THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Computing 1, 13 Computing Drive, Singapore 117417

TRA3/19

Evaluation of Differentially Private Non-parametric Machine Learning as a Service

Ashish Dandekar, Debabrota Basu and Stéphane Bressan

March 2019

T e c h n i c a l R e p o r t

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

Mohan KANKANHALLI
Dean of School

Evaluation of Differentially Private Non-parametric Machine Learning as a Service

Ashish Dandekar[1,2], Debabrota Basu[2], and Stéphane Bressan[1,2]

[1] NUS-Singtel Cyber Security R & D Lab, Singapore
[2] School of Computing, National University of Singapore
ashishdandekar@u.nus.edu, steph@nus.edu.sg

Abstract. Machine learning algorithms create models from training data for the purpose of estimation, prediction and classification. While releasing parametric machine learning models requires the release of the parameters of the model, releasing non-parametric machine learning models requires the release of the training dataset along with the parameters. Indeed, estimation, prediction or classification with non-parametric models computes some form of correlation between new data and the training data. The release of the training dataset creates a risk of breach of privacy. An alternative to the release of the training dataset is the presentation of the non-parametric model as a service. Still, the non-parametric model as a service may leak information about the training dataset.
We study how to provide differential privacy guarantees for non-parametric models as a service. This cannot be achieved by perturbation of the output but requires perturbation of the model functions. We show how to apply the perturbation to the model functions of histogram, kernel density estimator, kernel SVM and Gaussian process regression in order to provide (ϵ, δ)-differential privacy. We evaluate the trade-off between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms on benchmarks and real-world datasets.
Our contribution is twofold. We show that functional perturbation is not only pragmatic for releasing machine learning models as a service but also yields higher effectiveness than output perturbation mechanisms for specified privacy parameters. We show the practical step to perturbate the model functions of histogram, kernel SVM, Gaussian process regression along with kernel density estimator. We evaluate the tradeoff between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms for a real-world dataset as well as a selection of benchmarks.

Keywords: Differential Privacy, Data Privacy, Non-parametric models, Functional Perturbation

1 Introduction

Organisations are amassing data at an unprecedented scale and granularity. They release either the raw data or the machine learning models that are trained on the

raw data. All machine learning models do not fit the choice of releasing only the models. A parametric machine learning model [21] assumes a parametric model function[3] that maps a new data to the corresponding output. A non-parametric machine learning model [21] does not assume a parametric model function but calculates some form of correlation between a new data and the training data to compute the corresponding output. For instance, kernel density estimation [23] computes the probability density of a new data by assimilating the probabilities of the new data originating from the probability distributions centred at every data-point in the training data. Kernel SVM [7] and Gaussian process regression [25] compute kernel Gram matrix between the new data and the training data. Thus, while releasing parametric machine learning models requires the release of the parameters of the model function, releasing non-parametric machine learning models requires the release of the training dataset along with the parameters. An alternative to the release of the training dataset is utilising non-parametric models as a service. While using a non-parametric model as a service, user would send a new data to the model to obtain the output of estimation, prediction, or classification.

Publication of raw data without any processing leads to a violation of the privacy of users [2]. Not only raw data but also publication of a 'non-private' machine learning model as a service leads to a violation of the privacy of users. For instance, experiments in [27] show that models created using popular machine-learning-as-a-service platforms, such as Google and Amazon, can leak identity of a data-point in the training dataset with accuracy up to 94%. In order to reduce the risk of breach of privacy, we need to take preemptive steps and provide quantifiable privacy guarantees for the released machine learning model. Differential privacy [13] is one of such privacy definitions to quantify the privacy guarantees.

We study how to provide differential privacy guarantees for non-parametric models as a service. This cannot be achieved by the output perturbation using typical differential privacy mechanisms, such as Laplace mechanism and Gaussian mechanism [13]. Due to sequential composition [13] of differential privacy, the privacy guarantee of a mechanism linearly degrades with the number of times the noise is added from a given noise distribution. Output perturbation requires addition of calibrated noise in the output for every new data input. Therefore, it suffers from the degradation of privacy guarantee. When machine learning is provided as a service one can not limit the number of queries. Once the noise is added to a model function, further evaluations are performed on the noisy model. We adopt the functional perturbation proposed in [17] in order to provide a robust privacy guarantee. Functional perturbation adds a scaled noise sampled from a Gaussian process to the function. [17] proves that an appropriate calibration of this mechanism provides (ϵ, δ)-differential privacy. We show how to calibrate the functional perturbation for histogram, kernel density estimator, kernel SVM, and Gaussian process regression. We evaluate the trade-off between

[3] Model function refers to the mapping from input to output that is learned by the corresponding machine learning algorithm.

the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms on benchmarks and US census dataset.

Our contribution is twofold. Firstly, we show that functional perturbation is a viable alternative to output perturbation to provide privacy guarantees for machine learning models as a service. We also hypothesise as well as experimentally validate that output perturbation is less effective than functional perturbation for a given privacy level and a given test set. Additionally, output perturbation is not directly applicable for machine learning models with categorical outputs, such as classification, where functional perturbation operates naturally. Secondly, we show the practical step to perturb the model functions of histogram, kernel SVM, Gaussian process regression, and the kernel density estimator. We evaluate the trade-off between the privacy guarantee and the error incurred for each of these non-parametric machine learning algorithms for US census dataset [1] and a comprehensive range of benchmarks. The results validate that the error decreases for nonparametric machine learning as a service with increase in the size of training dataset and privacy parameters ϵ and δ.

