The Challenge of Differentially Private Screening Rules

Amol Khanna [1] **Fred Lu** [1 2] **Edward Raff** [1 2]

Abstract

Linear L_1-regularized models have remained one of the simplest and most effective tools in data science. Over the past decade, screening rules have risen in popularity as a way to reduce the runtime for producing the sparse regression weights of L_1 models. However, despite the increasing need of privacy-preserving models for data analysis, to the best of our knowledge, no differentially private screening rule exists. In this paper, we develop the first differentially private screening rule for linear and logistic regression. In doing so, we discover difficulties in the task of making a useful private screening rule due to the amount of noise added to ensure privacy. We provide theoretical arguments and experimental evidence that this difficulty arises from the screening step itself and not the private optimizer. Based on our results, we highlight that developing an effective private L_1 screening method is an open problem in the differential privacy literature.

1. Introduction

With the increasing number of large datasets maintained by governments and industries, there is an interest in building sparse regression algorithms which guarantee the privacy of training data. This would ensure that potentially malicious actors who gain access to a model trained on personal and private information would not be able to determine identifying information about individuals in the training dataset.

Differential privacy is a statistical technique which provides a provable guarantee of the privacy of training data when building a model. Specifically, given privacy parameters ϵ and δ and any two datasets $\mathcal{D}$ and $\mathcal{D}'$ differing on one datapoint, an approximately differentially private algorithm $\mathcal{A}$ satisfies $\mathbb{P}\left[\mathcal{A}(\mathcal{D}) \in O\right] \leq \exp\{\epsilon\}\mathbb{P}\left[\mathcal{A}(\mathcal{D}') \in O\right] + \delta$ for any $O \subseteq \text{image}(\mathcal{A})$ (Dwork et al., 2014). Intuitively, this means that if an adversary has access to the output of $\mathcal{A}$, they cannot determine whether or not a specific datapoint was used in training. Differential privacy is currently the most effective method to build statistical models on sensitive data for publication without betraying the privacy of individuals in the training data (Khanna et al., 2022; Toledo et al., 2016; Hui Yang & Zhang, 2018). Specifically, it has been shown that differential privacy can prevent dataset reconstruction attacks which are possible when models are trained in a nonprivate framework (Stock et al., 2022).

Currently, there is an interest in sparse differentially private regression algorithms. Private L_1-regularized or constrained optimizers have been developed. However, due to the addition of noise, these algorithms are either unable to maintain the sparsity of the output solution or unable to choose a good support set for the solution (Kifer et al., 2012; Talwar et al., 2015; Wang & Zhang, 2020). Other works attempt to perform private model selection, but they are computationally inefficient and assume an exact level of sparsity of the final solution in order to choose a support set prior to training (Lei et al., 2018; Thakurta & Smith, 2013).

In the nonprivate setting, sparsity on regression weights can be achieved by using screening rules. Screening rules are methods which discard features that do not contribute to a statistical model during training. They are most often used in L_1-regularized or L_1-constrained linear and logistic regression to set the coefficients of unimportant features to zero during the optimization process. In doing so, they improve the generalization of a model by counteracting overfitting and enable faster convergence. Screening rules have been used to improve the performance of sparse regression optimization on numerous datasets over the past decade and are even included the popular `R` package `glmnet` (Wang et al., 2013; 2014; Raj et al., 2016; Ghaoui et al., 2010; Olbrich, 2015; Tibshirani et al., 2012; Friedman et al., 2021).

Unlike the aforementioned approaches for private regression, screening rules do not require a predetermined support set or level of sparsity. They efficiently check a mathematical condition to determine if a feature should be screened to 0, and are implemented with sparse optimizers during training to improve the rate of learning and stability. However, to the best of our knowledge, no differentially private

[1]Booz Allen Hamilton, Annapolis Junction, MD, USA [2]University of Maryland, Baltimore County, MD, USA. Correspondence to: Edward Raff <Raff_Edward@bah.com>.

2^{nd} AdvML Frontiers workshop at 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

screening rule exists.

A differentially private screening rule has the potential to combat overfitting and help private optimizers focus on a model's most important features. However, to the best of our knowledge, no differentially private screening rule exists. In this work, we develop and explore a differentially private screening rule, and we show that it is difficult to accurately screen features. We provide an analysis of the private screening rule's behavior to explain the challenge of creating an effective differentially private screening rule.

