- Free parameter σ_{max}, the maximum allowable standard deviation of the additive Gaussian noise, α, the tradeoff between approximation and adaptation, and λ, a regularization parameter

Initialize:
- $S \leftarrow$ set of synthetic data points to be generated -
$l^{min} \leftarrow min_l |\delta_0 - \delta(\epsilon_0, \sigma_{max}, l, D)|$ as the minimum mixture size to ensure that the noise parameter doesn't exceed σ_{max}

for c = 1 to C **do**
  $\mathcal{X}_c \leftarrow \{x_i \mid y_i = c\}$ instances in class c
  $n \leftarrow \left\lceil \frac{|\mathcal{X}_c|}{l^{min}} \right\rceil$ as the number of clusters
  $\tilde{\mathcal{K}} \leftarrow$ GMMConstrained$(\mathcal{X}_c, n, l^{min})$
  $r^t \leftarrow \tilde{\mathcal{K}}_t$: instances in cluster t, and $N_t = |\tilde{\mathcal{K}}_t|$ clusters in c
  $\tilde{\mathcal{X}}_c \leftarrow \{\overline{r^1}, ..., \overline{r^N}\}$ cluster centroids: $\overline{r^t} = \frac{1}{N_t} \sum_{i=1}^{N_t} r_i^t$
end for
$\tilde{\mathcal{X}} \leftarrow \tilde{\mathcal{X}}_1 \cup ... \cup \tilde{\mathcal{X}}_C$
Backpropagate $\mathcal{X}$ until convergence with weighted KIP loss (initializing with $\tilde{\mathcal{X}}$ and staying as close to the GMM solution as indicated by α)

$$L(\mathcal{X}, y) = \alpha \frac{\text{tr}((\mathcal{X} - \tilde{\mathcal{X}})^T (\mathcal{X} - \tilde{\mathcal{X}}))}{\text{tr}(\tilde{\mathcal{X}}^T \tilde{\mathcal{X}})} + (1 - \alpha) \frac{1}{2} \| y - K_{\mathcal{X}\tilde{\mathcal{X}}} \left(K_{\tilde{\mathcal{X}}\tilde{\mathcal{X}}} + \lambda I \right)^{-1} y \|_2^2$$

for c = 1 to C **do**
  for t = 1 to $|\mathcal{K}|$ **do**
    Recompute cluster memberships based on adapted centroids, with each $\mathcal{K}_t$ containing $|\mathcal{K}_t| \geq l^{min}$ instances closest to adapted centroid $x_t \in \mathcal{X}$
    $l \leftarrow |r^t|$, where $l \geq l^{min}$
    $\sigma_t \leftarrow min_\sigma |\delta_0 - \delta(\epsilon_0, \sigma, l, D)|$ to obtain the standard deviation of Gaussian noise given privacy parameters, where δ depends on the given DP parameters as defined by Eq. (7)
    $Q_t \sim N(0, \sigma_t^2 I_D)$, where I is an identity matrix (size D)
    $x_t' \leftarrow \frac{1}{l} \sum_{i=1}^{l} r_i^t + Q_t$ as the cluster centroid with noise
    $S \leftarrow S \cup x_t'$
  end for
end for
return S: set of synthetic data points generated

Definition 3. *(trade-off function) For any two probability distributions P and Q on the same space, define the trade-off function $T(P, Q) : [0, 1] \rightarrow [0, 1]$ as,*

$$T(P, Q)(\alpha) = inf\{\beta_\phi : \alpha_\phi \leq \alpha\}$$

where the infimum is taken over all measurable rejection rules. α_ϕ and β_ϕ represent the distributions. (Dong, Roth, and Su 2019)

Definition 4. *(Gaussian Differential Privacy) A data publishing algorithm $M(DD)$ is called $\mu-$differentially private if it satisfies,*

$$T(M(S), M(S^{'})) \geq G_\mu$$

where $G_\mu(\alpha) = \Phi(\Phi^{-1}(1 - \alpha) - \mu)$ where ϕ denotes standard normal CDF. (Dong, Roth, and Su 2019)