2 Background and related work

This paper is at the crossroads of machine learning and differential privacy. In this section, we provide a brief background and literature review of both of these fields and the research works bridging them.

2.1 Machine learning models

Murphy [21] classifies machine learning models into two classes based on the assumption on the kind of the mapping between inputs and outputs, called as the *hypothesis*. In this paper, we refer to the hypothesis as the model function.

A *parametric model* assumes a parametric function as the model function. Parameters of the model function are estimated using a given training dataset. Values of the parameters represent a summary of latent patterns in the data. Linear regression [21], logistic regression [21], K-means clustering [21] are a few examples of the parametric models. Unlike a parametric model, a *non-parametric model* assumes a set of correlated parametric functions, one for each data point in the training dataset, as the model function. In order to compute outputs for new data, it computes a function of correlation between the new data and the training dataset using the functions in the model function. Parameters of functions in the model function, called as the *hyperparameters*, along with the training data represent a summary of latent patterns in the data. Histogram fitting [21], kernel density estimation [29], Gaussian process [25], kernel SVM [29] are a few examples of the non-parametric models.

2.2 Differential privacy

Dwork et al. [12] propose differential privacy as a quantifiable privacy definition for any randomised algorithm. Degree of indistinguishability in the outputs obtained from a randomised algorithm operated on two *neighbouring datasets*[4]

[4] Two datasets are neighbouring if they differ at one data point.

quantifies the privacy guarantee of differential privacy. There has been extensive research on differential privacy and interested readers can refer to [13] for further details.

Let $\mathcal{D}$ denote the universe of datasets. Two datasets of equal cardinality x and y are said to be *neighbouring datasets* if they differ in one data point. We want to provide privacy guarantees for *queries* that are functions on $\mathcal{D}$. A *privacy-preserving mechanism*, which is a randomised algorithm, explicitly adds noise to the query from a given family of distributions. For example, Laplace and Gaussian mechanisms add noise sampled from Laplace and Gaussian distributions respectively [12]. For a given query f and parameters of a noise distribution Θ, we denote a privacy-preserving mechanism as $\mathcal{M}(f, \Theta)$. With this paraphernalia, we define (ϵ, δ)-differential privacy in Definition 1.

Definition 1 ((ϵ, δ)-differential privacy [13]). *A privacy-preserving mechanism $\mathcal{M}$, equipped with a query f and with parameters Θ, is (ϵ, δ)-differentially private if for all $Z \subseteq Range(\mathcal{M})$, $\epsilon \geq 0$, $\delta \geq 0$, and neighbouring datasets $x, y \in \mathcal{D}$:*

$$\mathbb{P}(\mathcal{M}(f, \Theta)(x) \in Z) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(f, \Theta)(y) \in Z) + \delta$$

In Definition 1, ϵ quantifies the privacy guarantee and δ quantifies a slack in the inequality. In order to have stronger privacy guarantees, we require a small value of ϵ and close to zero value of δ. For $\delta = 0$, (ϵ, δ)-differential privacy reduces to the ϵ-differential privacy [12].

2.3 Related work

Most of the big technology companies offer machine learning as a service on their cloud platforms, such as Google's Cloud Machine Learning Engine[5], Microsoft's Azure Learning Studio[6], Amazon's Machine Learning on AWS[7], IBM's Bluemix[8]. These apps provide machine learning models as easy to use APIs for data scientists. Cloud services also provide storage space to host training datasets. For an extensive survey of such platforms, readers can refer to [24].

Privacy of machine learning models is a well-studied topic. Ateniese et al. [5] show the ability to learn statistical information about the training data through parameters of a trained machine learning model. They show a successful attack on support vector machine and hidden Markov model. Homer et al. [18] identify the presence of a certain genome in a publicly released highly complex genomic mixture microarray dataset. They do so by comparing distributions of genomes from the released sample to avaliable statistics of the population. Fredrikson et al. [15] propose the model inversion attack on machine learning models wherein they learn some sensitive attribute in the training dataset. Given black-box access to the model and access to demographic information about patients, they successfully learn genomic markers of patients. In the follow-up work, Fredrikson

[5] https://cloud.google.com/ml-engine/
[6] https://azure.microsoft.com/en-us/services/machine-learning-studio/
[7] https://aws.amazon.com/machine-learning/
[8] https://www.ibm.com/cloud/

et al. [16] show instantiation of a successful model inversion attack on decision trees and neural networks that are implemented on machine learning as a service platform. Shokri et al. [27] propose membership inference attack that infers the presence of a data-point in the training dataset based on the outputs machine learning models. They perform attacks on classification models provided by commercial platforms from Google and Amazon. We have enlisted the attacks that are pertinent to research in this work. For an extensive survey of attacks on various machine learning models, readers can refer to [14].