2. Related Works

Methods to produce private sparse regression weights all suffer in performance due to the addition of noise. Private L_1 optimizers must add higher levels of noise when run for more iterations, incentivizing practitioners to run fewer iterations (Talwar et al., 2015; Wang & Zhang, 2020). However, running an optimizer for fewer iterations means that the model will be limited in its learning. On the other hand, private model selection algorithms are computationally inefficient and run prior to training, meaning they are unable to reap the benefit of any information contained within partially trained coefficients of the weight vector (Lei et al., 2018; Thakurta & Smith, 2013). Although noise is necessary for privacy, an effective private screening rule would run with a private optimizer and improve the optimizer's performance by setting the coefficients of irrelevant features to 0. By using the screening rule on the current weight vector, it can adapt to the optimizer's updates and screen features more accurately.

To develop a differentially private screening rule, we adapt Raj et al.'s rule which is flexible to solving many types of regression problems (Raj et al., 2016). While other screening rules exist, they are geometry- and problem-specific (Ghaoui et al., 2010; Wang et al., 2014; 2013). We utilize their screening rule for L_1-constrained regression to determine whether it can improve the sparsity of Talwar et al.'s L_1-constrained private Frank-Wolfe algorithm (Raj et al., 2016; Talwar et al., 2015). To the best of our knowledge, this is the first work considering a differentially private screening rule. Although our experiments find that the noise added to the screening rule overpowers its ability to screen regression weights correctly, we provide an analysis of this result to shed insight into the challenge of creating an effective private screening rule. We believe this work is valuable as it explores a novel approach to private sparsity and provides a baseline for the development and implementation of differentially private screening rules.

Since we use the private Frank-Wolfe algorithm (`DP-FW`), we also review it here. `DP-FW` uses the Frank-Wolfe method for L_1-constrained convex optimization, which chooses a vertex of the feasible region (scaled L_1 ball) which minimizes a linear approximation of the loss function. By doing this for T iterations with appropriate step sizes, the algorithm satisfies $\mathcal{L}(\mathbf{w}^{(T)}) - \min_{\mathbf{w}^* \in \mathcal{C}} \mathcal{L}(\mathbf{w}^*) \leq \mathcal{O}(\frac{1}{T})$ (Frank et al., 1956; Jaggi, 2013). To privatize the Frank-Wolfe algorithm, Talwar et al. restrict the L_∞ norm of datapoints so they can calculate the exact sensitivity of the gradients. They then use the exponential mechanism to noisily choose which component of the weight vector to update (Talwar et al., 2015).

3. Methods

Raj et al. consider the problem $\min_{\mathbf{w} \in \mathcal{C}} f(\mathbf{X}\mathbf{w})$, where $\mathcal{C}$ is the feasible set of solutions and f is L-smooth and μ-strongly convex (Raj et al., 2016). They also define $\mathbf{x}_{(i)}$ to be the i^{th} column of the design matrix $\mathbf{X}$ and $\mathcal{G}_{\mathcal{C}}(\mathbf{w})$ to be the Wolfe gap function, namely $\max_{\mathbf{z} \in \mathcal{C}} (\mathbf{X}\mathbf{w} - \mathbf{X}\mathbf{z})^\top \nabla f(\mathbf{X}\mathbf{w})$. Given this information, they prove that if

$$|\mathbf{x}_{(i)}^\top \nabla f(\mathbf{X}\mathbf{w})| + (\mathbf{X}\mathbf{w})^\top \nabla f(\mathbf{X}\mathbf{w}) + L(\|\mathbf{x}_{(i)}\|_2 + \|\mathbf{X}\mathbf{w}\|_2)\sqrt{\mathcal{G}_{\mathcal{C}}(\mathbf{w})/\mu} \tag{1}$$

is less than 0, then $w_i^* = 0$, where $\mathbf{w}^*$ is the optimal solution to the optimization problem. Our goal is to determine the sensitivity of this calculation so we can add an appropriate amount of noise and ensure screening is differentially private.

We will conduct our analysis for the case where $\|\mathbf{x}_i\|_\infty \leq 1$ and $\mathcal{C}$ is the λ-scaled L_1-ball in $\mathbb{R}^d$. These conditions are also required by Talwar et al.'s private Frank-Wolfe algorithm, which we use for L_1-constrained optimization in Section 3.3 (Talwar et al., 2015).