Definition 5. *(Sensitivity) Assume $\theta(S)$ is a univariate statistic of the dataset. The sensitivity of θ is*

$$sens(\theta) = sup_{S,S'} |\theta(S) - \theta(S^{'})|$$

(Dwork 2006)

Theorem 1. *A Gaussian mechanism that operates on a statistic θ as $M(S) = \theta(S) + \xi$, where $\xi \sim \mathcal{N}\left(0, sens(\theta)^2/\mu^2\right)$, is μ-GDP.*

Proof. See theorem 2.7 in (Dong, Roth, and Su 2019) □

Corollary 1. *A mechanism is μ-GDP if and only if it is $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$. Where,*

$$\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon}{\mu} + \frac{\mu}{2}\right) - \mathrm{e}^{\varepsilon}\Phi\left(-\frac{\varepsilon}{\mu} - \frac{\mu}{2}\right)$$

Proof. See Corollary 2.13 in (Dong, Roth, and Su 2019) □

Corollary 2. *The n-fold composition of μ_i-GDP mechanisms is $\sqrt{\mu_1^2 + ... + \mu_n^2}$-GDP*

Proof. See Corollary 3.3 in (Dong, Roth, and Su 2019) □

Theorem 2. *Consider T data points consisting of a feature matrix $X := [X_1 X_2 ... X_n] \in R^{D \times T}$ and X is normalized such that $X \in [0,1]^{D \times T}$. We obtain each synthetic data point X_t' by averaging l original points, and adding Gaussian noise:*

$$X_t' = \frac{1}{l}\sum_{i=1}^{l} X_{t_i} + Q_t, \tag{6}$$

where Qt is sampled from $N(0, \sigma_t^2 I_D)$ and $l \leq D$.

Suppose each original data point can only be used one time in the generation of the synthetic dataset and the number of category equals to C, this data publishing algorithm is $(\epsilon, \delta(\epsilon, \sigma, l, C, D))$-DP for all $\epsilon \leq 0$, where $\delta(\epsilon, \sigma, l, C, D)$ is given by

$$\delta(\epsilon, \sigma, l, D) = \Phi(-\frac{\epsilon l\sigma}{\sqrt{CD}} + \frac{\sqrt{CD}}{2l\sigma}) - e^{\epsilon}\Phi(-\frac{\epsilon l\sigma}{\sqrt{CD}} - \frac{\sqrt{CD}}{2l\sigma}), \tag{7}$$

and Φ denotes the standard normal CDF of $N(0,1)$.

Proof. If we change a single data point in the original data from X_1 to $X_1^{'}$, this will at most change a single synthetic data entry from X_t to $X_t^{'}$. By Definition 5 and Equation 6, we have

$$\begin{aligned} sensitivity &= sup|X_t - X_t^{'}| \\ &= sup|\frac{1}{l}(X_1 - \frac{1}{l}X_1^{'})| \\ &\leq \frac{1}{l} \end{aligned}$$

as each data point in $X \in [0,1]^{D \times T}$. Thus, the sensitivity is $\frac{1}{l}$

Let $\mu = \frac{1}{l\sigma}$ and we have Q_t is sampled from $N(0, \sigma_t^2 I_D)$ Since

$$\begin{aligned} \sigma_t^2 &= (\frac{l\sigma}{l})^2 = (\frac{\frac{1}{l}}{\frac{1}{l\sigma}})^2 \\ &= (\frac{\frac{1}{l}}{\mu})^2 \end{aligned}$$

Since the sensitivity is $\frac{1}{l}$ and based on theorem 1, we can show that our mechanism is $\frac{1}{l\sigma}$-GDP.