Differential privacy [12] has become a popular privacy definition to provide privacy guarantees for machine learning algorithms. Researchers have devised privacy-preserving mechanisms to provide differential privacy guarantees for linear regression [9, 31], logistic regression [8, 30], support vector machines [26], deep learning [3, 27]. Chaudhury et al. [9] propose differentially private empirical risk minimisation, which lies at the heart of training of machine learning models. They propose output perturbation and objective perturbation. These mechanisms are helpful for releasing parametric machine learning models. Zhang et al. [31] propose the functional mechanism that introduces noise in the loss function of a machine learning model. The functional mechanism is useful for parametric machine learning models that estimate parameters of the model by minimising its loss function. Hall et al. [17] propose the use of functional perturbation that induced noise in the coefficient of expansion of a function in a functional basis. Functions of non-parametric models that use kernels lie in the RKHS spanned by the kernel. Therefore, it is possible to apply the functional perturbation to provide privacy guarantees for non-parametric models. Smith et al. [28] apply functional perturbation by Hall et al. to provide differntial privacy Gaussian processe regression. Aldá and Rubinstein [4] propose Bernstein mechanism that provides a differentially private way of releasing functions of machine learning models in a non-interactive way. Balog et al. [6] provide a functional perturbation that ensures the closure of functions in a finite dimensional RKHS under the appropriate perturbations. Nozari et al. [22] propose a functional perturbation algorithm that is catered to distributed machine learning task.

Jain and Thakurta [20] propose three ways, which are interactive, non-interactive and semi-interactive, of using machine learning models with differential privacy guarantees. This work is an instance of interactive use of non-parametric machine learning model wherein we provide differential privacy guarantees using the functional perturbation proposed in the work of Hall et al. [17].

3 Methodology

In this section, we discuss release of trained machine learning models. We argue that non-parametric machine learning models need to be released as a service to users. We further instantiate functional perturbation by Hall et al. [17], which provides (ϵ, δ)-differential privacy guarantee, to four non-parametric models.

3.1 Non-parametric machine learning models as a service

Jain and Thakurta [20] propose three ways in which an organisation can use machine learning models. Firstly, they propose a non-interactive model release

wherein an organisation releases a model with quantifiable privacy guarantees. Non-interactive model release is plausible for parametric machine learning models since values of the parameters are sufficient to compute outputs for a new data. Non-parametric machine learning models require training dataset along with the parameters to compute outputs for new data. Secondly, they propose a semi-interactive model release wherein an organisation releases model that provides quantifiable privacy guarantees for a specified set of test data. A priori knowledge of test data is not an assumption that can be realised in every business scenario. Lastly, they propose interactive model release wherein an organisation provides machine learning model as a service. It keeps trained model on the server and users send queries to the server. For non-parametric models, release of training dataset violates the privacy of the users. Therefore, interactive model release, i.e. release of machine learning as a service is a viable alternative.

Differential privacy, as defined in Definition 1, is a privacy definition for randomised algorithms. In order to provide quantifiable differential privacy guarantees, we need to introduce randomisation while using machine learning models as a service. A privacy-preserving mechanism introduces randomisation to avoid the release of true outputs. Under appropriately calibrated randomisation, privacy-preserving mechanisms provide differential privacy guarantees.

Firstly, randomisation can be introduced by adding an appropriately calibrated random noise to the output of the query. These privacy-preserving mechanisms are called as *output perturbation mechanisms*. For instance, Laplace mechanism [13] adds noise drawn from Laplace distribution whereas Gaussian mechanism [13] adds noise drawn from Gaussian distribution to the output of the model. Multiple evaluations of such mechanisms result in a sequential composition [13]. Privacy guarantee of the sequential composition of privacy-preserving mechanisms linearly degrades with the number of evaluations of privacy preserving mechanisms. Secondly, randomisation can be introduced by adding an appropriately calibrated random noise to the model function. Unlike output perturbation mechanisms, which add calibrated noise to every output of the query, the privacy-preserving mechanisms that perturb functions are *one-shot* privacy-preserving mechanisms. They add calibrated noise in a function leading to change in its functional form. The noisy functional form is used for computing outputs. Therefore, functional perturbation does not suffer from the degradation in the differential privacy guarantee with increasing the number of queries.

When a machine learning model is provided as a service, one cannot strictly control the number of times a user accesses the service. Therefore, we choose functional perturbation based privacy-preserving mechanism. Hall et al.[17] propose the functional mechanism that adds calibrated noise to the expansion of the model function in an appropriate functional basis. Functions of non-parametric machine learning models, especially the ones that use kernels, lie in Reproducing Kernel Hilbert Space (RKHS) [29] associated with the kernel. Thus, RKHS readily provides a functional basis for functions of non-parametric models. Zhang et al. [31] propose the functional mechanism that adds calibrated noise to the loss function of machine learning model. Loss functions are akin to parametric

models that train their parameters using some appropriate loss function. Therefore, we choose to use functional perturbation as proposed by Hall et al. [17] to provide differential privacy guarantees for non-parametric models released as a service.

3.2 Functional perturbation in RKHS

Hall et al. [17] propose a mechanism that provides a calibrated functional perturbation that provides quantifiable (ϵ, δ)-differential privacy guarantee. We briefly explain functional perturbation of a function that lies in a reproducing kernel Hilbert space (RKHS).

Suppose that a function $f : \mathbb{R}^d \to \mathbb{R}$ lies in RKHS, $\mathcal{H}_k$, associated with a kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. For a given dataset $D = \{x_i\}_{i=1}^n$ where each $x_i \in \mathbb{R}^d$, let $\{k(\cdot, x_i)\}_{i=1}^n$ denotes a basis of $\mathcal{H}_k$. In this basis, any function $f \in \mathcal{H}_k$ is expanded as:

$$f(\cdot) = \sum_{i=1}^{n} w_i^f k(\cdot, x_i)$$

where each $w_i^f \in \mathbb{R}$. Inner product between two functions $f, g \in \mathcal{H}_k$ is defined as:

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i^f w_j^g k(x_i, x_j)$$

The inner product is used to define norm of any function in $\mathcal{H}_k$ as $\|f\|_{\mathcal{H}_k} = \sqrt{\langle f, f \rangle}$.

