Differentially Private Regression Diagnostics

Yan Chen Ashwin Machanavajjhala
Dept. of Computer Science
Duke University, USA
{yanchen,ashwin}@cs.duke.edu

Jerome P. Reiter Andrés F. Barrientos
Dept. of Statistical Science
Duke University, USA
{jerry,afb26}@stat.duke.edu

ABSTRACT

Linear and logistic regression are popular statistical techniques for analyzing multi-variate data. Typically, analysts do not simply posit a particular form of the regression model, estimate its parameters, and use the results for inference or prediction. Instead, they first use a variety of diagnostic techniques to assess how well the model fits the relationships in the data and how well it can be expected to predict outcomes for out-of-sample records, revising the model as necessary to improve fit and predictive power. Often, however, analysts cannot access the data directly due to privacy concerns, making it impossible to use standard regression estimation and diagnostic procedures. Recently, a number of algorithms have been proposed that perform linear and logistic regression while ensuring privacy. However, the outputs of these algorithms alone do not allow analysts to perform regression diagnostics. This severely limits the usability of the existing private regression algorithms, as the outputs may well be giving noisy estimates of parameters for an underlying model that completely fails to describe the relationships in the data. In this article, we begin to fill this gap in privacy preserving data analysis by developing ϵ-differentially private diagnostics for regression. Specifically, we create differentially private versions of residual plots for linear regression and of receiver operating characteristic (ROC) curves for logistic regression. The former helps determine whether or not the data satisfy the assumptions underlying the linear regression model, and the latter is used to assess the predictive power of the logistic regression model. Our empirical studies show that these algorithms are adequate for diagnosing the fit and predictive power of regression models as long as the size of the dataset times the privacy parameter (ϵ) is not less than 1000.

1. INTRODUCTION

The increasing digitization of personal information in the form of medical records, administrative & financial records, social networks, etc., has helped catalyze scientific inquiry in a number of domains. However, this digitization also has increased concerns over ensuring the confidentiality of the individuals from whom such data are collected. A wealth of literature now shows that anonymization techniques that redact identifiers or coarsen attributes as well as systems that allow users to query the database indiscriminately [7], do not prevent determined adversaries from being able to learn sensitive properties of individuals.

In response to this realization, researchers and data stewards have developed strategies that limit how analysts can access the data, with the goal of protecting confidentiality. One approach is to require analysts to use algorithms that result in noisy answers to queries of the confidential data, with the noise specified so attackers cannot infer sensitive information from the algorithm outputs. For example, algorithms can be designed to satisfy ϵ-differential privacy [5], a popular privacy constraint that ensures that the presence or absence of any one record does not significantly alter the output of the algorithm. Another approach, taken by some government statistical agencies, is to release one-off *synthetic*, i.e., entirely simulated, databases, so that the simulated records cannot be meaningfully matched to genuine records [22, 19].

In this work, we consider contexts where analysts, working under privacy preserving access constraints, seek to explain or predict some outcome y from a multi-variate set of variables $\mathbf{x}$. Absent privacy constraints, the go-to tool for this task is regression modeling. For continuous outcomes, the most popular model is *linear regression*, in which we assume that $y = \beta_0 + \boldsymbol{\beta} \cdot \mathbf{x} + \xi$, where the βs are called *coefficients*, and ξ is the *error* that is assumed to be normally distributed with 0 mean and some unknown fixed variance σ^2. For binary outcomes ($y \in \{0, 1\}$), the most popular model is *logistic regression*, in which we assume that y, given $\mathbf{x}$, has a Bernoulli distribution with probability $P(y = 1 \mid \mathbf{x})$ where $\log[P(y = 1 \mid \mathbf{x})/(1 - P(y = 1 \mid \mathbf{x}))] = \beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}$. We note that one can use non-linear functions of $\mathbf{x}$ in the predictor function in either model.

Regression is used for two primary tasks – *explain* associations between the outcome and the explanatory variable and to use the model to *predict* the outcome on unseen data. The former is especially important in public health, epidemiology, and the social sciences, where the scientific questions typically focus on whether specific variables are important (e.g., statistically significant) or unimportant predictors of the outcome. For example, in a linear regression seeking to explain salaries from demographic variables, a social scientist might be interested in whether or not the coefficient (β) for an explanatory variable indicating "sex is female" is statistically significant, given the other variables in the model. The validity of such inferences depends critically on the reasonableness of the assumptions underpinning the model. For example, if the analyst specifies a linear relationship between salary and age (in years) in the model, but the true generative relationship is quadratic (as it generally is), it is pointless to interpret the inferences for the coefficient for age from the ordinary least squares fit. The fitted

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

model completely misrepresents the association. Thus, in addition to estimating the parameters of the model, an analyst must be able to evaluate how well the posited model fits the theoretical assumptions underpinning regression analyses for the data at hand. Similarly, when regression is used for prediction, the analyst must be able to assess the extent to which the model can predict outcomes accurately for unseen data (e.g., out of sample records).

Absent privacy constraints, analysts have a variety of tools to diagnose the fit and predictive power of regression models. Model fit for linear regression is typically assessed by examining the distribution of *residuals*, which are observed values minus predicted values (defined formally in Section 4). To evaluate predictive power for logistic regression, a popular tool is the receiver operating charateristic (ROC) curve that plots the true positive rate against the false positive rate of predictions on unseen data. The area under the ROC curve (or AUC, a value in [0,1]) is typically used to compare models.

Assessing model fit and predictive power are especially crucial for analysts working under privacy constraints. While there are a number of algorithms that compute regression coefficients (βs) in a differentially private manner (typically by adding noise) [3, 23, 25, 26, 27], there is no guarantee that the underlying model accurately describes the relationships in the confidential data, nor that it yields high-quality predictions. Similarly, a model that fits or predicts well on synthetic data may not do so for the confidential data. Unfortunately, analysts working under privacy constraints cannot generate residual plots and ROC curves using the confidential data. Indeed, releasing residuals [18] or ROC curves [13] computed directly on the confidential data can disclose sensitive properties about individual records.

Despite its importance, there has been little work on designing private regression diagnostics. We are not aware of any work on assessing model fit in a differentially private manner, especially using plots of residuals. Boyd et al. [2] consider the problem of computing the AUC under differential privacy, but there is no known algorithm for plotting ROC curves.[1]

In this article, we develop the first algorithms for differentially private regression diagnostics. Our contributions are:

- We design private algorithms for constructing plots of residuals and ROC curves. An important challenge in these tasks is to preserve the visual characteristics of these plots, since most analysts are trained to assess model fit based on visual inspections. This is a departure from prior work on differential privacy that has solely focused on optimizing error on counting queries.
- Using real datasets, we show that our private ROC curves approximate the true ROC curves well. Moreover, when the product of the privacy loss ϵ and dataset size is at least 1000, our techniques can distinguish ROCs with 0.025 difference in AUC.
- Plotting residuals under differential privacy is challenging, since *a priori* there is no upper bound on how large a residual can be. Hence, we first design a differentially private algorithm to estimate an upper bound, and then use differentially private space partitioning techniques to visualize the density of the residuals in this bounded region. Using synthetic datasets, we perform controlled experiments to show that an analyst can accurately determine whether or not the regression assumptions are satisfied when the product of the dataset size and ϵ exceeds 1000.

Organization: The remainder of the paper is organized as follows. Section 2 introduces some basic knowledge about differential privacy. In Section 3, we present the differentially private algorithm for estimating the ROC curve in logistic regression and evaluate its utility. In Section 4, we present the differentially private algorithm for estimating plots of residuals versus predicted values in linear regression and show comprehensive experiments for evaluation of its performance. We discuss related work in Section 5, and conclude in Section 6.

[1]Except a preliminary version of our work [24].

2. PRELIMINARIES

Let D be a confidential dataset comprising of n individuals. For each $i \in [n]$, the i^{th} record is composed of an outcome y_i and a $p \times 1$ vector of predictor variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$. For our discussion on logistic regression (in Section 3), we will restrict $y_i \in \{0, 1\}$ to be a binary outcome.

We define a neighborhood relation Q on databases as follows: Two databases D_1 and D_2 are considered neighboring datasets if D_1 and D_2 differ in one entry. That is, there is some i where D_1 has $(\mathbf{x}_i, y_i)$ but D_2 has $(\mathbf{x}'_i, y'_i)$, and the rest of the records in the two databases are the same. An algorithm satisfies differential privacy [5] if its outputs are statistically similar on neighboring databases.

