Differential Privacy via DRO: GitHub Appendices

Aras Selvi*[1], Huikang Liu[2] and Wolfram Wiesemann[1]

[1]Imperial College Business School, Imperial College London, United Kingdom
[2]Research Institute for Interdisciplinary Sciences, School of Information Management and Engineering,
Shanghai University of Finance and Economics, China

May 1, 2023

Suboptimalities of Several Mainstream Assumptions

In the main paper we claimed that several mainstream assumptions taken in the literature, which we do not take, are **not** without loss of generality. Indeed, we next show some counterexamples of each popular assumption on optimal noise distributions: *(i)* they are monotone around the origin; *(ii)* they are symmetric around the origin; *(iii)* they come from a certain family of distributions.

Monotone Distributions

To prove that monotone distributions are not always optimal, it is sufficient to give one counter example. To this end, we take $\varepsilon = 3.0$ and $\delta = 0.3$ and optimize the ℓ_1-loss over a sufficiently large support. While the optimal distribution achieves a loss of 0.1705, the monotonicity-constrained optimization problem gives a lower bound of 0.1830 for a discretization granularity of $\beta = 0.02$. In other words, the optimal monotone distribution cannot achieve a loss better than 0.1830 for any granularity and support, whereas a non-monotone distribution achieves a better loss already for $\beta = 0.02$.

Symmetric Distributions

We optimize symmetric ($c(x) = |x|$) and asymmetric ($c(x) = |x| + \mathbb{1}[x > 0] \cdot |x|$) loss functions and share the result in Figure 1. One can observe that while the distribution minimizing the former loss function is symmetric, the distribution minimizing the latter loss function is not symmetric around origin or any pother point. We note that increasing the asymmetry of loss functions further (*e.g.*, increasing the slope of $x > 0$), or increasing the privacy regime (that makes the optimal distributions use larger supports, hence incurring larger losses when $x > 0$) makes the asymmetry of the optimal distributions more severe.

[*] Corresponding author: a.selvi19@imperial.ac.uk

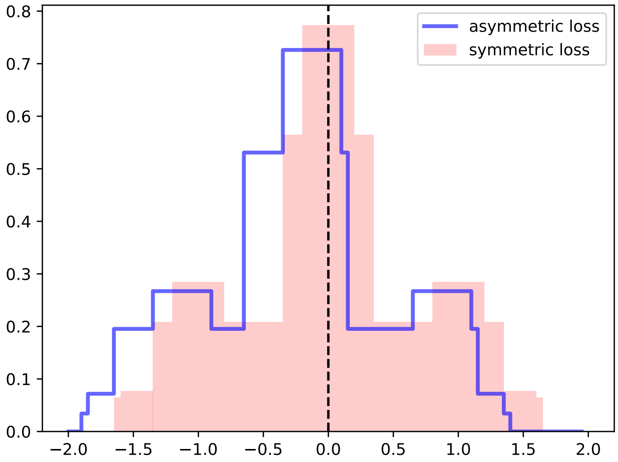

Figure 1: *Optimization-based noise distributions for synthetic data independent instances with $\Delta f = 1$, $\beta = 0.05$, $\varepsilon = 1$, $\delta = 0.2$, and two loss functions. The distribution minimizing the symmetric loss ($c(x) = |x|$) is shown in red shading, whereas the distribution minimizing the asymmetric loss ($c(x) = |x| + \mathbb{1}[x > 0] \cdot |x|$) is shown as blue lines.*

Restriction to a Family of Distributions

Finally, we give counterexamples on restricting the feasible noise distributions to a specific family. Although the previous counterexample on asymmetric loss functions would be sufficient for this purpose, we show a stronger result: even two different symmetric loss functions would yield the optimal solutions looks significantly different. In Figure 2, we observe that optimizing ℓ_1- and ℓ_2-losses result in distributions that could not belong to the same family of distributions, that is, there is no trivial density function that would generalize these distributions.