Since changing one original data point can influence a maximum of one synthetic data point and D features, $\mu = \sqrt{1/(l\sigma)^2 \times D} = \frac{\sqrt{D}}{l\sigma}$. However, since we use KIP algorithm, one single point could potentially influence other classes in that region $\mu = \sqrt{1/(l\sigma)^2 \times D \times C} = \frac{\sqrt{CD}}{l\sigma}$. Since there are C classes, Based on Corollary 2, our data publishing algorithm is thus $\frac{\sqrt{CD}}{l\sigma}$-GDP

Corollary 1 in (Dong, Roth, and Su 2019) states that a mechanism is μ-GDP if and only if it is $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$. Here, $\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon}{\mu} + \frac{\mu}{2}\right) - \mathrm{e}^{\varepsilon}\Phi\left(-\frac{\varepsilon}{\mu} - \frac{\mu}{2}\right)$, in order to ensure that a Gaussian output perturbation mechanism is $(\epsilon, \delta(\epsilon))$-DP, as first shown by (Balle and Wang 2018). Substituting $\mu = \frac{\sqrt{CD}}{l\sigma}$, our data publishing algorithm is thus $(\epsilon, \delta(\epsilon))$-DP, where $\delta(\epsilon)$ is given by

$$\delta(\epsilon, \sigma, l, C, D) = \Phi(-\frac{\epsilon l\sigma}{\sqrt{CD}} + \frac{\sqrt{CD}}{2l\sigma}) - e^{\epsilon}\Phi(-\frac{\epsilon l\sigma}{\sqrt{CD}} - \frac{\sqrt{CD}}{2l\sigma})$$

□

Since each data point in the original set can be used by one randomly selected cluster to fit on GMM, and we only apply KIP algorithm to this region, it can only influence the generation of C sample. Thus, our data publishing algorithm is differentially private.

Data and Experimental Setup

Implementation Details: we leverage an extension of Gaussian Mixture Model implementation of scikit-learn for the Approximate step (adapted to ensure clusters never contain fewer data points than the given constraint - see Section 3.1; reverting to constrained KMeans at timeout or if no solution was found), and an open-source implementation of KIP[1] for the Adapt step, modified to optimize for the weighted objective function in Eq. eq:kip, trading off faithfulness to the output of the Approximate step and the utility for ML by means of the α parameter. Noise addition is based on GDP; we obtain the standard deviation of the Gaussian to sample noise from Eq. eq:epsilon by means of optimization, as it does not have a closed-form solution. Specifically, we use Nelder-Mead procedure (Olsson and Nelson 1975) to obtain the smallest possible noise that can be added that ensures that given DP parameters ϵ and δ are not exceeded (and discard generated data points where no such solution can be found). Obtained noise parameters were cached for computational performance.

Data preprocessing: data features were scaled to the interval [0,1] for ClustMix to simplify proportionate noise addition (using min-max scaling or a Normal quantile transform, whichever worked better during hyperparameter optimization). For the only dataset with continuous target variables (MIMIC-III hospital length of stay), lengths of stay were discretized into 4 buckets for classification (less than 3 days, less than a week, less than two weeks, or two weeks or more).

Hyperparameter tuning: for all benchmark algorithms as well as for ClustMix, we use heteroscedastic evolutionary Bayesian optimisation (HEBO) (Cowen-Rivers et al. 2020) to optimize free parameters - over 30 iterations for all approaches - excepting DP parameters which are fixed ($\epsilon = 1$

[1] https://github.com/google-research/google-research/tree/master/kip

	samples	classes	features	Real	DP-GAN	PATE-GAN	ADS-GAN	DP-MERF	ClustMix
adult AUC	32,561	2	14	0.931	0.511	0.732	0.821	0.650	**0.863**
census AUC	299,285	2	40	0.952	0.529	0.544	0.856	0.686	**0.906**
credit AUC	284,807	2	29	0.973	0.435	0.739	0.911	0.772	**0.969**
isolet AUC	4,366	2	617	0.988	0.618	0.529	**0.924**	0.547	0.903
MIMIC3 EHR Mortality AUC	58,976	2	18	0.896	0.632	0.473	0.530	0.728	**0.749**
MIMIC3 EHR Length of stay AUC	57,171	4	25	0.728	0.497	0.501	0.550	0.508	**0.624**
MNIST Accuracy $\delta = 10^{-5}$	60,000	10	784	0.972	0.403	0.563	0.710	0.650	**0.818**
Avg. score difference compared to real				0	0.402	0.337	0.162	0.271	**0.087**

Table 1: Results of evaluating classifiers trained on synthetic data and tested on real held-out test sets. DP models employ (1,1/N)-DP with N being sample size, except for MNIST, where $\delta = 10^{-5}$ (following the authors of the compared approaches). Length of Stay was discretized into 4 categories ($\leq$ 3 days, $\leq$ 1 week, $\leq$ 2 weeks or $\geq$ 2 weeks) to facilitate classification, and evaluated by one vs. one AUC

and $\delta = 1/N$ with N being sample size; except for MNIST, where $\delta = 10^{-5}$ to facilitate easier comparison with benchmark papers using the same approach).