Functional perturbation adds calibrated noise sampled from a Gaussian process to a model function. Gaussian process uses the kernel that is associated with RKHS where the function lies. We formally define functional perturbation in Definition 2.

Definition 2 (Functional perturbation [17]). *Let f_D denotes a model function, whose parameters (or hyperparameters) are estimated on a dataset $D \in \mathcal{D}$. Assume that f_D lies in a reproducing kernel Hilbert space, $\mathcal{H}_k$, with an associated kernel k. Functional perturbation is a privacy-preserving mechanism that perturbs f_D as follows:*

$$f'_D = f_D + \Delta \frac{c(\delta)}{\epsilon} G. \tag{1}$$

where G is a sample path of a Gaussian process with mean zero and covariance function k and $\Delta, \epsilon, \delta > 0$.

Functional perturbation in Definition 2 satisfies (ϵ, δ)-differential privacy when parameters are calibrated as $\Delta \geq \max_{D,D'} \|f_D - f_{D'}\|_{\mathcal{H}_k}$ and $c(\delta) \geq \sqrt{2 \log \frac{2}{\delta}}$. Δ is *sensitivity* of the functions in RKHS $\mathcal{H}_k$. Sensitivity is the maximum deviation of model functions that are trained on any two neighbouring datasets D and D'. In order to apply functional perturbation for machine learning tasks, we need to compute the sensitivity of respective model functions.

3.3 Applications to four non-parametric machine learning models

We now illustrate application of functional perturbation for four non-parametric machine learning models. We use non-parametric models that are based on kernel methods. For such non-parametric models, model functions lie in the RKHS associated with the specified kernel.

Histogram. Histogram [21] is used for solving discretised probability density estimation problem. It discretises the domain of a given dataset into a finite number of *bins*. Each bin defines an interval in the domain of the training dataset. Probability of a data-point inside an interval is commensurate to the number of training data-points that lie in the interval.

For a fixed number of bins b, histogram is a vector in $\mathbb{R}^b$. Therefore, we can consider histogram estimation as a function $f : \mathcal{D} \to \mathbb{R}^b$ wherein $\mathcal{D}$ is a universe of datasets. Let, $\{e_i\}_{i=1}^b$ be the standard basis of Euclidean space $\mathbb{R}^b$. Standard basis spans RKHS associated with the dot product kernel, i.e. $k(x, y) = x^T y$. For a pair of neighbouring datasets, L_1 norm between two histograms is two in the case when the distinct data-points occupy two different bins. Therefore, sensitivity of the histogram function is 2. Let, f_D denotes the histogram for a dataset D with the number of bins b. Thus, the functional perturbation of Equation 1 for histograms takes the form

$$f'_D = f_D + \frac{2}{n}\frac{c(\delta)}{\epsilon}G.$$

Kernel density estimation. Kernel density estimation [21] is a probability density estimation problem that estimates the probability density function of a training dataset. It assumes a probability density function centred at every data-point in the training dataset. Probability of a new data-point is computed as weighted average of the probabilities computed using the probability densities centred at every data-point.

We consider the kernel function namely Gaussian kernel that outputs values in the range $[0, 1]$. It acts as a probability density function. Let k denotes a Gaussian kernel with bandwidth h. Estimate of the probability density function for a dataset $D = \{x_i\}_{i=1}^n$ for the Gaussian kernel k is presented as

$$f_D(\cdot) = \frac{1}{n}\sum_{x_i \in D} k(\cdot, x_i) = \frac{1}{n}\sum_{x_i \in D} \frac{1}{(2\pi h^2)^d} \exp\left(-\frac{\langle \cdot, x_i \rangle}{2h^2}\right).$$

Hall et al. [17] compute the sensitivity Δ of kernel density estimator with a Gaussian kernel as $\frac{\sqrt{2}}{n(2\pi h^2)^{d/2}}$. Thus, from Equation 1 the functional perturbation for kernel density estimate with Gaussian kernel is

$$f'_D = f_D + \left(\frac{\sqrt{2}}{n(2\pi h^2)^{d/2}}\right)\frac{c(\delta)}{\epsilon}G.$$

Gaussian process regression. Gaussian process [25] is a collection of Gaussian random variables such that any subset follows a multivariate Gaussian distribution. Covariance function for the multivariate Gaussian distribution

is calculated using a kernel function k. Gaussian process regression outputs a response sampled from posterior distribution of a test data-point given the training dataset. Mean function $\bar{f_D}$ and variance function $Var(f_D)$ of the posterior distribution computed on a training dataset D are given in Equation 2.

$$\begin{aligned} \bar{f_D}(\cdot) &= \sum_{d_i \in D} \sum_{d_j \in D} (K_D + \sigma_n^2 \mathbb{I})^{-1}_{ij} y_j k(\cdot, x_i) \\ Var(f_D)(\cdot) &= k(\cdot, \cdot) - \sum_{d_i \in D} \sum_{d_j \in D} (K_D + \sigma_n^2 \mathbb{I})^{-1}_{ij} k(\cdot, x_i) k(\cdot, x_j) \end{aligned} \tag{2}$$

K_D is the Gram matrix computed using kernel k on the training dataset and d is the dimension of each training data-point.