3.1. Sensitivity of Linear Regression

Let $\mathbf{u} = \mathbf{X}\mathbf{w}$. For linear regression, $f(\mathbf{u}) = \frac{1}{2n}(\mathbf{u} - \mathbf{y})^\top(\mathbf{u} - \mathbf{y})$, implying $\nabla f(\mathbf{u}) = \frac{1}{n}(\mathbf{u} - \mathbf{y})$ and $\nabla^2 f(\mathbf{u}) = \frac{1}{n}\mathbf{I}_n$. Therefore, from the definitions of Lipschitz smoothness and strong convexity, we can see that $f(\mathbf{u})$ is $\frac{1}{n}$-smooth and $\frac{1}{n}$-strongly convex with respect to $\mathbf{u}$.

By using the triangle inequality and the fact that the maximum of a sum is at most the sum of each element's maximum, we can bound the sensitivity of Equation 1 for linear regression by summing the sensitivity of each of its terms. These calculations are shown below. Assume without loss of generality that $\mathbf{X}$ and $\mathbf{X}'$ differ in their first row for ease of notation.

To bound the first term of Equation 1, note that

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{X}'} \left|\langle \mathbf{x}_{(i)}, \nabla f(\mathbf{X}\mathbf{w})\rangle - \left\langle \mathbf{x}'_{(i)}, \nabla f(\mathbf{X}'\mathbf{w})\right\rangle\right| \\ =&\max_{\mathbf{X},\mathbf{X}'} \left|x_{1i}\left[\frac{1}{n}\left(\mathbf{x}_1^\top \mathbf{w} - y_1\right)\right] - x'_{1i}\left[\frac{1}{n}\left(\mathbf{x'}_1^\top \mathbf{w} - y_1\right)\right]\right| \\ \leq&\frac{2\lambda}{n},\end{aligned}$$

where the first simplification comes from expanding the inner products and the second follows directly from the triangle inequality and restrictions on the feasible set and norms of the input data. This means that the sensitivity of the first term for one value of i is $\frac{2\lambda}{n}$. By the sensitivity principles of vectors, this means that releasing the values of the first term for all i has L_2-sensitivity of $\frac{2\lambda\sqrt{d}}{n}$ (Dwork et al., 2014).

For the second term,

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{X}'} |\langle \mathbf{X}\mathbf{w}, \nabla f(\mathbf{X}\mathbf{w})\rangle - \langle \mathbf{X}'\mathbf{w}, \nabla f(\mathbf{X}'\mathbf{w})\rangle| \\ =&\max_{\mathbf{X},\mathbf{X}'} \left|\mathbf{x}_1^\top \mathbf{w}\left[\frac{1}{n}\left(\mathbf{x}_1^\top \mathbf{w} - y_1\right)\right] - \right. \\ &\qquad \left. \mathbf{x'}_1^\top \mathbf{w}\left[\frac{1}{n}\left(\mathbf{x'}_1^\top \mathbf{w} - y_1\right)\right]\right| \\ \leq&\frac{2\lambda^2}{n},\end{aligned}$$

using the same logic as the previous calculation. This means its sensitivity is $\frac{2\lambda^2}{n}$ and its L_2-sensitivity is $\frac{2\lambda^2\sqrt{d}}{n}$.

To find the sensitivity of the third term, we note that the maximum of a product is bounded by the product of maximums. Given this, we find that

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{X}'} \left|\left\|\mathbf{x}_{(i)}\right\|_2 - \left\|\mathbf{x}'_{(i)}\right\|_2\right| \\ \leq&\max_{\mathbf{X},\mathbf{X}'} \sqrt{\left|x_{1i}^2 - x'^2_{1i}\right|} \\ \leq&1,\end{aligned}$$

where the first simplification is derived from expanding the first and noting that the difference of square roots must be less than or equal to the square root of the absolute value of the difference in their squared terms. The same logic can be applied for

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{X}'} \left|\|\mathbf{X}\mathbf{w}\|_2 - \|\mathbf{X}'\mathbf{w}\|_2\right| \\ \leq&\max_{\mathbf{X},\mathbf{X}'} \sqrt{\left|\left(\mathbf{x}_1^\top \mathbf{w}\right)^2 - \left(\mathbf{x'}_1^\top \mathbf{w}\right)^2\right|} \\ \leq&\lambda.\end{aligned}$$