DEFINITION 1 (ϵ-DIFFERENTIAL PRIVACY). *A randomized algorithm $\mathcal{M}$ satisfies ϵ-differential privacy if for any pair of neighboring databases $(D_1, D_2) \in Q$, and $\forall S \in$ range($\mathcal{M}$),*

$$Pr[\mathcal{M}(D_1) = S] \leq e^{\epsilon} \cdot P[\mathcal{M}(D_2) = S].$$

The value of ϵ, called the *privacy budget*, controls the level of the privacy, and limits how much an adversary can distinguish one dataset with its neighboring datasets given the output. Smaller ϵ's correspond to more privacy.

Differentially private algorithms satisfy the following *composition* properties [14]. Suppose $M_1(\cdot)$ and $M_2(\cdot)$ be ϵ_1- and ϵ_2-differentially private algorithms.

- *Sequential Compositon:* Releasing the outputs of $M_1(D)$ and $M_2(D)$ satisfies $\epsilon_1 + \epsilon_2$-differential privacy.
- *Postprocessing*: For any algorithm $M_3(\cdot)$, releasing $M_3(M_1(D))$ still satisfies ϵ_1-differential privacy. That is, post-processing an output of a differentially private algorithm does not incur any additional loss of privacy.

Thus, complex differentially private algorithms can be built by composing simpler private algorithms. The *Laplace Mechanism* [5] is one such widely used building block that achieves differential privacy by adding noise from a Laplace distribution with a scale proportional to the *global sensitivity*.

DEFINITION 2 (GLOBAL SENSITIVITY). *The global sensitivity of a function $f : \mathcal{D} \rightarrow \mathbb{R}^n$, denoted as $\Delta(f)$, is defined to be the maximum L_1 distance of the output from any two neighboring datasets D_1 and D_2.*

$$\Delta(f) = \max_{(D_1, D_2) \in Q} ||f(D_1) - f(D_2)||_1.$$

DEFINITION 3 (LAPLACE MECHANISM). *For any function $f : \mathcal{D} \rightarrow \mathbb{R}^d$, the Laplace Mechanism $\mathcal{M}$ is given by: $\mathcal{M}(D) = f(D) + \eta$. η is a vector of independent random variables drawn from a Laplace distribution with the probability density function $p(x|\lambda) = \frac{1}{2\lambda} e^{-|x|/\lambda}$, where $\lambda = \Delta(f)/\epsilon$.*

THEOREM 1. *Laplace Mechanism satisfies ϵ-differential privacy.*

3. ROC CURVES

We now present an algorithm for differentially private ROC curves for logistic regression. We note, however, that the algorithm could be used to measure the predictive power of any binary classifier.

3.1 Review of ROC curves

Let $D_{raw} = \{(y_i, \mathbf{x}_i) : i \in [n]\}$ be a confidential dataset. The logistic regression model posits that $P(y_i = 1 \mid \mathbf{x}_i) = \pi_i$, where $\log(\pi_i/(1-\pi_i)) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j$. Let $b = (b_0, \ldots, b_p)$ be the maximum likelihood estimate of $\beta = (\beta_0, \ldots, \beta_p)$. For any $\mathbf{x}_i$, we can compute predicted probabilities,

$$p(\mathbf{x}_i) \;=\; P(y_i = 1 \mid \mathbf{x}_i) \;=\; \frac{exp(b_0 + \sum_{j=1}^{p} x_{ij} b_j)}{1 + exp(b_0 + \sum_{j=1}^{p} x_{ij} b_j)}. \quad (1)$$

Analysts often evaluate the fit of a logistic regression model by comparing $p(\mathbf{x}_i)$ to y_i. Ideally, $p(\mathbf{x}_i)$ tends to be large when $y_i = 1$, and tends to be small when $y_i = 0$. To facilitate these comparisons, it is convenient to convert $p(\mathbf{x}_i)$ into a predicted outcome, $\hat{y}_i \in \{0, 1\}$, by setting $\hat{y}_i = 1$ when $p(\mathbf{x}_i) > \theta$ and $\hat{y}_i = 0$ otherwise. Here, θ is an analyst-defined threshold.

For any given θ, using all n values of $(y_i, \hat{y}_i)$, we can quantify the accuracy of the logistic regression model on a hold out test set D as follows. *True positives*, $TP(\theta)$, are the individuals in D with true and predicted outcomes equal to 1, i.e., $(y_i = 1, \hat{y}_i = 1)$. *False positives*, $FP(\theta)$, are individuals with $(y_i = 0, \hat{y}_i = 1)$. We use the notation $TP(\theta), FP(\theta)$, etc., to denote both the set of individuals as well as the cardinality of these sets.

Let n_1 and n_0 be the number of individuals with $y_i = 1$ and $y_i = 0$, respectively. The true-positive rate $TPR(\theta)$ is the probability that an individual in D with $y_i = 1$ is correctly classified with $\hat{y}_i = 1$. The false-positive rate $FPR(\theta)$, is the probability that an individual in D with $y_i = 0$ is wrongly classified with $\hat{y}_i = 1$. Thus, we have

$$TPR(\theta) \;=\; \frac{TP(\theta)}{n_1} \quad \text{and} \quad FPR(\theta) \;=\; \frac{FP(\theta)}{n_0}. \qquad (2)$$

The ROC curve is defined by plotting $TPR(\theta)$ versus $FPR(\theta)$ over all possible θ. It starts at $(0, 0)$ for $\theta = 1$ and ends at $(1, 1)$ for $\theta = 0$. It can be approximated numerically by computing Equation 2 over a large set of candidate θ values, say Θ. The area under the ROC curve, abbreviated as AUC, is often used to evaluate the accuracy of the logistic regression. When the regression predicts the outcomes in D accurately, the ROC curve is close to the left and upper boundary, so that AUC is close to 1. When the regression predicts the outcomes in D poorly, the ROC curve is close to the 45° line from (0,0) to (1,1), and the AUC is around 0.5.

Recent work [13] has shown that releasing actual ROC curves computed directly from Equation 1 can allow attackers with prior knowledge to reconstruct D. For an extreme yet illustrative example, suppose an attacker knows all of D except for $(y_i, \mathbf{x}_i)$. Given the ROC curve based on Equation 1, the attacker can determine the unknown y_i by simply enumerating over all values of $(y_i, \mathbf{x}_i)$, and finding the value that reproduces the given ROC curve. Hence, directly releasing the ROC curve may leak information, inspiring us to create a differentially private method for generating ROC curves.

We note that one can compute a differentially private AUC (but not the ROC curve itself) and a variant of ROC curve called the symmetric binormal ROC curve based on noisy AUC, as described in [2]. To do so, one adds noise to the AUC value using the smooth sensitivity paradigm [15]. However, the smooth sensitivity of the AUC can be high when n is small, or when either of n_1 or n_0 is small (which happens in many contexts); this can result in noisy AUC values with excessive error.

3.2 Private ROC curves

To generate differentially private ROC curves, we must (i) privately compute TPR and FPR values, (ii) decide how many and what thresholds θ to use, and (iii) ensure the monotonicity of the TPR and FPR values, that is, for all $\theta_1 \geq \theta_2$, the private versions of $TPR(\theta_1) \leq TPR(\theta_2)$ and of $FPR(\theta_1) \leq FPR(\theta_2)$. These three tasks are not trivially implemented. One might be tempted to use the *Laplace mechanism* to compute $TPR(\theta)$ and $FPR(\theta)$, since the global sensitivity of releasing n_0 and n_1 is 1, as is the sensitivity of each $TP(\theta)$ and $FP(\theta)$. Thus, they all can be released by adding Laplace noise with sensitivity $2|\Theta| + 1$, where $|\Theta|$ is the number of thresholds; however, large $|\Theta|$ can lead to unacceptable errors. In the non-private case, one can set $\Theta = \{p(\mathbf{x}_i) : \mathbf{x}_i \in D, i \in [n]\}$, i.e., the predicted probabilities for each individual. Under differential privacy, we can not use this set of thresholds since predictions themselves cannot be publicly released, and hence the thresholds must be chosen in a private manner. Finally, adding noise to $TPR(\theta)$ and $FPR(\theta)$ could result in noisy values that violate monotonicity.

We propose a novel algorithm *PriROC* that addresses these concerns in three steps: (i) privately selecting thresholds using ϵ_1 of the privacy budget, (ii) computing noisy TPR and FPR values using the remaining budget $\epsilon_2 = \epsilon - \epsilon_1$ on all selected thresholds, and (iii) post-processing the output to ensure that the noisy TPR and FPR values correspond to a valid, monotonic ROC curve.