Smith et al. [28] use the functional perturbation to provide differential privacy guarantee to Gaussian process regression. Equation 2 shows that the posterior covariance function does not require responses y_j's in the training data. Since only the responses are sensitive towards the disclosure, Smith et al. [28] proposed to perturb only the posterior mean function. Since the sensitivity of the posterior mean function with Gram matrix K_D is $d\|(K_D + \sigma_n^2 \mathbb{I})^{-1}\|_\infty$, they apply the functional perturbation to the posterior mean function as

$$\bar{f_D}' = \bar{f_D} + (d\|(K_D + \sigma_n^2 \mathbb{I})^{-1}\|_\infty) \frac{c(\delta)}{\epsilon} G.$$

Kernel support vector machine. Support vector machine (SVM) [10] is used for solving a classification problem. SVM outputs the class label of a data-point that is specified as the input. Linear SVM is a parametric machine learning model whereas kernel SVM is a non-parametric machine learning model.

Let us consider a data-point $d = (x, y)$ where $x \in \mathbb{R}^d$ are the predictors and $y \in \{-1, 1\}$ is the associated class label. Let $\mathcal{D}$ denotes universe of datasets with n data-points each. We fit a support vector machine classifier with a kernel k on a training dataset $D \in \mathcal{D}$ with n data-points. Kernel support vector machine assumes the form $f(\cdot) = \langle w, \phi(\cdot) \rangle$ where $w \in \mathbb{R}^F$ and $\phi : \mathbb{R}^d \to \mathbb{R}^F$. w is estimated by solving the optimisation problem in Equation 3. In Equation 3, C denotes the regularisation constant and l denotes the loss function.

$$\max_{w \in \mathbb{R}^F} \frac{\|w\|^2}{2} + C \sum_{d \in D} l(y_i, f_D(x_i)) \tag{3}$$

Using *hinge loss*, $l_{hinge}(x, y) = max(0, xy)$, as the loss function we obtain a closed form solution. It is presented in Equation 4. In the solution, α^*'s are called support vectors that are solutions to *dual* of the optimisation problem in Equation 3.

$$w_D = \sum_{i=1}^{n} \alpha_i^* y_i k(\cdot, x_i) \tag{4}$$

Hall et al. [17] compute the sensitivity of the minimisers of regularised functionals in RKHS. Equation 3 represents an instance of the same problem. Since the

sensitivity of w_D is $\frac{2C}{n}$, following Equation 1 the functional perturbation for kernel SVM takes the form

$$w'_D = w_D + \left(\frac{2C}{n}\right)\frac{c(\delta)}{\epsilon}G.$$

4 Performance Evaluation

In this section, we present effectiveness and efficiency evaluation of functional perturbation for four non-parametric models, *viz.* histogram, kernel density estimation (KDE), Gaussian process regression (GP regression) and kernel support vector machine (kernel SVM), as a service. We comparatively evaluate output perturbation and functional perturbation mechanism. We observe that output perturbation mechanism are less effective than functional perturbation mechanism for a specified setting of differential privacy parameters.

4.1 Dataset

Real world dataset. We conduct experiments on a subset of the 2000 US census dataset provided by Minnesota Population Center in its Integrated Public Use Microdata Series [1]. The census dataset consists of 1% sample of the original census data. It spans over 1.23 million households with records of 2.8 million people. The value of several attributes is not necessarily available for every household. We have therefore selected 212, 605 records, corresponding to the household heads, and 6 attributes, namely, *Age, Gender, Race, Marital Status, Education, Income*. We treat this dataset as the population from which we draw samples of desired sizes.

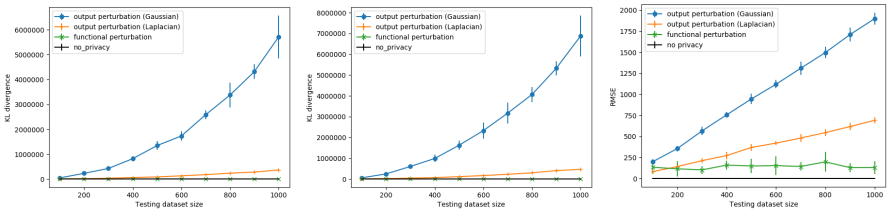
Benchmark datasets. For histogram and kernel density estimation, we follow Hall et al. [17] and synthetically generate a dataset from a known probability distribution. We generate 5000 points from a Gaussian distribution with mean and variance of 2 and 1.3 respectively. For Gaussian process regression, we follow Smith [28] and use !Kung San woman demographic dataset [19]. It comprises of heights and ages of 287 women. For kernel SVM, we use Iris dataset [11]. It comprises of three species of Iris flower with four attributes: length and width of sepal and petal.

4.2 Experimental Setup

All experiments are run on Linux machine with 12-core 3.60GHz Intel® Core i7™processor with 64GB memory. Python® 2.7.6 is used as the scripting language. We use RBF kernel for the experiments. Hyperparameters of the kernel are tuned by performing cross-validation on respective dataset.