For the Wolfe gap function,

$$\begin{aligned}&\max_{\mathbf{X},\mathbf{X}',\mathbf{z}\in\mathcal{C}} \left|\langle \mathbf{X}\mathbf{w} - \mathbf{X}\mathbf{z}, \nabla f(\mathbf{X}\mathbf{w})\rangle \right. \\ &\qquad \left. - \langle \mathbf{X}'\mathbf{w} - \mathbf{X}'\mathbf{z}, \nabla f(\mathbf{X}'\mathbf{w})\rangle\right| \\ =&\max_{\mathbf{X},\mathbf{X}',\mathbf{z}\in\mathcal{C}} \left|\mathbf{x}_1^\top(\mathbf{w}-\mathbf{z})\left[\frac{1}{n}\left(\mathbf{x}_1^\top \mathbf{w} - y_1\right)\right]\right. \\ &\qquad \left. -\mathbf{x'}_1^\top(\mathbf{w}-\mathbf{z})\left[\frac{1}{n}\left(\mathbf{x'}_1^\top \mathbf{w} - y_1\right)\right]\right| \\ \leq&\frac{4\lambda^2}{n}.\end{aligned}$$

Plugging in each of these calculations, the L_2-sensitivity of the third term is bounded by $\frac{1}{n}\left(\sqrt{d}+\lambda\sqrt{d}\right)\sqrt{\frac{4\lambda^2/n}{1/n}}$. This means the total L_2-sensitivity of the screening rule is bounded by

$$\frac{2\lambda\sqrt{d}}{n} + \frac{2\lambda^2\sqrt{d}}{n} + \frac{1}{n}\left(\sqrt{d}+\lambda\sqrt{d}\right)\sqrt{\frac{4\lambda^2/n}{1/n}}. \quad (2)$$

3.2. Sensitivity of Logistic Regression

For logistic regression, the binary cross-entropy loss is $f(\mathbf{u}) = -\frac{1}{n}\sum y_i \log\sigma(u_i) + (1-y_i)\log(1-\sigma(u_i))$, where σ is the element-wise sigmoid function. This implies $\nabla f(\mathbf{u}) = -\frac{1}{n}(\mathbf{y}-\sigma(\mathbf{u}))$ and $\nabla^2 f(\mathbf{u}) = \frac{1}{n}(\sigma(\mathbf{u})\odot(\mathbf{1}-\sigma(\mathbf{u})))\mathbf{I}_n$, where $\odot$ represents elementwise multiplication. From these equations we find that the binary cross-entropy loss is $\frac{1}{4n}$-smooth and $\frac{\sigma(\lambda)(1-\sigma(\lambda))}{n}$-strongly convex.

The derivation for the sensitivity of Equation 1 for logistic regression follows the same steps as used for linear regression in Section 3.1, so we do not show the full calculations here. The final L_2-sensitivity bound derived is

$$\begin{aligned}&\frac{2\sigma(\lambda)\sqrt{d}}{n} + \frac{2\lambda\sigma(\lambda)\sqrt{d}}{n} \\ &+\frac{1}{4n}\left(\sqrt{d}+\lambda\sqrt{d}\right)\sqrt{\frac{4\lambda\sigma(\lambda)/n}{\sigma(\lambda)(1-\sigma(\lambda))/n}}.\end{aligned} \quad (3)$$

3.3. Implementing Private Screening

Now that we have calculated the bounds for the L_2-sensitivity of the screening rule in Equation 1, we discuss how we implemented it into a differentially private regression training procedure.

Since our screening rule requires L_1-constrained optimization, we employ the private Frank-Wolfe algorithm developed by Talwar et al. to train regression models (Talwar et al., 2015). To the best of our knowledge, this is the only differentially-private algorithm for L_1-constrained optimization. Additionally, both linear and logistic loss have Lipschitz constant 1 with respect to the L_1 norm, which

Algorithm 1 L_1-Constrained Regression with Screening

Require: Privacy Parameters: $\epsilon_1 > 0$, $\epsilon_2 > 0$, $0 < \delta_1 \leq 1$, $0 < \delta_2 \leq 1$; Constraint: $\lambda > 0$; Iterations: T; Design Matrix: $\mathbf{X} \in \mathbb{R}^{n \times d}$ where $\|\mathbf{x}_i\|_\infty \leq 1$ for all $i \in \{1, \ldots, n\}$; Target: $\mathbf{y}$; L_2-Sensitivity: s; Iterations to Screen: $\mathbf{i}$.