3.2.1 Choosing Thresholds

There are two important considerations when choosing the set of thresholds Θ. The number of thresholds must not be very large in order to reduce error as well as to ensure good runtime (as our technique to ensure monotonicity of TPR and FPR values in the post-processing step is $O(|\Theta|^3)$). At the same time, enough thresholds must be chosen so that the resulting ROC curve is a close approximation of the true ROC curve. We present two heuristics for choosing Θ that take into account these considerations.

A simple, data-independent strategy for picking Θ is to choose them uniformly from $[0, 1]$. More precisely, if we fix $|\Theta| = N$. then we set $\Theta = \{0, \frac{1}{N}, \ldots, \frac{N-1}{N}, 1\}$. We call this strategy N-FIXEDSPACE. This strategy works well when the predictions $P = \{p(\mathbf{x}_i) | \mathbf{x}_i \in D, i \in [n]\}$ are uniformly spread out in $[0, 1]$. Since, N-FIXEDSPACE is data independent, $\epsilon_1 = 0$, and all the privacy budget can be used for computing the TPR and FPR values. However, N-FIXEDSPACE may be ineffective when predictions are skewed. For instance, suppose all the predictions $p(\mathbf{x}_i)$ are either $< \frac{1}{N}$, or $> 1 - \frac{1}{N}$. The ROC curve is then approximated with just 2 points $(TPR(\frac{1}{N}), FPR(\frac{1}{N}))$ and $(TPR(1 - \frac{1}{N}), FPR(1 - \frac{1}{N}))$, possibly resulting in a significant loss in accuracy in the AUC.

For this potential shortcoming, we present s-RECURSIVEMEDIANS (Algorithm 1), a data-dependent strategy that recursively partitions the data domain so that each partition has roughly the same number of individuals. Algorithm 1 takes as input the privacy budget for choosing thresholds ϵ, depth of the recursion s, and the multi-set of predictions P. Each of the s recursive steps uses a privacy budget of $\epsilon/2s$, due to bounded differential privacy.

The algorithm recursively calls a subroutine FINDMEDIANS to compute the noisy median of all predictions within the range (ℓ, r). Initially, $(\ell, r) = (0, 1)$. Since the median function f_{med} has a high global sensitivity (equal to $r - \ell$), we use the smooth sensitivity framework [15] for computing the noisy median.

When the global sensitivity of a function $\Delta(f)$ is high, we can use a data dependent sensitivity function $S^*_{f,\epsilon}(D)$ instead to ensure privacy. $S^*_{f,\epsilon}(D)$ is an upper bound on the (local) sensitivity of f at the input database (i.e. the maximum change in function output on between D and its neighbors) and is smooth (i.e., $S^*_{f,\epsilon}(D)$ is close to $S^*_{f,\epsilon}(D')$, when D and D' are neighbors). Given $S^*_{f,\epsilon}(\cdot)$, one way to ensure ϵ-differential privacy is by perturbing the true output $f(D)$ using $\frac{8}{\epsilon} \cdot S^*_{f,\epsilon}(D) \cdot \eta$, where η is random noise sampled from

Algorithm 1 s-RECURSIVEMEDIANS

function s-RECURSIVEMEDIANS(P, ϵ, s)
  $\epsilon_1 \leftarrow \frac{\epsilon}{2s}$
  return FINDMEDIANS($P, \epsilon_1, s, 0, 1$)
end function
function FINDMEDIANS(P, ϵ, s, ℓ, r)
  if $k = 0$ **then return** $\emptyset$
  end if
  $m \leftarrow median(P)$
  $\tilde{m} \leftarrow m + \frac{8S^*_{f_{med},\epsilon}(P)}{\epsilon} * \eta$, η is random noise $\propto \frac{1}{1+\eta^2}$
  if $\tilde{m} \leq \ell$ or $\tilde{m} \geq r$ **then** $\tilde{m} \leftarrow (\ell + r)/2$
  end if
  $P_1 \leftarrow \{P[i] \mid P[i] < \tilde{m}\}$
  $P_2 \leftarrow \{P[i] \mid P[i] > \tilde{m}\}$
  return FINDMEDIANS($P_1, \epsilon, s - 1, \ell, \tilde{m}$) $\cup$ $[\tilde{m}] \cup$ FINDMEDIANS($P_2, \epsilon, s - 1, \tilde{m}, r$)
end function

the distribution $\propto 1/(1 + |\eta|^2)$. We refer the reader to [15] for proofs and details.

For the median function f_{med}, the smooth sensitivity $S^*_{f_{med},\epsilon}(P)$ can be computed as [15]:

$$S^*_{f_{med},\epsilon}(P) = \max_{k=0,\ldots,n}(e^{-k\epsilon} \cdot \max_{t=0,\ldots,k+1}(P[m+t] - P[m+t-k-1])) \quad (3)$$

where $P[t]$ is the t^{th} value in P when sorted in increasing order. We generate samples η by drawing U uniformly from $(0, 1)$ and computing $tan(\pi(U - 0.5))$ (as the CDF of the noise distribution is $\propto arctan(\eta)$).

The resulting noisy median $\tilde{m}$ could fall out of the range (ℓ, r). This could happen by random chance or because the smooth sensitivity of the points within the range is high. When either occurs, we use a point in the middle of the range (e.g., $(\ell + r)/2$) as the partition point instead of $\tilde{m}$. The algorithm proceeds recursively to find the medians of points in $(\ell, \tilde{m})$ and $(\tilde{m}, r)$. The algorithm returns after it completes s levels of recursion. The number of thresholds output by s-RECURSIVEMEDIANS is 2^s.

THEOREM 2. *Algorithm 1 (s-RECURSIVEMEDIANS) satisfies ϵ-differential privacy.*

PROOF. (sketch) The proof follows from the following statements. Computing the median of a set of points in each iteration of FINDMEDIANS satisfies $\epsilon/2s$-differential privacy. In each recursive step, there are at most two partitions different in neighboring databases, satisfying $2 \times \epsilon/2s = \epsilon/s$-differential privacy. Since the depth of the recursion is bounded by s, s-RECURSIVEMEDIANS satisfies ϵ-differential privacy by sequential composition. □

3.2.2 Computing noisy TPRs and FPRs

Suppose we are given a set of thresholds $\Theta = \{\theta_1, \ldots, \theta_\ell\}$ to use for approximating the ROC curve, where $\theta_k > \theta_{k+1}$ for all k, $\theta_0 = 1$, and $\theta_\ell = 0$. By definition, $TP(\theta_\ell) = n_1$ and $FP(\theta_\ell) = n_0$. The sets $\{TP(\theta_k) : k \in [\ell]\}$ and $\{FP(\theta_k) : k \in [\ell]\}$ each correspond to a set of one-sided range queries, defined as follows.

DEFINITION 4 (ONE-SIDED RANGE QUERY). *Let $R = \{r_1, \ldots, r_t\}$ denote a set of t counts. A query q_j is called a* one-sided range query *when $q_j(R)$ is the sum of the first j elements in R, i.e., $q_j(R) = \sum_{k=1}^{j} r_k$. The set $C_t = \{q_1, \ldots, q_t\}$ denotes the workload of all one-sided range queries.*

For computing ROC curves, let $R^{TP}_\Theta = \{r^{TP}_1, \ldots, r^{TP}_\ell\}$, where r^{TP}_k is the number of individuals $i \in D$ with $y_i = 1$ and $\theta_{k-1} \geq p(\mathbf{x}_i) > \theta_k$. Clearly, $TP(\theta_k)$ is the sum of the first k counts in R^{TP}_Θ. We can define R^{FP}_Θ similarly, thereby showing that each $FP(\theta_k)$ is also the answer to a one-sided range query on R^{FP}_Θ.

It is well known that the *Laplace mechanism* is not optimal in terms of error for the workload of one-sided range queries C_t. Under the *Laplace mechanism*, each query answer would have an expected mean square error of $O(t^2/\epsilon^2)$. A recent experimental study of differentially private algorithms for answering range queries [9] indicates that *DAWA* [11] and *HB* [17] are the best algorithms for answering one-sided range queries. HB works by constructing $\log t$ equiwidth histograms at varying resolutions, and using the *Laplace mechanism* to answer each histogram with a privacy budget of $\epsilon/\log t$. *DAWA* additionally approximates the histograms R^{TP}_Θ and R^{FP}_Θ by grouping together similar counts.