4.3 Evaluation Metrics

We perform experiments on four non-parametric models solving the problems of estimation, prediction, and classification. Therefore, we use different metrics of effectiveness for the evaluation. Histogram and kernel density estimation are used for estimating probability density of a given data-point and we use Kullback-Leibler divergence (KL divergence) as the metric of effectiveness. Gaussian process regression is used for predicting real-valued attribute and we use root mean



(a) Histogram (b) KDE (c) GP regression

Fig. 1. Comparative evaluation of functional and output perturbation mechanisms for varying size of test datasets. We compare (0.4, 0.001)-differentially private functional perturbation, (0.4, 0.001)-differentially private Gaussian mechanism and (0.4, 0.0)-differentially private Laplace mechanism.

squared error (RMSE) as the metric of effectiveness. Kernel SVM is used for classification and we use classification error as the metric of effectiveness. Smaller the value of any of these metrics higher is the effectiveness of the model. In order to evaluate efficiency, we compute query execution time, i.e. the time required to compute output of the model.

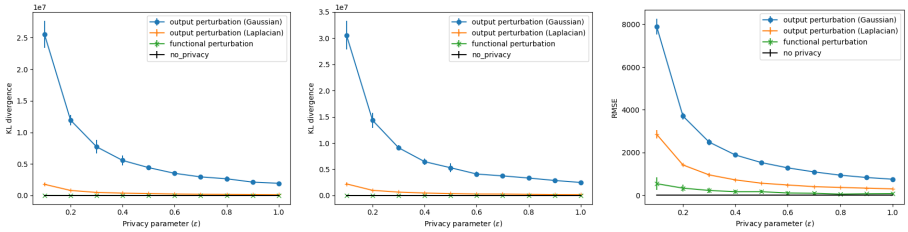
4.4 Effectiveness Evaluation

In this section, we present the results on the real-world census dataset.

We start by the comparative evaluation of the functional perturbation and output perturbation mechanisms, namely Gaussian mechanism and Laplace mechanism. Output perturbation mechanisms are not directly applicable for machine learning models with categorical outputs, such as SVMs. Therefore, we perform comparative study for histograms, KDE and GP regression. We also plot the effectiveness of the model without any application of privacy-preserving mechanism. We denote it by "no privacy". In case of histogram and KDE, we do not have the true distributions of the attributes from the census dataset. Therefore, we compute effectiveness by computing KL divergence between functionally perturbed estimators and their non-private counterparts.

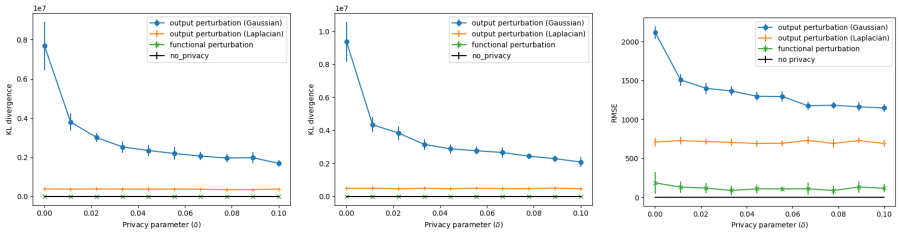
In Figure 1, we comparatively evaluate effectiveness for varying size of testing datasets. Across three models, we observe that effectiveness of the output perturbation mechanisms degrades as the testing dataset size increases. We do not observe such a phenomenon with the functional perturbation. Due to sequential composition [13], privacy guarantee of output perturbation mechanisms linearly degrades with the number of evaluations. In order to attain differential privacy with specified privacy parameters, output perturbation mechanisms introduce higher amount of noise for testing datasets of large sizes. Higher amount of noise results in reduction in the effectiveness.

In Figures 2 and 3, we comparatively evaluate effectiveness for varying privacy parameters ϵ and δ respectively. Across three models, we observe that the effectiveness of the output perturbation mechanisms increases as values of privacy parameters increase. Privacy parameter ϵ quantifies the privacy guarantee of differential privacy. Higher values of ϵ provides weaker privacy guarantees. Weaker privacy guarantees require less amount of noise and hence yield higher



(a) Histogram (b) KDE (c) GP regression

Fig. 2. Comparative evaluation of functional and output perturbation mechanisms for varying privacy parameter ϵ and $\delta = 0.001$. We use dataset of size 5000 to train the models.



(a) Histogram (b) KDE (c) GP regression

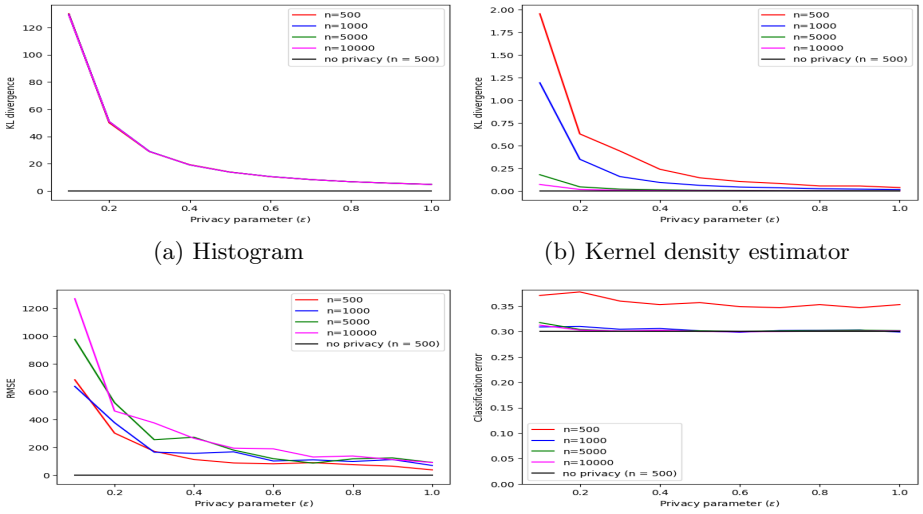
Fig. 3. Comparative evaluation of functional and output perturbation mechanisms for varying privacy parameter δ and $\epsilon = 0.4$. We use dataset of size 5000 to train the models.

effectiveness. Privacy parameter δ is a quantifier of the extent of slack provided in the privacy guarantee of ϵ-differential privacy. In order to provide a robust differential privacy guarantee, we require the value of δ to be as small as possible. Thus, with increasing value of δ the amount of perturbation in the function reduces and hence, the effectiveness increases.