$l \leftarrow \texttt{Length}(\mathbf{i})$
$\delta_{\text{iter}} \leftarrow \frac{\delta_2}{l+1}$
$\epsilon_{\text{iter}} \leftarrow \frac{\epsilon_2}{2\sqrt{2l\log(1/\delta_{\text{iter}})}}$
$\sigma^2 \leftarrow \frac{2s^2\log(1.25/\delta_{\text{iter}})}{\epsilon^2_{\text{iter}}}$
$\widehat{\mathbf{w}}^{(0)} \leftarrow$ Random Vector in the λ-scaled L_1 Ball
for $t = 1$ to T **do**
  $\widehat{\mathbf{w}}^{(t)} \leftarrow \texttt{DP-FW Step}\left(\epsilon_1, \delta_1, \lambda, T, \mathbf{X}, \mathbf{y}, \widehat{\mathbf{w}}^{(t-1)}\right)$
  if $t \in \mathbf{i}$ **then**
    $screen \leftarrow \texttt{Equation 1}\left(\mathbf{X}, \mathbf{y}, \widehat{\mathbf{w}}^{(t)}, \lambda\right)$
    $screen \leftarrow screen + \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$
    $\widehat{\mathbf{w}}^{(t)}_j \leftarrow 0$ **if** $screen_j < 0$ **for all** j
  end if
end for
Output $\widehat{\mathbf{w}}^{(T)}$

satisfies the algorithm's requirement for L_1-Lipschitz loss functions.

Our algorithm is shown in Algorithm 1, abstracting away the steps required for the private Frank-Wolfe algorithm. By using the Gaussian mechanism and the advanced composition theorem for approximate differential privacy, the screening technique in Algorithm 1 is (ϵ_2, δ_2)-differentially private (Dwork et al., 2006; 2010). Following this, the basic composition theorem of approximate differential privacy guarantees that the results of Algorithm 1 is $(\epsilon_1+\epsilon_2, \delta_1+\delta_2)$-differentially private (Dwork et al., 2006).

4. Experiments

To test whether Algorithm 1 performs well in practice, we tested how it performed on linear and logistic regression. To do so, we used the synthetic dataset which Raj et al. used to test their nonprivate screening algorithm (Raj et al., 2016). Specifically, we generated 3000 datapoints in $\mathbb{R}^{600}$ from the standard normal distribution, and scaled the final dataset so $\|\mathbf{x}_i\|_\infty \leq 1$. We set the true weight vector $\mathbf{w}^*$ to be sparse with 35 entries of $+1$ and 35 entries of -1. For linear regression, $\mathbf{y} = \mathbf{X}\mathbf{w}^*$, and for logistic regression, $\mathbf{y} = \mathbb{1}_{\mathbf{X}\mathbf{w}^* > 0}$. Raj et al. demonstrated that the nonprivate screening rule listed in Equation 1 performs well on this dataset for linear regression (Raj et al., 2016). We verified this result, finding that using the nonprivate Frank-Wolfe optimizer with the nonprivate screening rule at every iteration produced a final weight vector in which nonzero components were only at the locations of nonzero components in the true weight vector