Using one of these two algorithms we compute the TP and FP counts privately using a privacy budget of $\epsilon_2/2$ for each count. One-sided range queries are computed on $\hat{R}^{TP}_\Theta$ to get the TP counts and on $\hat{R}^{FP}_\Theta$ to get the FP counts, which in turn are used to construct the noisy $TPR(\theta)$ and $FPR(\theta)$ values. Since all subsequent steps do not use the original data, the fact that releasing $TPR(\theta)$ and $FPR(\theta)$ satisfies ϵ_2-differential privacy follows from the privacy of *DAWA* or *HB*.

3.2.3 Ensuring monotonicity

The noisy TPR and FPR values generated using the algorithms from Section 3.2.1 and 3.2.2 are not guaranteed to satisfy monotonicity. We leverage the ordering constraint between the TPR and FPR values to boost accuracy by using a constrained inference method [10]. Since this is a post-processing step, there is no impact on privacy.

THEOREM 3. PriROC *satisfies ϵ-differential privacy.*

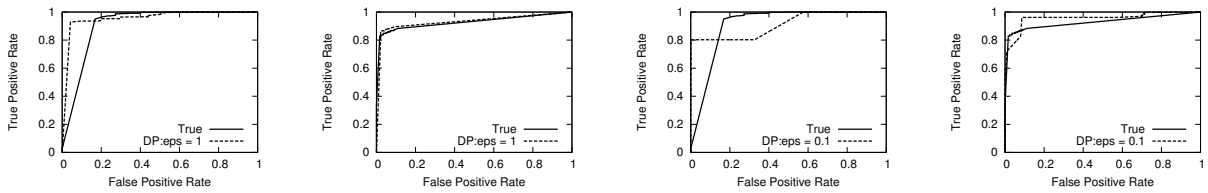
PROOF. We have already shown that selecting thresholds with Algorithm 1 ensures ϵ_1 differential privacy (Theorem 2). Using *DAWA* or *HB* to perturb $TPRs$ and $FPRs$ satisfies ϵ_2 differential privacy. Post-processing to ensure monotonicity does not leak extra information. Thus, *PriROC* ensures $\epsilon_1 + \epsilon_2 = \epsilon$-differential privacy. We note that a data independent method for selecting Θ, like using N-FIXEDSPACE, incurs no privacy loss, so that all of ϵ can be devoted to *DAWA* or *HB* for perturbing $TPRs$ and $FPRs$. □

3.3 Evaluation

In this section we empirically evaluate the quality of the ROC curves output by *PriROC*. Our experiments focus on:

- Visually inspecting whether the ROC curves output by *PriROC* are close to the true ROC curves generated using the confidential predictions.
- Measuring the average absolute difference between the AUC for the true ROC curve and the AUC for noisy ROC curves output by *PriROC*.
- Measuring the similarity between true and noisy ROC curves by computing the symmetric difference between them.
- Quantifying the discriminatory power of private ROC curves; i.e., whether they can discriminate between "good" and "bad" classifiers.

Dataset: We use two text classification datasets - RAW-TWITTER and RAW-SMS. The RAW-TWITTER dataset [8] was collected for the task of sentiment classification. Each tweet is associated with a binary sentiment outcome – positive or negative. The dataset contains 1.6 million tweets from which we randomly sampled 6840 tweets for our experiments. The RAW-SMS dataset [1] contains 5574 SMS messages associated with spam/ham outcome variable.



(a) SMS,$\epsilon = 1$ (b) TWITTER,$\epsilon = 1$ (c) SMS,$\epsilon = 0.1$ (d) TWITTER,$\epsilon = 0.1$

Figure 1: Confidential and differentially private ROC curves using *PriROC* at $\epsilon = 1$ (left) and $\epsilon = 0.1$ (right).

For both datasets, we use a vector of binary explanatory variables representing presence or absence of individual words (excluding stop words).

RAW-TWITTER and RAW-SMS are randomly divided into a training set (containing 90% records) and a test set (containing 10% records). We use differentially private empirical risk minimization [3] to compute a logistic regression model on the training set. Then we apply the private model on the two test datasets to obtain predictions, which we call SMS and TWITTER. SMS contains 558 records, and each record has a true label $y_i \in \{0, 1\}$ as well as a prediction $P(y_i = 1 \mid \mathbf{x}_i) \in [0, 1]$. In SMS, 481 out of 558 records have label 1. The TWITTER contains 684 different records, 385 of which have label 1.

Algorithms: We consider two strategies for selecting Θ. The first uses the s-RECURSIVEMEDIANS strategy, setting $|\Theta| = \max\{2^l : 2^l \leq n\}$ where n is the size of the test dataset (we suppose n is known), and spending 0.2ϵ budget to select thresholds. We refer to the resulting differentially private algorithm for ROC curves as *PriROC*.

The second strategy for selecting Θ, used for comparisons in forthcoming sections, uses fixed thresholds selected uniformly from $[0, 1]$. Here, we again set $|\Theta| = \max\{2^l : 2^l \leq n\}$, and we dedicate the full ϵ to perturb the $TPRs$ and $FPRs$. We call the resulting differentially private algorithm for ROC curves *PriROC-fix*.

For both *PriROC* and *PriROC-fix*, we apply both *DAWA* and *HB* to perturb the TPR and FPR values. Since we find *DAWA* consistently outperforms *HB* in the experiments. All the results shown in this section are based on *DAWA*.

We also consider Boyd et al's [2] algorithm for directly computing the AUC, and call that *SmoothS*.

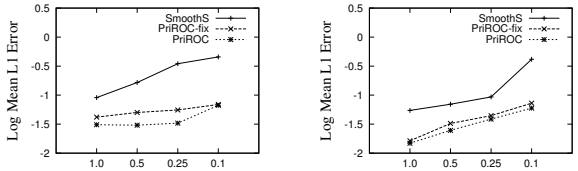
3.3.1 Visualizing ROC curves

Figure 1 displays the ROC curve computed on confidential data and one randomly generated differentially private ROC curve at ϵ = 1 and 0.1 using *PriROC* on both SMS and TWITTER. The *PriROC* curves track the confidential ROC curves quite well for all datasets and ϵ settings. The reason why the *PriROC* for TWITTER is more accurate than SMS is that the prediction distribution of SMS is more skewed than TWITTER s.t. s-RECURSIVEMEDIANS performs better on TWITTER.

3.3.2 AUC Error

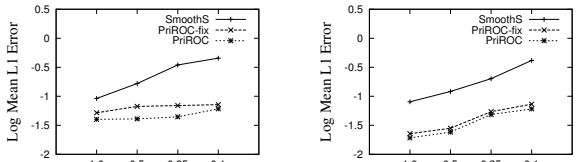
Figure 2 reports the comparison of AUC error for the three algorithms *PriROC*, *PriROC-fix* and *SmoothS* on both SMS (Figure 2(a)) and TWITTER (Figure 2(b)). Errors reported are means over 20 independent trials. Both *PriROC* and *PriROC-fix* have significantly lower errors than *SmoothS* under both datasets and all ϵ settings. Using s-RECURSIVEMEDIANS to choose thresholds results in better AUC accuracy compared to using fixed thresholds.

3.3.3 ROC Similarity



(a) SMS (b) TWITTER

Figure 2: Comparison of AUC error. For each figure, the x-axis represents different ϵ settings and the y-axis shows the $\log_{10}$ of the mean L_1 error between the confidential AUC and the private AUC over 20 repetitions.



(a) SMS (b) TWITTER

Figure 3: Comparison of symmetric difference between confidential and private ROC curves. For each figure, the x-axis represents different ϵ settings, and the y-axis shows the $\log_{10}$ of the mean symmetric difference between the confidential and private ROC curves over 20 repetitions.

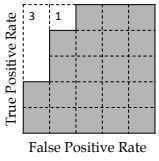
Even when two ROC curves have the same AUC, they may look quite different. Here, we directly examine how close the real ROC and private ROC are by computing the symmetric difference between them; i.e., the mean area between the true and private curves. We compare three methods: *PriROC*, *PriROC-fix*, and *SmoothS* (*SmoothS* computes a data independent, symmetric binormal ROC curves based on the noisy AUC of [2]). Figure 3 displays the symmetric differences (averaged over 20 trials) between the confidential ROC curves and differentially private ROC curves for the three different algorithms on both SMS (Figure 3(a)) and TWITTER (Figure 3(b)). Both *PriROC* and *PriROC-fix* generate private ROC curves with much lower symmetric differences than those for *SmoothS*. Once again, *PriROC* offers slight gains over *PriROC-fix*.