We continue our evaluation of functional perturbation for four non-parametric models on the census dataset. In Figure 4, we present the effectiveness as privacy parameter ϵ varies between 0 to 1 keeping $\delta = 0.0001$ for different sizes of training dataset sizes. We observe that effectiveness of the models increases with increasing the size of dataset. The reason for this is twofold. Firstly, effectiveness of non-parametric models increases with increasing size of the training dataset [21]. Secondly, closer inspection of equations of functional perturbation for each of the four models tells that the amount of noise is inversely proportional to the number of training data-points. Thus, the functional perturbation adds lesser amount of noise for specified privacy parameters as the size of training dataset increases. We make similar observations while evaluating the effectiveness under variation in privacy parameter δ for a fixed value of ϵ. Due to lack of space, we do not provide these results.

4.5 Efficiency evaluation

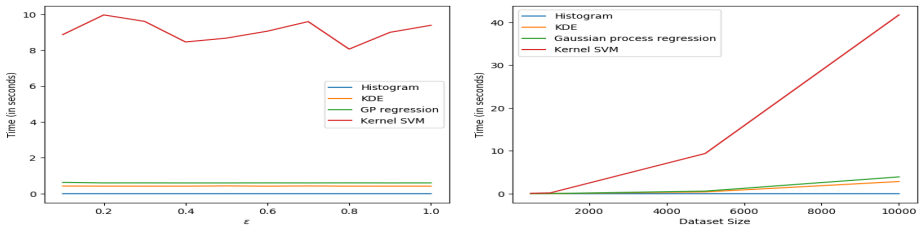
In Figure 5(a), we plot the query execution time that is the time required to compute the output for non-parametric models as a service, on a dataset of size 5000 with varying privacy levels. For a given non-parametric model, we observe that query execution time does not depend on the value of the privacy parameter



(a) Histogram (b) Kernel density estimator

(c) Gaussian process regression (d) Kernel SVM

Fig. 4. Variation in the utility as the privacy parameter ϵ changes for datasets of varying sizes. Experiments are carried out with $\delta = 0.0001$ on census dataset.

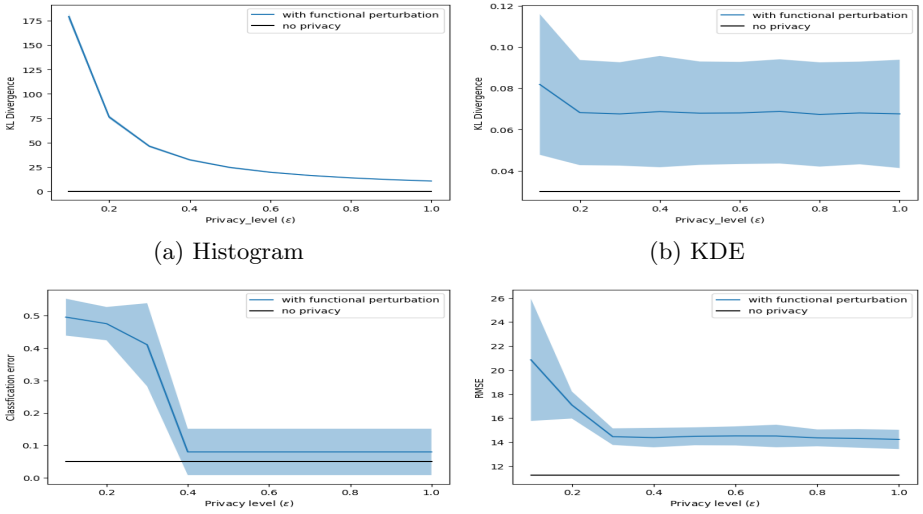


(a) Privacy parameter (b) Training dataset size

Fig. 5. Evaluation of efficiency of functional perturbation for various four non-parametric machine learning models. Figure (a) plots query execution time versus privacy level. Figure (b) plots query execution time versus training dataset size. For both experiments, we set $\delta = 0.001$. We set $\epsilon = 0.2$ for the plot in Figure (b).

ϵ. Functional perturbation involves sampling a path from the Gaussian process with zero mean function and covariance function computed using the kernel function used in the non-parametric model. The computation of covariance functions requires a significant computational time. This computation time is not affected by any particular value of privacy level. We make similar observation for the privacy parameter δ, which we do not include in the paper due to lack of space.

In Figure 5(b), we plot query evaluation time for varying size of the training datasets. For this experiment, we set privacy parameters ϵ and δ to 0.2 and 0.001 respectively. We observe that evaluation time increases with increasing size of the training dataset. Large training datasets require large amount of correlations to



(a) Histogram (b) KDE

(c) Gaussian process regression (d) Kernel SVM

Fig. 6. Variation in the utility as the privacy level changes for datasets of varying sizes. Experiments are carried out with $\delta = 0.0001$ on benchmark datasets.

be computed for every new data-point. Therefore, larger training datasets incur higher amount of time.