Benchmark Datasets and Methods: We compare with state-of-the-art DP data generation algorithms such as DP-GAN, DP-MERF, PATE-GAN. We also included non DP data generation algorithm ADS-GAN as it can lower the risk of information leakage. We compare on 7 different real datasets which we list deatils of datasets in Table 2.

Evaluation: qualitative results (i.e. MNIST images, and plots of toy data) can be inspected in Appendix A. Quantitative results are presented in Table 1 using Area Under the ROC Curve (AUC), micro-averaged (i.e. one vs. one for all pairwise combinations) in the case of multi-class classification, except for MNIST, conventionally evaluated using accuracy. For each dataset, we set aside 25% of the data as a held-out test set (except MNIST, which has its own conventionally used test set, comprising 14% of the total data). 'Real' performance metrics are calculated by training a LightGBM model of gradient boosted decision trees (Ke et al. 2017) on the training set and evaluating on the test set. For each benchmark method and for ClustMix, performance metrics are calculated by first running the method to generate a privacy-preserving synthetic training set based on the real training set, then training a LightGBM model on that synthetic training set, and evaluating it on the real test set.

Results

We evaluate ClustMix against several states of the art differentially private data generation approaches in Table 1. The combination of cluster-based instead of random mixing for preserving differential privacy (unlike similar mixing-based approaches (Lee et al. 2018, 2019)), as well as adaptation for machine learning utility, result in significantly higher predictive accuracy of models trained on synthetic and tested on real data, when compared against differentially private methods.

The mechanism can be more clearly understood from the simple 2D toy datasets illustrated in Figure 2. In an attempt to pick σ_{max} that maximizes utility on the training set, ClustMix is forced to generate a very small number of noisy cluster centroids when $\epsilon = 0.1$ (which ensures large l and thus minimizes the additive noise), but yields many low-noise centroids at $\epsilon = 10$ (since at this ϵ, smaller clusters are sufficient to ensure that the noise levels are small enough).

Similar observations apply to the higher-dimensional MNIST digit dataset (Figure 3), where larger clusters yield the optimal accuracy at $\epsilon = 1$ (each digit is a linear combination of 118 training digits on average, with added noise), but smaller clusters are more helpful at $\epsilon = 0.2$ (where each digit is obtained by averaging 20 training digits plus noise).

	# samples	# classes	# features
adult	32,561	2	14
census	299,285	2	40
credit	284,807	2	29
isolet	4366	2	617
MIMIC3 mortality	58,976	2	18
MIMIC3 length of stay	57,171	4	25
MNIST	60,000	10	784

Table 2: Dataset characteristics

Conclusion

We propose a simple framework for privacy-preserving synthetic data generation that explicitly takes into account machine learning utility, in addition to faithfully modeling data distribution. We have also presented a specific instantiation of this framework, ClustMix, and have presented substantial increases in classification performance metrics compared to state-of-the-art models; implying that the presented framework constitutes a promising direction of research increasing the utility of low-risk synthetic data release for machine learning.

Appendix

MNIST Example

Acknowledgements Many thanks to Cemre Zor for helpful comments on the Manuscript.

References

Agrawal, R.; and Srikant, R. 2000. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 439–450.

Balle, B.; and Wang, Y.-X. 2018. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, 394–403. PMLR.

Bishop, C. M. 2006. Pattern recognition. *Machine learning*, 128(9).

Blum, A.; Ligett, K.; and Roth, A. 2013. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2): 1–25.