3.3.4 Discriminatory Power

An evaluation methodology is useless if it can't discriminate "good" regression models (or binary classifiers) that have high AUCs from "bad" models that have low AUCs on a given test set. To quantify this *discriminatory power*, we generate synthetic datasets with different AUCs as follows (Figure 4(b) is an example). We

Scale*ε	Discriminatory Power	Max p-value
200	0.1	3.7E-3
500	0.05	1.31E-2
1000	0.025	3.29E-3
2000	0.025	1.58E-5

(a) p-value of two sample t-test



(b) Sync ROC

Figure 4: Discriminatory Power of *PriROC*

construct a test dataset with n rows, half of which have $y_i = 0$ and the other half have $y_i = 1$ (note that $n = 10$ in Figure 4(b)). A logistic regression that is being evaluated associates a prediction with each row of the test dataset. Sorting the predictions in descending (or ascending) order results in a binary sequence (corresponding to the true y of the records in that order). The ROC curve then can be drawn by starting from (TPR=0, FPR=0) drawing a line segment vertically up of length $2/n$ if we see a 1 and drawing a line horizontally to the right of length $2/n$ if we see a 0. Since there are $n/2$ 0s and 1s, we end up at (TPR=1, FPR=1).

The area above the ROC curve can be decomposed into a set of rectangles of width $2/n$, we can construct an ROC curve with AUC a by picking any integral solution to the problem $g_1 + g_2 + \ldots + g_{n/2} = (1 - a) \cdot n/2$ such that $g_1 \geq g_2 \geq \ldots \geq g_{n/2}$. In Figure 4(b), $g_1 = 3, g_2 = 1$ and $g_3 = g_4 = g_5 = 0$, which in turn corresponds to the sequence of y values $[1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0]$. We can thus construct ROC curves for any AUC that is a multiple of $4/n^2$.

To determine the discriminatory power of *PriROC*, we generate a pair of synthetic ROC curves R_a and $R_{a+\delta}$ with AUCs a and $a+\delta$, respectively. We then construct noisy ROC curves for both R_a and $R_{a+\delta}$ 20 times each, and compute their AUC. We then perform two sample t-tests (between noisy AUC's generated from R_a and $R_{a+\delta}$, respectively) and test whether the mean noisy AUC generated from R_a is significantly different (p-value < 0.05) from the mean noisy AUC generated from $R_{a+\delta}$. We say that the noisy AUC computed using *PriROC* at data size n and privacy ϵ has discriminatory power δ, if $\forall a$, the noisy AUC values for R_a have a significantly different mean than noisy AUC values for $R_{a+\delta}$.

Figure 4(a) displays the discriminatory power of private ROC curves. We generate ROC curves with AUC $a \in A = \{0.95, 0.925, \ldots, 0.7\}$. We report the smallest δ (in multiples of 0.025) at which noisy AUCs have significantly different means for R_a and $R_{a+\delta}$ for $a, a + \delta \in A$. We also report the max p-value attained for all such comparisons R_a and $R_{a+\delta}$. Smaller p-values imply the AUC computed using *PriROC* are more discriminating between pairs R_a and $R_{a+\delta}$. Rather than varying n and ϵ independently, we only vary the product $n \cdot \epsilon$. Recent work [9] has shown that algorithms like DAWA satisfy scale-epsilon exchangeability; i.e., increasing n and ϵ have equivalent effects on utility (which we see in our experiments as well). Larger $n \cdot \epsilon$ products allow us to discriminate between ROC curves with smaller differences in AUC with more significance.

In summary, ROC curves computed by *PriROC* satisfy ϵ-differential privacy, and closely match the true ROC curves both visually and in terms of qualitative measures like AUC and symmetric difference. Moreover, for $n \cdot \epsilon \geq 1000$, private ROC curves can discriminate between "good" and "bad" models even if they differ in AUC by only 0.025.

4. RESIDUAL PLOT

In this section, we introduce a differentially private algorithm for computing distributions of residuals, which are used to verify model fit. As far as we know, this is the first work on generating plots of residuals under differential privacy.

4.1 Review of Residual Diagnostics

Let $D = \{(y_i, \mathbf{x}_i) : i \in [n]\}$ be a confidential training dataset. Linear regression models the outcome as:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \xi_i, where\ \xi_i\ \sim N(0, \sigma^2),\ \forall i.$$

Here, $\beta = (\beta_0, \ldots, \beta_p)$ is the $(p+1) \times 1$ vector of true regression coefficients; these are unknown and estimated from D, typically using ordinary least squares (OLS). The model is based on four key assumptions. One is that each ξ_i is independent, so that individual outcomes are independent; we take this as given. The remaining assumptions are listed below in order of decreasing importance.

1. At any value of $\mathbf{x}_i$, $E(y_i \mid \mathbf{x}_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$.
2. At any value of $\mathbf{x}_i$, the true variance of y values around the regression line is constant, i.e., $Var(y_i \mid \mathbf{x}_i) = \sigma^2$.
3. At any value of $\mathbf{x}_i$, ξ_i follow a normal distribution.

The ideal check of these assumptions is to examine the distribution of ξ_i to see if it lines up with $N(0, \sigma^2)$. Since ξ_i can never be known, we instead examine the distribution of the residuals, $r_i = y_i - \mathbf{x}_i b$, where b is the estimate of β computed from OLS. When the regression model assumptions are valid, one can show that $E(r_i \mid \mathbf{x}_i) = 0$ and that $Cov(r_i, \mathbf{x}_i) = 0$ for any i. Hence, when the assumptions are reasonable, the residuals should be centered around zero and exhibit no systematic pattern with any function of $\mathbf{x}$. For example, the plot of r versus the predicted values, $\hat{y} = \mathbf{x}b$, should look like a randomly scattered set of points (see Figure 5(a)). On the other hand, when linearity is violated, the plot exhibits systematic curvatures or increasing (decreasing) trends of r with $\hat{y}$ (see Figure 5(c)). When constant variance is violated but linearity holds, we often see a fan shape to the residuals, e.g., the spread of the residuals increases with $\hat{y}$ (see Figure 5(b)). This phenomenon is called *heteroscedasticity*. When normality is violated but the linearity and constant variance hold, we can see a few residuals that are outliers. With bad fitting models, it is often the case that multiple assumptions fail simultaneously, and the plots exhibit non-random patterns. It is important to note, however, that residuals only can reveal problematic models; a non-random pattern only indicates lack of evidence that the model is mis-specified, not proof that it is correctly specified.

Of course, when D is confidential, one cannot release the residual plot as is. For example, given b (or a noisy version of b), and the values of $\mathbf{x}_i$ for some target record i, an intruder can deduce the confidential value of y_i simply by computing $r_i + \mathbf{x}_i b$. Hence, we propose to release a noisy version of the plot of residuals versus predicted values that reveals whether or not one can trust the results from the specified regression.

4.2 Private Residual Plots

We propose the algorithm *PriRP* for constructing ϵ-differentially private residual plots of r versus $\hat{y}$. *PriRP* proceeds in two steps: 1) privately compute bounds on the range of $\hat{y}$ and r in the graph using ϵ_1 of the total ϵ budget, and 2) perturb the distribution of residuals within the bounds using the remaining budget $\epsilon_2 = \epsilon - \epsilon_1$. We now present these two steps in details.

4.2.1 Computing private bounds of graph

To understand why the first step is generally needed, suppose that we have little knowledge about the distributions of variables in

Algorithm 2 Private Bounds

Input: Dataset D, unit μ, private budget ϵ, threshold θ
Output: a value d (indicating a bound $[-d, d]$)

$\tilde{\theta} \leftarrow \theta + Lap(2/\epsilon)$
$d = \mu$
while True **do**
  $q \leftarrow$ number of $x \in D$ within bound $[-d, d]$
  if $q + Lap(4/\epsilon) < \tilde{\theta}$ **then**
    $d \leftarrow d * 2$
  else
    return d
  end if
end while

D. Since $(\mathbf{x}_i, y_i, b)$ is unknown, theoretically the range of both $\hat{y}$ (on the x-axis) and r (on the y-axis) could go to infinity. Obviously, we cannot construct a graph with unbounded axes. However, using bounds that are too small, i.e., they leave out large fractions of the data, provides a view of the residuals over a region that may be too small to allow the user to distinguish violations of the linear regression assumptions. On the other hand, using bounds that are too large can distort the plot, as the irrelevant area may affect the performance of perturbing residual distribution in the second step.