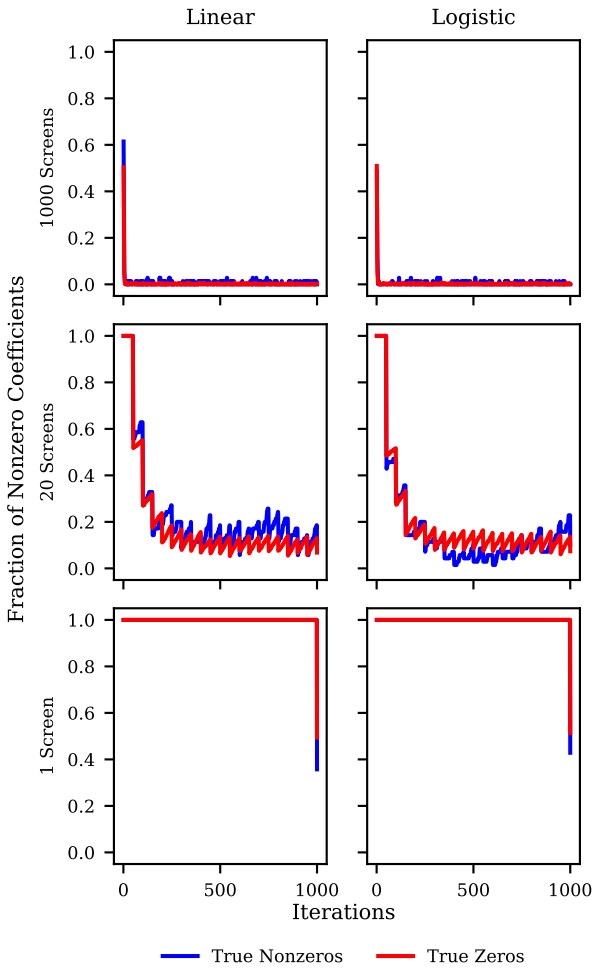

Figure 1. Testing Algorithm 1 on a synthetic dataset.

and 55% of the true nonzero components were nonzero after training in both linear and logistic regression. We then ran the private Algorithm 1 on this dataset for 1000 iterations with $\epsilon_1 = \epsilon_2 = 2.5, \delta_1 = \delta_2 = \frac{1}{6000}$, and $\lambda = 5$.

Figure 1 shows the results of this experiment when we implemented screening after every iteration, every 50^{th} iteration, and after the last iteration. It is clear that Algorithm 1 is not able to discriminate between screening true zero and true nonzero coefficients in any of these cases. Additionally, when private screening is implemented too often, it screens too many coefficients, and since the Frank-Wolfe algorithm only updates one coefficient at a time, after a few iterations, the weight vector is only able to have up to a few nonzero coefficients which have not been screened to zero. To identify whether the private Frank-Wolfe algorithm or the private screening methods were causing these poor results, we ran the following two experiments:

(A) We tested how well nonprivate screening performed using the private Frank-Wolfe algorithm with $\epsilon = 2.5$

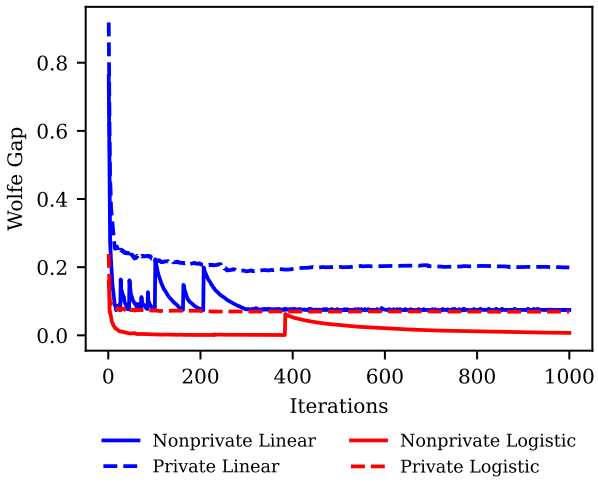

Figure 2. Comparing the Wolfe gap function for nonprivate and private optimization when nonprivate screening is applied at every iteration.

and $\delta = \frac{1}{6000}$. We found that no matter how often we implemented the screening rule, no coefficients were screened from the solution.

(B) We tested how well private screening performed using the nonprivate Frank-Wolfe algorithm with $\epsilon = 2.5$ and $\delta = \frac{1}{6000}$. We found that when screening every 50^{th} iteration, the screening rule would produce a final weight vector with nonzero components in approximately 10% of the true nonzero components and none of the true zero components. The results of this experiment when screening every iteration or only at the last iteration mimicked those found in the respective rows of Figure 1.

These experiments provide key insights into the results shown in Figure 1. Experiment A suggests that all of the screening occurring in Figure 1 arises from noise added to the screening rule. This is because without the screening's noise, no screening occurs. It also implies that the noise added for private optimization made it more difficult to screen coefficients, since when we tested completely nonprivate screening (without noisy optimization), the nonprivate screening rule worked well. Heuristically, this outcome may arise because noisy weights make the Wolfe gap function in Equation 1 very large, meaning that it overpowers the second term, which is the only term that can be negative and is thus essential to effective screening. We verified that the Wolfe gap function evaluates to smaller values when using nonprivate optimization for both linear and logistic regression. This result can be seen in Figure 2.

Experiment B indicates that the noise added in the private screening rule makes it much stronger that its nonprivate counterpart. This is observed by noting that without a noisy screening rule, screening at every iteration with the nonprivate Frank-Wolfe optimizer would not screen all the true nonzero components to zero, whereas with the noisy screening rule, almost all components are screened to zero after only a few iterations.