Bu, Z.; Dong, J.; Long, Q.; and Su, W. J. 2019. Deep Learning with Gaussian Differential Privacy.

Chapelle, O.; Weston, J.; Bottou, L.; and Vapnik, V. 2001. Vicinal risk minimization. *Advances in neural information processing systems*, 416–422.

Cohen, A.; and Nissim, K. 2020. Towards formalizing the GDPR's notion of singling out. *Proceedings of the National Academy of Sciences*, 117(15): 8344–8352.

Cowen-Rivers, A. I.; Lyu, W.; Wang, Z.; Tutunov, R.; Jianye, H.; Wang, J.; and Ammar, H. B. 2020. Hebo: Heteroscedastic evolutionary bayesian optimisation. *arXiv e-prints*, arXiv–2012.

Dong, J.; Roth, A.; and Su, W. J. 2019. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*.

Dwork, C. 2006. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, 1–12. Springer.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hardt, M.; Ligett, K.; and McSherry, F. 2012. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems*, 25.

Inoue, H. 2018. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Jitta, A.; and Klami, A. 2018. On controlling the size of clusters in probabilistic clustering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jordon, J.; Yoon, J.; and Van Der Schaar, M. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*.

Karakus, C.; Sun, Y.; Diggavi, S.; and Yin, W. 2017. Straggler mitigation in distributed optimization through data encoding. *Advances in Neural Information Processing Systems*, 30: 5434–5442.

Kasiviswanathan, S. P.; Lee, H. K.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2011. What can we learn privately? *SIAM Journal on Computing*, 40(3): 793–826.