Because the axes potentially are unbounded, we cannot simply use a method like Laplace Mechanism to compute the bounds under differential privacy. Indeed, even the smooth sensitivity [15] of the range cannot be bounded. Instead, we propose Algorithm 2 to find the bounds on each axis. The intuition behind Algorithm 2 is to search for bounds that include some pre-specified number of records θ (e.g., $\theta = 0.95 * n$, where $n = |D|$). To do so, the algorithm successively answers the query, $q_i =$"number of observations in D within the bounds $[-\mu * 2^i, \mu * 2^i]$", where $i = 0, 1, \ldots$ and μ is a scaling constant, until it finds the first q_i that exceeds θ. Algorithm 2 can be viewed as an application of the *Sparse Vector Technique* [6], which is known to ensure ϵ-differential privacy.

THEOREM 4 ([6]). *Algorithm 2 satisfies ϵ-differential privacy. Moreover, the algorithm halts after k queries $q_1, \ldots, q_k$ with probability $1 - \beta$ if for all but the last query $q_i(D) < \theta - \alpha$, where $\alpha = \frac{8(\log k + \log(2/\beta))}{\epsilon}$.*

The input parameters in Algorithm 2 can be readily interpreted. The parameter μ defines the unit quantity of $\hat{y}$ and r. It should be specified so that one gets sensible values of $\hat{y}$ and r without looping through Step 3 in Algorithm 2 too many times (in which case the bounds could be quite noisy, as the error grows logarithmically with the number of queries answered in the algorithm [6]) or too few times (in which case the bounds might be too wide). For example, when y describes an integer-valued outcome with few levels (e.g., a test score), it makes sense to set $\mu = 1$. When the attribute describes a long-tailed variable (e.g., salary), it is better to set $\mu = 1000$ or possibly even $\mu = 10000$. The parameter θ controls how much of the support of $\hat{y}$ or r we seek to represent in the plot. In our experiments, we assume $n = |D|$ is known and set $\theta = 0.95 * n$.

To compute the bounds for both axes, we use Algorithm 2 with privacy budget $\frac{\epsilon_1}{2}$ per axis. Thus, computing both bounds of graph ensures ϵ_1-differential privacy.

4.2.2 Perturbing residual plots

Regression diagnostics are primarily based on the patterns in the residual plots, e.g., whether there is evidence of non-linearity or non-constant variance. Hence, instead of directly perturbing each residual separately, we estimate the two dimensional probability density of the residuals in the bounded region computed using Algorithm 2. This is done using private space partitioning techniques in three steps – *discretization*, *perturbation* and *sampling*.

Discretization: We first discretize the bounded region into $m \times s$ grid cells of equal size. We then construct a histogram $\mathbf{h}$ with m^2 counts, where the count in each cell represents the number of records in D with values $(\hat{y}_i, r_i)$ contained in that cell. Based on analysis similar to [12], we set $m = \sqrt{\frac{N_0 \epsilon}{10}}$, where N_0 is the number of records in D that have $(\hat{y}_i, r_i)$ in the bounded region. Rather than estimating N_0 noisily (with some loss in privacy budget), we approximate N_0 using θ^2/n, the expected number of records that are captured in the bounded region by Algorithm 2.

Perturbation: We could now perturb the counts in histogram $\mathbf{h}$ using the Laplace mechanism. However, the number of cells in the histogram grows with N_0. E.g., for $\epsilon = 1$ and $n = 100000$, we would expect $N_0 = 0.95^2 n$ and $m = 0.95\sqrt{\frac{n}{10}} = 95$. Moreover, since residuals are usually clustered, a majority of the histogram cells will contain 0 counts. Adding noise to all these cells can result in perturbed histograms with high error.

Instead, we use the space partitioning technique *DAWA* [11] to perturb the histogram. Recent work [9] shows that *DAWA* is one of the best algorithms for privately answering range queries over 2D histograms. *DAWA* works by first ordering the cells in the histogram according to a Hilbert curve, and then grouping together contiguous ranges of cells (in this ordering) that have similar counts. *DAWA* then noisily computes the total counts for each of these groups, and reconstructs the original cell counts by assuming that all cell counts within a group are the same. Partitioning the cells into groups of similar counts allows *DAWA* to minimize the total noise added.

The noisy histogram counts $\tilde{\mathbf{h}}$ are postprocessed to ensure non-negativity (by setting negative values to 0), and integrality (by rounding to the nearest integer) to get $\hat{\mathbf{h}}$. This is the only step that consults the private database D, and uses a privacy budget of ϵ_2.

Sampling: The final perturbed residual plot is constructed by doing uniform sampling $\hat{\mathbf{h}}$ number of points at each grid cell.

THEOREM 5. PriRP *ensures ϵ-differential privacy.*

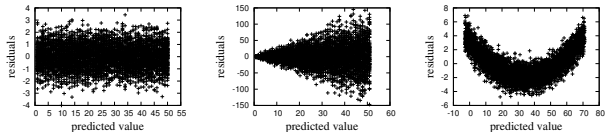
PROOF. Bounds on $\hat{y}$ and r can be computed satisfying ϵ_1-differential privacy using Algorithm 2. Perturbing residual plots within the bounds by using *DAWA* ensures ϵ_2-differential privacy. Therefore, *PriRP* satisfies $\epsilon_1 + \epsilon_2 = \epsilon$-differential privacy. □

Since the grouping of the histogram cells induced by *DAWA* is data dependent, a theoretical analysis of error is hard. Hence, we next empirically evaluate the residual plots output by $PriRP$.

4.3 Evaluation

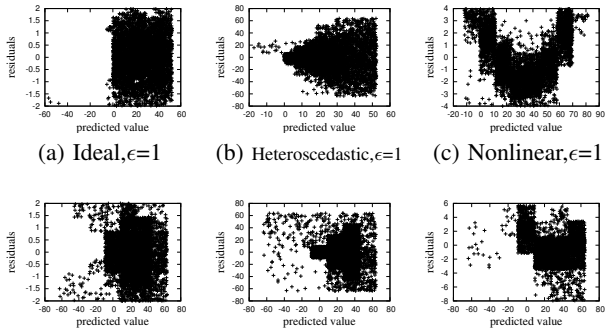
In this section, we empirically evaluate the quality of residual plots output by our algorithm *PriRP*, and whether or not an analyst can determine model fit using the noisy residual plots. We design the following four experiments:

1. We show that noisy residual plots can help analysts correctly identify whether or not the confidential data satisfies the regression assumptions (linearity and equal variance of error) using synthetic datasets.
2. We investigate the degree to which the linearity and unequal variance must be violated before an analyst can detect the violation using *PriRP*.
3. We next examine whether *PriRP* can evaluate model fit for a model output by a differentially private regression algorithm.
4. We finally present a case study of using *PriRP* to verify regression assumptions on a real dataset.



(a) Ideal (b) Heteroscedastic (c) Nonlinear

Figure 5: Residual plots for one confidential dataset randomly sampled from each of generative scenarios ($n = 5000$).



(a) Ideal,ϵ=1 (b) Heteroscedastic,ϵ=1 (c) Nonlinear,ϵ=1

(d) Ideal,ϵ=0.1 (e) Heteroscedastic,ϵ=0.1 (f) Nonlinear,ϵ=0.1

Figure 6: Differentially private versions of the residual plots from Figure 5 generated via *PriRP* with $\epsilon \in \{0.1, 1\}$.

Quality Measures: We evaluate *PriRP* using visual comparisons of confidential and private residual plots. We also quantify the similarity between each perturbed plot and the confidential data plots. Specifically, for any confidential plot RP and private residual plot PRP, we compute the metric as follows.

- Suppose RP contains points $(\hat{y}_i, r_i)$ for every record i. Let $min_x = \min\{\hat{y}_i\} - 0.1$, $max_x = \max\{\hat{y}_i\} + 0.1$, $min_y = \min\{\hat{r_i}\} - 0.1$, $max_y = \max\{r_i\} + 0.1$. We discretize the area between $[min_x, max_x] \times [min_y, max_y]$ into 10×10 equal-width grid cells.
- For each grid cell c_i, where $i = 1, \ldots, 100$, we compute the percentages $P(c_i)$ and $P_p(c_i)$ that points fall in grid cell i in RP and PRP, respectively.
- The similarity between RP and PRP is defined as the total variation distance between the two distributions:

$$Sim(RP, PRP) = \frac{1}{2}(\sum_{i=1}^{100} |P(c_i) - P_p(c_i)| + (1 - \sum_{i=1}^{100} P_p(c_i)))$$

The value of $Sim(\cdot, \cdot)$ is between [0,1]. Values close to zero indicate that the two plots have similar counts in each cell.