4.6 Experiments on the Benchmark Datasets

For reproducibility of the results, we also conduct experiments on the datasets that are either synthetic or publicly available. We observe results that are consistent with the results on the real-world dataset.

In Figure 6, we present effectiveness of functional perturbation technique on the *benchmark datasets*. We perform 10 experimental runs for each value of the privacy level. Solid lines in Figure 4 show mean effectiveness whereas shaded region covers values that are one standard deviation away from the mean. We invariably observe that effectiveness of the models increases when we increase the privacy level in the functional perturbation. Our observation for the other experiments on the benchmark datasets are consistent with the observations that we make for the same experiment on the census datasets.

5 Conclusion

We show that functional perturbation is not only pragmatic for releasing machine learning models as a service but also yields higher effectiveness than output perturbation mechanisms for specified privacy parameters. We show how to apply functional perturbation to the model functions of histogram, kernel density estimator, kernel SVM and Gaussian process regression in order to provide (ϵ, δ)-differential privacy. We evaluate the tradeoff between the privacy guarantee and

the error incurred for each of these non-parametric machine learning algorithms for a real-world dataset as well as a selection of benchmarks.

We are now studying functional perturbation for non-parametric machine learning methods such as k-nearest neighbour density estimation and kernel Bayesian optimisation. We are also interested in studying a step by step functional perturbation method that perturbs a model function in adaptive way balancing the specified privacy and utility requirements.

Acknowledgement

This project is supported by the National Research Foundation, Singapore Prime Minister's Office under its Corporate Laboratory@University Scheme between National University of Singapore and Singapore Telecommunications Ltd.

References

1. Minnesota population center. integrated public use microdata series international: Version 5.0. https://international.ipums.org. (2009)
2. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)(text with eea relevance). Official Journal of the European Union **L**(119), 1–88 (2016), https://eur-lex.europa.eu/eli/reg/2016/679/oj
3. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318. ACM (2016)
4. Ald, F., Rubinstein, B.: The bernstein mechanism: Function release under differential privacy (2017)
5. Ateniese, G., Mancini, L.V., Spognardi, A., Villani, A., Vitali, D., Felici, G.: Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. International Journal of Security and Networks **10**(3), 137–150 (2015)
6. Balog, M., Tolstikhin, I., Schölkopf, B.: Differentially private database release via kernel mean embeddings. arXiv preprint arXiv:1710.01641 (2017)
7. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 144–152. ACM (1992)
8. Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. In: Advances in Neural Information Processing Systems. pp. 289–296 (2009)
9. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. Journal of Machine Learning Research **12**(Mar), 1069–1109 (2011)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
11. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml

12. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming. pp. 1–12. Springer Berlin Heidelberg (2006)
13. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
14. Dwork, C., Smith, A., Steinke, T., Ullman, J.: Exposed! a survey of attacks on private data. Annual Review of Statistics and Its Application **4**, 61–84 (2017)
15. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In: Proceedings of the USENIX Security Symposium. UNIX Security Symposium. vol. 2014, pp. 17–32. NIH Public Access (2014)
16. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1322–1333. ACM (2015)
17. Hall, R., Rinaldo, A., Wasserman, L.: Differential privacy for functions and functional data. Journal of Machine Learning Research **14**(Feb), 703–727 (2013)
18. Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J.V., Stephan, D.A., Nelson, S.F., Craig, D.W.: Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. PLoS genetics **4**(8), e1000167 (2008)
19. Howell, N.: Data from a partial census of the !kung san, dobe. (1967), https://public.tableau. com/profile/john.marriott#!/vizhome/kung-san/Attributes
20. Jain, P., Thakurta, A.: Differentially private learning with kernels. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 118–126. PMLR (2013)
21. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. The MIT Press (2012)
22. Nozari, E., Tallapragada, P., Cortés, J.: Differentially private distributed convex optimization via functional perturbation. IEEE Transactions on Control of Network Systems **5**(1), 395–408 (2018)
23. Parzen, E.: On estimation of a probability density function and mode. The annals of mathematical statistics **33**(3), 1065–1076 (1962)
24. Pop, D.: Machine learning and cloud computing: Survey of distributed and saas solutions. arXiv preprint arXiv:1603.08767 (2016)
25. Rasmussen, C.E.: Gaussian processes in machine learning. In: Advanced lectures on machine learning, pp. 63–71. Springer (2004)
26. Rubinstein, B.I., Bartlett, P.L., Huang, L., Taft, N.: Learning in a large function space: Privacy-preserving mechanisms for svm learning. Journal of Privacy and Confidentiality **4**(1) (2012)
27. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: Security and Privacy (SP), 2017 IEEE Symposium on. pp. 3–18. IEEE (2017)
28. Smith, M.T., Zwiessele, M., Lawrence, N.D.: Differentially private gaussian processes. arXiv preprint arXiv:1606.00720 (2016)
29. Smola, A.J., Schölkopf, B.: Learning with kernels, vol. 4. Citeseer (1998)
30. Yu, F., Rybar, M., Uhler, C., Fienberg, S.E.: Differentially-private logistic regression for detecting multiple-snp association in gwas databases. In: International Conference on Privacy in Statistical Databases. pp. 170–184. Springer (2014)

31. Zhang, J., Zhang, Z., Xiao, X., Yang, Y., Winslett, M.: Functional mechanism: regression analysis under differential privacy. Proceedings of the VLDB Endowment **5**(11), 1364–1375 (2012)