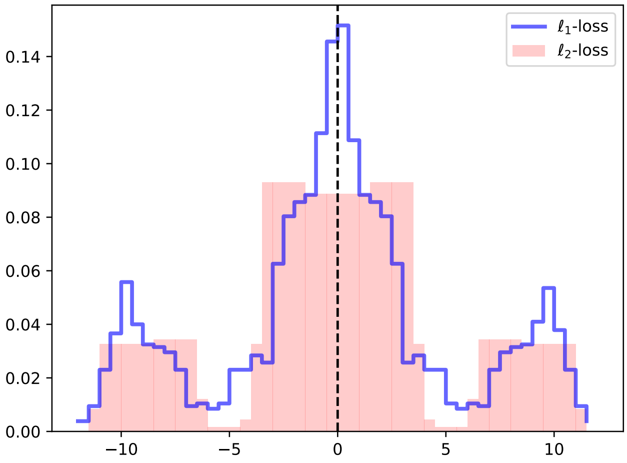

Figure 2: *Optimization-based noise distributions for synthetic data independent instances with $\Delta f = 10$, $\beta = 0.5$, $\varepsilon = 1$, $\delta = 0.4$, and two loss functions. The distribution minimizing the ℓ_1-loss ($c(x) = |x|$) is shown in red shading, whereas the distribution minimizing the ℓ_2-loss ($c(x) = x^2$) is shown as blue lines.*

Sampling Noise from Various Distributions

For the experiments on differentially private Naïve Bayes and proximal coordinate descent, we sample noise from various probability distributions to ensure privacy. Some distributions are easy to sample from by using the *Distributions* package of Julia, including the Laplace and Gaussian distributions. Here we give details on how to sample noise from the optimized distribution as well as the truncated Laplace distribution.

Optimized distribution.

Recall that, for data independent noise optimization, we solve an upper bound problem to obtain the mixture weights $\{p(j)\}_{j=1}^N$ of a mixture of uniform distributions. Recall that the probability of the noise being sampled from the j-th interval $\Pi_j(\beta)$ is $p(j)$. Hence, we first sample the interval from a discrete distribution with probabilities $\{p(j)\}_{j=1}^N$. Then, we sample the noise from a uniform distribution supported on $\Pi_j(\beta)$. Extension of this method to the data dependent setting is straightforward because although we have multiple optimized distributions, we sample noise from only the distribution corresponding to the true query value.

Truncated Laplace distribution.

For given $\delta \in (0, 0.5)$, $\varepsilon > 0$, $\Delta f > 0$, the Truncated Laplace distribution is defined by the probability density function:

$$f_{\text{TLap}}(x) = \begin{cases} B \cdot e^{\frac{-|x|}{\lambda}} & \text{for } x \in [-A, A] \\ 0, & \text{otherwise} \end{cases}$$

where $\lambda := \dfrac{\Delta f}{\varepsilon}$, $A := \dfrac{\Delta f}{\varepsilon} \cdot \log\left(1 + \dfrac{e^{\varepsilon} - 1}{2 \cdot \delta}\right)$, $B := \dfrac{1}{2 \cdot \lambda \cdot (1 - e^{-\frac{A}{\lambda}})}$. To sample noise from this distribution, we derive the inverse cumulative distribution function. To this end, we first derive the cumulative distribution function of f_{TLap}, which, after using some algebraic manipulations, can be expressed as:

$$F_{\text{TLap}}(x) = \begin{cases} 0 & \text{for } x \leq -A \\ 1 & \text{for } x \geq A \\ \dfrac{1}{2} - \text{sign}(x) \cdot \left[-\dfrac{1}{2} + B \cdot \lambda \cdot \exp(-|x|/\lambda) - B \cdot \lambda \cdot \exp(-A/\lambda)\right] & \text{for } x \in [-A, A]. \end{cases}$$