5. Discussion

The goal of a differentially private screening rule is to improve sparse private optimization during training. The screening rule described in Algorithm 1 is computationally efficient and able to benefit from the information contained in partially trained coefficients, unlike private model selection algorithms in prior works. However, the results in Figure 1 indicate that it is not effective at screening only true zero coefficients. Given the sensitivities derived in Equation 2 and Equation 3 and the results in Section 4, we use this section to discuss the major challenge to developing an effective differentially private screening rule.

The sensitivities of linear and logistic regression are on the order of $\sqrt{d}$, so a private screening rule would have to add noise with a scale of $\mathcal{O}\left(\sqrt{d}\right)$. This is a property of private sensitivity, and noise with a scale of $\mathcal{O}\left(\sqrt{d}\right)$ is also found in private optimization and private model selection algorithms. Our results clearly show this noise level is too large. We also note the sample complexity of nonprivate L_1 models is known to grow at a $\mathcal{O}(\log d)$ rate, seemingly implying that our screening model may not work well asymptotically.

Though discouraging, our results leave a number of open questions now that we have identified the difficulty of private screening rules. 1) Can differentailly private screening rules be effective in the finite values of d that occur in practice? 2) Can the noise added by screening be reduced to a rate of $\mathcal{O}(\log d)$? 3) How can the concept of a "safe" screening rule be adapted for differential privacy, since no screening rule can avoid false-positives when noise is added?

6. Conclusion

In this paper, we are the first to consider differentially private screening rules. We attempt to develop such a rule by modifying a general-purpose nonprivate screening rule, but when testing our algorithm on a synthetic dataset, we find that noisy optimization and screening produces poor performance. By analyzing how different sources of noise affect the screening rule's behavior, we identify the limitations to our algorithm. We conclude by discussing the challenges to developing a useful private screening rule. We highlight that developing an effective differentially private screening rule is an open problem with the potential to improve the efficiency and accuracy of high dimensional private regression.

References

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pp. 486–503. Springer, 2006.

Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60. IEEE, 2010.

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.

Frank, M., Wolfe, P., et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., and Qian, J. Package 'glmnet'. *CRAN R Repositary*, 2021.

Ghaoui, L. E., Viallon, V., and Rabbani, T. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*, 2010.

Hui Yang, G. and Zhang, S. Differential privacy for information retrieval. WSDM '18, pp. 777–778, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355810.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pp. 427–435. PMLR, 2013.

Khanna, A., Schaffer, V., Gürsoy, G., and Gerstein, M. Privacy-preserving model training for disease prediction using federated learning with differential privacy. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1358–1361. IEEE, 2022.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1. JMLR Workshop and Conference Proceedings, 2012.

Lei, J., Charest, A.-S., Slavkovic, A., Smith, A., and Fienberg, S. Differentially private model selection with penalized and constrained likelihood. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181 (3):609–633, 2018.

Olbrich, J. Screening rules for convex problems. Master's thesis, ETH-Zürich, 2015.

Raj, A., Olbrich, J., Gärtner, B., Schölkopf, B., and Jaggi, M. Screening rules for convex problems. *arXiv preprint arXiv:1609.07478*, 2016.

Stock, P., Shilov, I., Mironov, I., and Sablayrolles, A. Defending against reconstruction attacks with r\'enyi differential privacy. *arXiv preprint arXiv:2202.07623*, 2022.

Talwar, K., Guha Thakurta, A., and Zhang, L. Nearly optimal private lasso. *Advances in Neural Information Processing Systems*, 28, 2015.

Thakurta, A. G. and Smith, A. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pp. 819–850. PMLR, 2013.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and Tibshirani, R. J. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74 (2):245–266, 2012.

Toledo, R. R., Danezis, G., and Goldberg, I. Lower-cost ϵ-private information retrieval. *Proceedings on Privacy Enhancing Technologies*, 2016:184 – 201, 2016.

Wang, J., Zhou, J., Wonka, P., and Ye, J. Lasso screening rules via dual polytope projection. *Advances in neural information processing systems*, 26, 2013.

Wang, J., Zhou, J., Liu, J., Wonka, P., and Ye, J. A safe screening rule for sparse logistic regression. *Advances in neural information processing systems*, 27, 2014.

Wang, P. and Zhang, H. Differential privacy for sparse classification learning. *Neurocomputing*, 375:91–101, 2020.