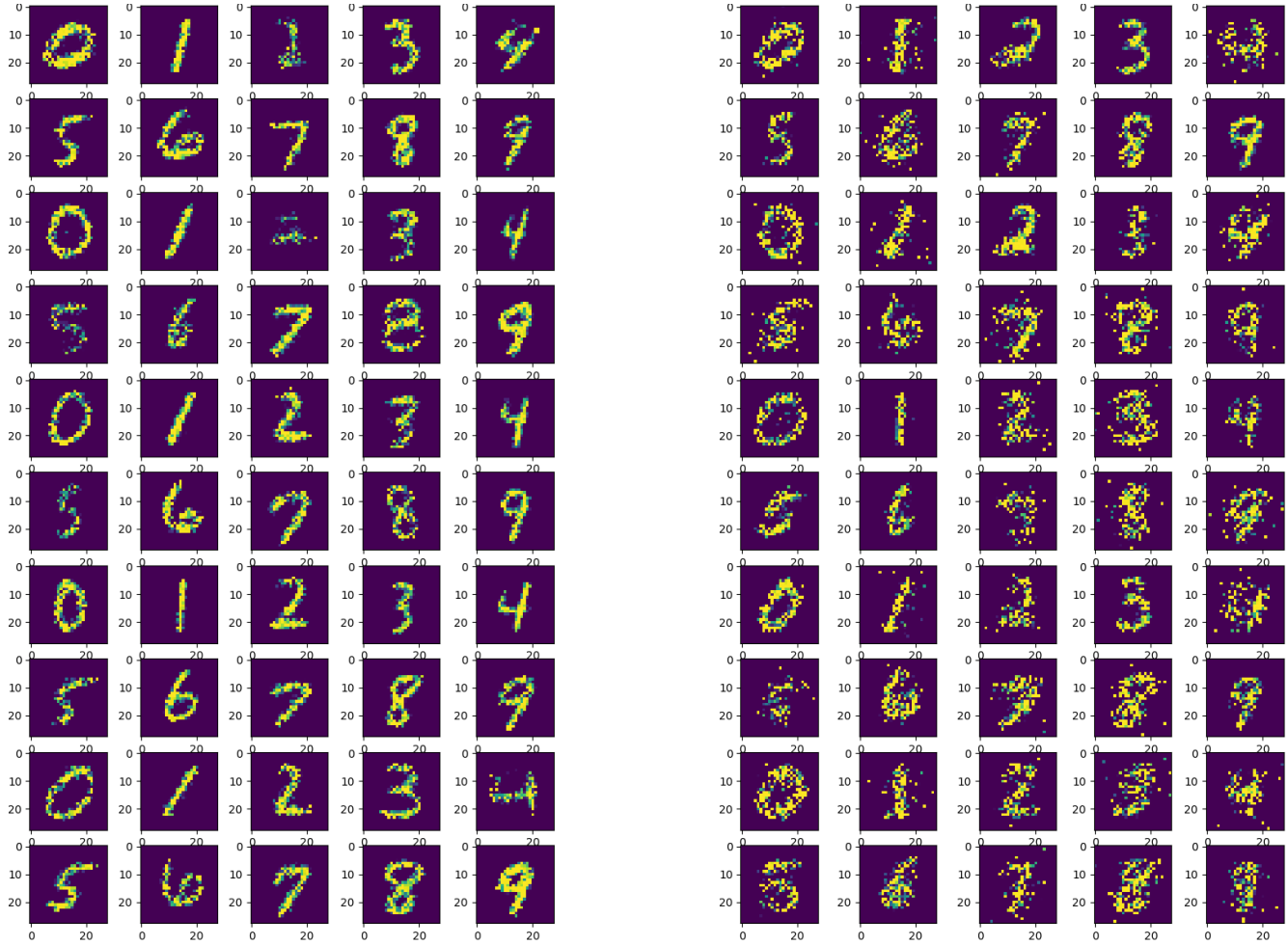

Figure 3: Example synthetic data points generated with ϵ and δ parameters (1,10^{-5}) in the left plot and (0.2,10^{-5}) in the right, on the 784-dimensional MNIST dataset. Models trained on this data and tested on held-out real images perform at 0.818 (at $\epsilon = 1$) and 0.650 (at $\epsilon = 0.2$) accuracy respectively, compared to 0.972 accuracy on the real dataset with different cluster sizes.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kenthapadi, K.; Korolova, A.; Mironov, I.; and Mishra, N. 2012. Privacy via the johnson-lindenstrauss transform. *arXiv preprint arXiv:1204.2606*.

Lee, K.; Kim, H.; Lee, K.; Suh, C.; and Ramchandran, K. 2019. Synthesizing differentially private datasets using random mixing. In *2019 IEEE International Symposium on Information Theory (ISIT)*, 542–546. IEEE.

Lee, K.; Lee, K.; Kim, H.; Suh, C.; and Ramchandran, K. 2018. SGD on Random Mixtures: Private Machine Learning under Data Breach Threats. *ICLR 2018 Workshop*.

McLachlan, G. J.; Lee, S. X.; and Rathnayake, S. I. 2019. Finite mixture models. *Annual review of statistics and its application*, 6: 355–378.

Mishra, N.; and Sandler, M. 2006. Privacy via pseudorandom sketches. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 143–152.

Mohammed, N.; Chen, R.; Fung, B. C.; and Yu, P. S. 2011. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 493–501.

Nguyen, T.; Chen, Z.; and Lee, J. 2020. Dataset Meta-Learning from Kernel Ridge-Regression. In *International Conference on Learning Representations*.

Nguyen, T.; Novak, R.; Xiao, L.; and Lee, J. 2021. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34.

Olsson, D. M.; and Nelson, L. S. 1975. The Nelder-Mead simplex procedure for function minimization. *Technometrics*, 17(1): 45–51.

Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; and Talwar, K. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*.

Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Erlingsson, Ú. 2018. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*.

Xie, L.; Lin, K.; Wang, S.; Wang, F.; and Zhou, J. 2018. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*.

Xu, C.; Ren, J.; Zhang, Y.; Qin, Z.; and Ren, K. 2017. DPPro: Differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security*, 12(12): 3081–3093.

Yoon, J.; Drumright, L. N.; and Van Der Schaar, M. 2020. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8): 2378–2388.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017a. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, J.; Cormode, G.; Procopiuc, C. M.; Srivastava, D.; and Xiao, X. 2017b. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4): 1–41.

Zhu, T.; Li, G.; Zhou, W.; and Philip, S. Y. 2017. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8): 1619–1638.