The inverse of this function can be obtained from the equation $F_{\text{TLap}}(F_{\text{TLap}}^{-1}(u)) = u$:

$$\frac{1}{2} - \underbrace{\text{sign}(F_{\text{TLap}}^{-1}(u))}_{=\text{sign}(u-1/2)} \left[-\frac{1}{2} + B \cdot \lambda \cdot \exp\left(\frac{-|F_{\text{TLap}}^{-1}(u)|}{\lambda}\right) - B \cdot \lambda \cdot \exp\left(\frac{-A}{\lambda}\right)\right] = u$$

$$\iff -\frac{1}{2} + B \cdot \lambda \cdot \exp\left(\frac{-|F_{\text{TLap}}^{-1}(u)|}{\lambda}\right) - B \cdot \lambda \cdot \exp\left(\frac{-A}{\lambda}\right) = \underbrace{\frac{u - 1/2}{-\text{sign}(u - 1/2)}}_{=-|u-1/2|}$$

$$\iff B \cdot \lambda \cdot \exp\left(\frac{-|F_{\text{TLap}}^{-1}(u)|}{\lambda}\right) - B \cdot \lambda \cdot \exp\left(\frac{-A}{\lambda}\right) = \underbrace{-|u - 1/2| + \frac{1}{2}}_{=\min\{u, 1-u\}}$$

$$\begin{aligned}
&\iff \exp\left(\frac{-|F_{\text{TLap}}^{-1}(u)|}{\lambda}\right) = \frac{\min\{u, 1-u\}}{B \cdot \lambda} + \exp\left(\frac{-A}{\lambda}\right) \\
&\iff |F_{\text{TLap}}^{-1}(u)| = -\lambda \cdot \log \cdot \left[\frac{\min\{u, 1-u\}}{B \cdot \lambda} + \exp\left(\frac{-A}{\lambda}\right)\right] \\
&\iff F_{\text{TLap}}^{-1}(u) = -\text{sign}(u - 0.5) \cdot \lambda \cdot \log\left[\frac{\min\{u, 1-u\}}{B \cdot \lambda} + \exp\left(\frac{-A}{\lambda}\right)\right].
\end{aligned}$$

We then sample $u \sim [0,1]$ uniformly at random, and compute $F_{\text{TLap}}^{-1}(u)$ to obtain a sample from the Truncated Laplace distribution.

Instance Optimality Guarantees

In the main paper, we discussed an advantage of the optimization approach to DP: we can add arbitrary constraints on the optimal distributions as long as they are tractable. One example is *instance optimality* in data dependent noise optimization. Recall that in data dependent noise optimization we minimize

$$\int_{\phi \in \Phi} w(\phi) \cdot \left[\int_{x \in \mathbb{R}} c(x) \, \mathrm{d}\gamma(x \mid \phi)\right] \mathrm{d}\phi.$$

In the numerical experiments, we observed that the optimal value of this objective (let this value be $o^\star$), is significantly smaller than the optimal value of the data independent noise optimization (let this value be o'); however, there are instances ϕ where $\int_{x \in \mathbb{R}} c(x) \, \mathrm{d}\gamma(x \mid \phi) > o'$ at optimality. In other words, although the (weighted) average of losses attained by each ϕ is significantly low, there are instances whose losses are larger than what we would have obtained in the data independent noise optimization setting. Thus, we added constraints on each instance as $\int_{x \in \mathbb{R}} c(x) \, \mathrm{d}\gamma(x \mid \phi) \leq o$, for some feasible o (typically o'), which ensured that none of the instances will have a loss more than o, while still minimizing the objective function. These constraints can use any other loss function, and they do not need to coincide with the c in the objective function.

Partitioning Φ for Proximal Coordinate Descent

Recall that we do not need to partition Φ with the uniform length intervals $\{\Phi_k(\beta)\}_{k \in [K]}$; this was only taken for the ease of exposition. We can instead use a non-uniform partitioning and here we give one example for the proximal coordinate descent method.