4.3.1 Ability to verify regression assumptions

Setup: To illustrate and evaluate the performance of the *PriRP* algorithm, we generate three sets of artificial datasets. The first set, we call "Ideal," perfectly follows the linear model assumptions; we set $y_i = x_i + \xi_i$, where $\xi_i \sim N(0, 1)$. The second set, which we call "Heteroscedastic," satisfies linearity but fails the constant variance assumption; we set $y_i = x_i + \xi_i$, where $\xi_i \sim N(0, x_i)$. The third set, which we call "Nonlinear," fails the linear assumption; we set $y_i = 0.01 \times x_i^2 + x_i + \xi_i$, where $\xi_i \sim N(0, 1)$. We generate each x_i from independent uniform distributions on $[1, 50]$. For each generative model, we create multiple confidential datasets of different size $n \in \{500, 1000, 2000, 5000\}$.

For each generated confidential dataset, we first use the OLS estimates, $b = (b_o, b_1)$, for the linear model $y_i = \beta_0 + \beta_1 x_i + \tau_i$, where $\tau_i \sim N(0, \sigma^2)$. This model is not appropriate for the "Heteroscedastic" or "Nonlinear" generating distributions; we fit these to examine whether or not the private residual plots can help analysts identify the lack of fit. We use b to compute each value of $\hat{y}_i$ and $r_i = y_i - \hat{y}_i$, which form the inputs to *PriRP*.

Results: Figure 5 displays the confidential residual plots for one dataset randomly sampled from "Ideal", "Heteroscedastic" and "Nonlinear". The usefulness of residual plots is clearly evident: the fanning pattern in the "Heteroscedastic" plot indicates the increasing spread in y with x, and the hook pattern in the "Nonlinear" plot indicates the quadratic relationship between y and x. In contrast, the "Ideal" case shows the classic random scatter that should be present when the assumptions of the posited model fit the data well.

Figure 6 displays exemplary private residual plots constructed from single runs of *PriRP* on the datasets in Figure 5. When $\epsilon = 1$, the overall pattern in the plots is well preserved—one can easily diagnose the "heteroscedasticity" and "nonlinearity". When $\epsilon = 0.1$, although these signals are diluted, the nonlinearity continues to be diagnosable.

We now examine how well the private residual plots reveal that the linear model assumptions are violated when they in fact are, especially as n and ϵ are varied. We compute the distribution of similarity values between the real "Ideal" residual plot and the noisy plots from different models ("Ideal", "Heteroscedastic" and "Nonlinear") using 1000 independently generated noisy plots in each case. In each plot in Figure 7, the distributions labeled I, H and N represent the empirical distribution of the similarity of the noisy "Ideal", "Heteroscedastic" and "Nonlinear" residual plot, respectively, with the true "Ideal" residual plot. When the product of n and ϵ is ≥ 1000, the distributions H and N are well separated from the distribution I. Even at $n \cdot \epsilon = 500$, we see little overlap between the distribution H and I as well as N and I. This means that at $n \cdot \epsilon \geq 500$, an analyst is very likely able to correctly tell using the noisy residual plot whether or not it represents the "Ideal" case.

Figure 7(e) and 7(j) confirm this using the total variation distance (TVD) between the distributions I vs H and I vs N. The TVD exceeds 0.95 when $n\epsilon \geq 1000$ and is ≥ 0.9 when $n\epsilon \geq 500$.

4.3.2 Discriminatory power

We investigate the degree to which the linearity and unequal variance must be violated before an analyst can detect the violation using residual plots generated using *PriRP*.

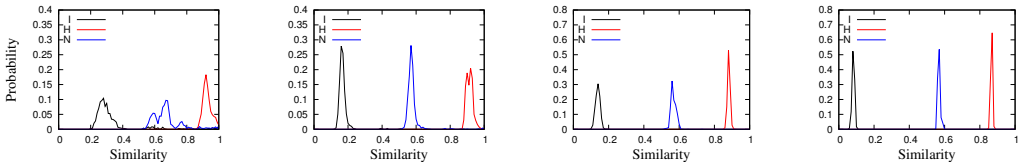
Setup: We generate data from the following three models, sampling each x_i from uniform distribution on [0,50].

1. M_1: $y_i = x_i + \xi_i, \xi_i \sim N(0, 1)$
2. M_2: $y_i = x_i + \xi_i, \xi_i \sim N(0, \alpha x_i + 1)$
3. M_3: $y_i = x_i + \beta x_i^2 + \xi_i, \xi_i \sim N(0, 1)$

When α and β are close to zero, the violation from the standard linear model assumptions can be considered minor, so that using M_1 (instead of M_2 or M_3) is not unreasonable.

First, in the non-private setting, for any dataset generated from model $M_k, k \in \{1, 2, 3\}$ with fixed α and β, let T_k represent the probability distribution of the counts in the 100 grid cells overlaid on the confidential residual plot. Let D_k = TVD($T_k, E[T_1]$) be the total variation distance between T_k and the expected value of T_1. Using 1000 simulated datasets, we generate 1000 values of (D_1, D_2, D_3) for the specified values of α and β.

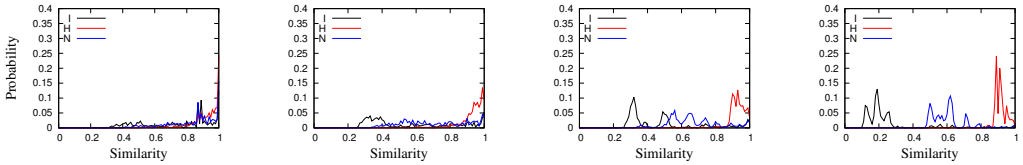
Using these 1000 draws, we find the minimum values of α and β such that the TVD(D_1, D_2) and TVD(D_1, D_3) are at least 0.95.



(a) s500,ϵ=1 (b) s1000,ϵ=1 (c) s2000,ϵ=1 (d) s5000,ϵ=1

Scale	TVD(I,H)	TVD(I,N)
500	0.959	0.902
1000	0.997	0.996
2000	1	1
5000	1	1

(e) TVD, ϵ=1



(f) s500,ϵ=0.1 (g) s1000,ϵ=0.1 (h) s2000,ϵ=0.1 (i) s5000,ϵ=0.1

Scale	TVD(I,H)	TVD(I,N)
500	0.471	0.276
1000	0.658	0.433
2000	0.857	0.676
5000	0.974	0.908

(j) TVD, ϵ=0.1

Figure 7: Comparison of similarity values for simulated confidential and private residual plots.

Non-private			Private		
n	min α	min β	$n \cdot \epsilon$	min α	min β
1000	0.0097	0.000301	1000	0.02	0.002
2000	0.0067	0.000215	2000	0.01	0.0015
5000	0.0041	0.000135	5000	0.008	0.001

Table 1: Discriminatory power of *PriRP*

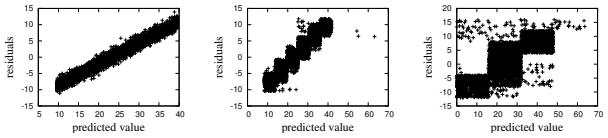
We should be able to differentiate M_1 from M_2 or M_3 in these cases. Now we repeat this process using the noisy residual plots. For any specific α and β, let $\tilde{T}_k$ be the probability distribution of the counts in the 100 grid cells overlaid on the noisy residual plot. Let $\tilde{D}_k$ = TVD($\tilde{T}_k, T_1$). Using 1000 values of $(\tilde{D}_1, \tilde{D}_2, \tilde{D}_3)$ for the specified values of α and β, we find the minimum values of α and β such that the TVD($\tilde{D}_1, \tilde{D}_2$) and TVD($\tilde{D}_1, \tilde{D}_3$) are ≥ 0.95.

Results: The first panel in Table 1 displays the minimum α and β under different values of n. As n increases, one can detect violations even at smaller degrees (i.e., smaller α and β). The second panel of Table 1 represents the minimum α and β at which residual plots output by *PriRP* can detect violations. While adding noise to the residuals makes it harder to detect violations, we still are able to detect even small violations from the linear regression assumptions when $n \cdot \epsilon \geq 1000$.