The term we are adding noise to, which we will refer as *the query*, in the proximal coordinate descent algorithm (l-th coordinate of the sum of gradients) is

$$\sum_{i=1}^{n} \frac{\exp(-y^i \cdot \boldsymbol{h}^{t,k\top} \boldsymbol{x}^i)}{1 + \exp(-y^i \cdot \boldsymbol{h}^{t,k\top} \boldsymbol{x}^i)} \cdot (-y^i \cdot x_l^i) \in (-n, n),$$

hence we have $\Phi = (-n, n)$. If the number of instances n in the training set is large, then, we cannot hope to have the uniform partitioning $\{\Phi_k(\beta)\}_{k \in [K]}$ with small $\beta > 0$. However, interestingly, we observe that this term is rarely close to $\pm n$ throughout the iterations of the proximal coordinate descent, and most of the values accumulate around zero. This is evidence for us to take a fine partition around the origin, and a coarse partition on the tails of Φ.

To give an example, consider the cylinder-bands dataset, which has 432 instances in its training set after any 80% training set split. Although this implies that $\Phi = (-432, 432)$, Figure 3 shows us that most realizations of the query are around the origin, and they become rarer further away from the origin. More than 80% of the realizations are in the range $(-10, 10)$; we thus take a partition as

$$\Phi_1 = [-432, -10),\ \Phi_2 = [-10, -9.5),\ \Phi_3 = [-9.5, -9), \ldots, \Phi_{41} = [9.5, 10),\ \Phi_{42} = [10, 432).$$

This helps us use fewer distributions (a uniform partition with 0.5 increments would give us more than $2,000$ distributions). By using our intuition, instead of a uniform weight w, one might further revise it so that more importance is given to the intervals that are closer to 0.

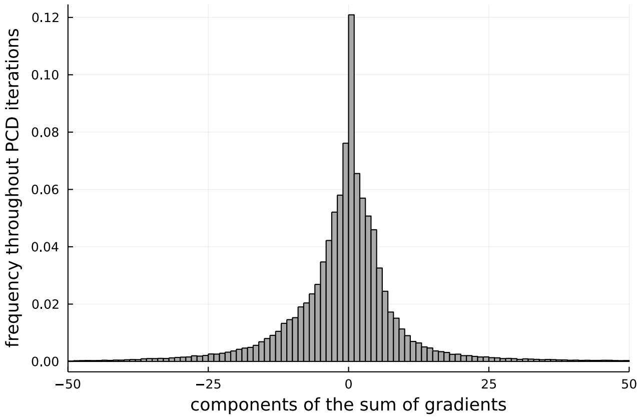

Figure 3: Histogram of the true query answers throughout 100 simulations of a PCD training on the cylinder-bands dataset where each simulation evaluates the query $K \cdot T$ times.

Extension of Algorithm 3 to Non-Uniform Partitions of Φ

To obtain a non-uniform partitioning, it is sufficient to impose equality constraints on some consecutive maps p_k. For example, if we want to take an interval that contains both $\Phi_1(\beta)$ and $\Phi_2(\beta)$, we can impose $p_1 = p_2$. We now formally introduce this setting and revise Algorithm 3 accordingly.

Let $\boldsymbol{\lambda} \in \{1, \ldots, K+1\}^{M+1}$ be an index vector satisfying

$$1 = \lambda_1 \ < \ \lambda_2 \ < \ \ldots \ < \ \lambda_M \ < \ \lambda_{M+1} = K+1,$$

and let $\Lambda_j = \{\lambda_j, \ldots, \lambda_{j+1} - 1\}$, $j \in [M]$, denote the set of intervals that will be grouped together so to give us the j-th new interval. Consider a variant of $\mathrm{P}'(\boldsymbol{\pi}, \beta)$ that enforces equalities $p_k = p_{k'}$ for all k, k' that satisfy $k, k' \in \Lambda_j$ for some $j \in [M]$. Then, given some fixed $l, l' \in [M]$, for any