4.3.3 *Illustration with differentially private model*

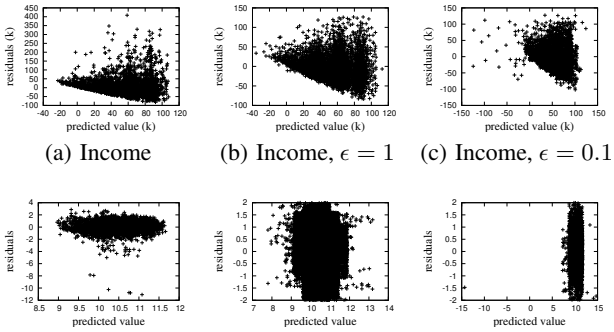
We now consider evaluating a noisy model, e.g. noisy coefficients $\tilde{b}$, obtained from a differentially private regression algorithm. There is a key difference in the interpretation of the noisy plots. The residual plots reflect lack of fit potentially from two sources. First, the posited linear model may be a poor fit to D, even in the fortunate case where $\tilde{b} \approx b$ (e.g., with large scale). Second, $\tilde{b}$ may be a poor approximation of b, even when the posited linear model would fit reasonably well if estimated on the confidential data. As such, the noisy residual plot based on $\tilde{b}$ provides an omnibus assessment of the quality of the model and the approximation $\tilde{b}$, as opposed to a focused check of the regression model assumptions.

Figure 8 shows an example of evaluating noisy model. We use the recently proposed state of the art algorithm that uses output perturbation [25] to compute the differentially private linear model coefficients $\tilde{b}$ with $\epsilon = 1$ on the "Ideal" dataset. Based on the real plots Figure 8(a), the sloped pattern indicates that the noisy coefficients $\tilde{b}$ do not accurately describe the relationship between y and $\mathbf{x}$ in "Ideal" dataset. This pattern is still evident in the noisy plots (Figure 8(b), 8(c)).



(a) Real plot on $\tilde{b}$ (b) noisy plot,ϵ=1 (c) noisy plot,ϵ=0.1

Figure 8: Residual plots based on private model $\tilde{b}$ with n = 5000



(a) Income (b) Income, $\epsilon = 1$ (c) Income, $\epsilon = 0.1$

(d) log(Income) (e) log(Income), $\epsilon = 1$ (f) log(Income), ϵ=0.1

Figure 9: Confidential and differentially private residual plots for a linear regression of income and log**(income) on several explanatory variables for 2000 Current Population Survey data.**

4.3.4 *Illustration with CPS Data*

Finally, we illustrate the use of *PriRP* on a subset of real data from the March 2000 Current Population Survey public use file. The data comprise $n = 49436$ heads of households, each with non-negative income. We seek to predict household income from age in years, education (16 levels), marital status (7 levels), and sex (2 levels). We start by fitting the OLS model of income on main effects only for each variable, using indicator variable coding for the categorical variables. As evident in Figure 9(a), the residual plot based on the confidential data reveals an obvious fan-shaped pattern, reflecting non-constant variance. This pattern is also evident in the noisy plots (Figures 9(b), 9(c)). To remedy the non-constant variance, we instead fit the regression using the natural logarithm

of income as the outcome variable. As evident in Figure 9(d), this transformation has made the constant variance assumption more believable (as it often does). The noisy residual plots also reveal the improvement in model fit (Figures 9(e), 9(f)).

5. RELATED WORK

Differentially Private Learning: Private models for regression classification has been a popular area of exploration for privacy research. Previous work has produced differentially private training algorithms for logistic and linear regression [3, 23, 25, 26, 27]. These only output the noisy model parameters. Sheffet's algorithm [23] additionally outputs confidence intervals for the model parameters. No prior work considers model diagnostics.

Private Evaluation: Receiver operating characteristic (ROC) curves are used to quantify the prediction accuracy of binary classifiers. However, directly releasing the ROC curve may reveal the sensitive information of the input dataset [13]. In this paper, we propose the first differentially private algorithm for generating private ROC curves under differential privacy. Chaudhuri et al. [4] proposes a generic technique for evaluating a classifier on a private test set. However, they assume that the global sensitivity of the evaluation algorithm is low. Hence, their work will not apply to generating ROC curves, since the sufficient statistics for generating the ROC curve (the set of true and false positive counts) have a high global sensitivity. Despite this high sensitivity, we present strategies that can privately compute ROC curves with very low noise by modeling the sufficient statistics as one-sided range queries.

The idea of releasing perturbed residual plots was first suggested by [18], who suggested that remote query systems returning regression parameter estimates also return residual plots with random noise added to the residuals from the confidential regression; see also [16]. The noise distribution did not satisfy differential privacy, or any other formal guarantee. Similar ideas were used for logistic regressions by [20].

6. CONCLUSIONS

We present differentially private versions of diagnostic tools for linear and logistic regression. Such tools provide analysts the ability to assess the quality of the assumptions and predictive capabilities of regression models in ways that are simply not possible using only the output of private algorithms or estimates based on synthetic data. The methods underlying the differentially private residual plots can be extended to other regression models.

In the future, we expect that the noisy private regression diagnostics presented here, and others like them, will comprise the output of verification servers [21], i.e., query systems that allow users to assess the quality of inferences made on synthetic data. With such outputs and systems, users can know whether or not their private data analyses result in reliable conclusions.

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant Nos. 1012141, 1131897, 1253327, 1408982, and 1443014.

7. REFERENCES

[1] T. A. Almeida, J. M. G. Hidalgo, and T. P. Silva. Towards sms spam filtering: Results under a new dataset. *International Journal of Information Security Science*, pages 1–18, 2012.

[2] K. Boyd, E. Lantz, and D. Page. Differential privacy for classifier evaluation. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pages 15–23. ACM, 2015.

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

[4] K. Chaudhuri and S. A. Vinterbo. A stability-based validation procedure for differentially private machine learning. In *Advances in Neural Information Processing Systems*, pages 2652–2660, 2013.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[6] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4):211–407, 2013.

[7] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[8] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[9] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang. Principled evaluation of differentially private algorithms using dpbench. In *SIGMOD*, 2016.

[10] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1-2):1021–1032, Sept. 2010.

[11] C. Li, M. Hay, G. Miklau, and Y. Wang. A data- and workload-aware query answering algorithm for range queries under differential privacy. *PVLDB*, 7(5):341–352, 2014.

[12] N. Li, W. Yang, and W. Qardaji. Differentially private grids for geospatial data. In *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE '13, pages 757–768, Washington, DC, USA, 2013. IEEE Computer Society.

[13] G. Matthews and O. Harel. An examination of data confidentiality and disclosure issues related to publication of empirical roc curves. *Academic radiology*, 20(7):889, 2013.

[14] F. McSherry and K. Talwar. Mechanism design via differential privacy. FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[15] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *ACM Symposium on Theory of Computing*, pages 75–84, 2007.

[16] C. M. O'Keefe and N. M. Good. Regression output from a remote analysis system. *Data & Knowledge Engineering*, 68, 2009.

[17] W. Qardaji, W. Yang, and N. Li. Understanding hierarchical methods for differentially private histograms. *Proc. VLDB Endow.*, 6(14):1954–1965, Sept. 2013.

[18] J. P. Reiter. Model diagnostics for remote access servers. *Statistics and Computing*, 13:371–380, 2003.

[19] J. P. Reiter. Releasing multiply-imputed, synthetic public use microdata: An illustration and empirical study. *Journal of the Royal Statistical Society, Series A*, 168:185–205, 2005.

[20] J. P. Reiter and C. N. Kohnen. Categorical data regression diagnostics for remote servers. *Journal of Statistical Computation and Simulation*, 75:889–903, 2005.

[21] J. P. Reiter, A. Oganian, and A. F. Karr. Verification servers: Enabling analysts to assess the quality of inferences from public use data. *Computational Statistics and Data Analysis*, 53:1475–1482, 2009.

[22] D. B. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9:462–468, 1993.

[23] O. Sheffet. Differentially private least squares: Estimation, confidence and rejecting the null hypothesis. *CoRR*, abs/1507.02482, 2015.

[24] B. Stoddard, Y. Chen, and A. Machanavajjhala. Differentially private algorithms for empirical machine learning. *CoRR*, abs/1411.5428, 2014.

[25] X. Wu, M. Fredrikson, W. Wu, S. Jha, and J. F. Naughton. Revisiting differentially private regression: Lessons from learning theory and their consequences. *CoRR*, abs/1512.06388, 2015.

[26] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett. Privgene: Differentially private model fitting using genetic algorithms. In *SIGMOD*, pages 665–676, 2013.

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