$k \in \Lambda_l$, $m \in \Lambda_{l'}$ and $(\varphi, A) \in \mathcal{E}'_{km}(L, \beta)$, the privacy shortfall can be expressed as

$$\begin{aligned}
&\sum_{j \in [N]} p_k(j) \cdot \frac{|A \cap \Pi_j(\beta)|}{|\Pi_j(\beta)|} - e^{\varepsilon} \cdot \sum_{j \in [N]} p_m(j) \cdot \frac{|A \cap (\Pi_j(\beta) + \varphi)|}{|\Pi_j(\beta)|} \\
=& \sum_{i \in [\pm L]} \mathbb{1}[I_i(\beta) \subseteq A] \cdot \beta \cdot \left[\sum_{j \in [N]} p_k(j) \cdot \frac{\mathbb{1}[I_i(\beta) \subseteq \Pi_j(\beta)]}{|\Pi_j(\beta)|} - e^{\varepsilon} \cdot \sum_{j \in [N]} p_m(j) \cdot \frac{\mathbb{1}[I_i(\beta) \subseteq (\Pi_j(\beta) + \varphi)]}{|\Pi_j(\beta)|} \right] \\
=& \sum_{i \in [\pm L]} \mathbb{1}[I_i(\beta) \subseteq A] \cdot \beta \cdot \left[\sum_{j \in [N]} p_{\lambda_l}(j) \cdot \frac{\mathbb{1}[I_i(\beta) \subseteq \Pi_j(\beta)]}{|\Pi_j(\beta)|} - e^{\varepsilon} \cdot \sum_{j \in [N]} p_{\lambda_{l'}}(j) \cdot \frac{\mathbb{1}[I_i(\beta) \subseteq (\Pi_j(\beta) + \varphi)]}{|\Pi_j(\beta)|} \right],
\end{aligned}$$

where the last inequality holds because $k \in \Lambda_l$ and $m \in \Lambda_{l'}$ imply $p_k = p_{\lambda_l}$ and $p_m = p_{\lambda_{l'}}$, respectively. We already proved in the main paper that for a specific pair (k, m) the privacy violation will be violated for some $\varphi \in \{(m-k-1) \cdot \beta, (m-k) \cdot \beta, (m-k+1) \cdot \beta\}$, and here we show that the privacy violation only depends on l and l' so that $k \in \Lambda_l$ and $m \in \Lambda_{l'}$. This implies that, for any $l, l' \in [M]$ we can search for the φ maximizing the privacy shortfall in:

$$\begin{aligned}
&\bigcup \{\{(m-k-1) \cdot \beta, (m-k) \cdot \beta, (m-k+1) \cdot \beta\} \ : \ k, m \in [K], \ k \in \Lambda_l, \ m \in \Lambda_{l'}\} \\
=&\{(\lambda_{l'} - \lambda_{l+1}) \cdot \beta, (\lambda_{l'} - \lambda_{l+1}) \cdot \beta + \beta, \ldots, (\lambda_{l'+1} - \lambda_l) \cdot \beta\}.
\end{aligned}$$

Similar to Lemma C.17, for each such φ we can greedily construct the worst-case events $A^\star$. Thus, we have Algorithm G.1.

Proposition G.1. *The total complexity of Algorithm G.1 is $\mathcal{O}(M^2 \cdot N^3)$.*

Algorithm G.1: *Identification of a constraint in* $\mathrm{P}'(\boldsymbol{\pi}, \beta)$ *with maximum privacy shortfall*

```
input  : $\boldsymbol{\pi}$, $\beta$, $p$
output: constraint $(\varphi^\star, A^\star)$ with maximum privacy shortfall $V(\varphi^\star, A^\star)$
Initialize $V^\star = 0$;
for $l, l' \in [M]$ do
    for $\varphi \in \{(\lambda_{l'} - \lambda_{l+1}) \cdot \beta, (\lambda_{l'} - \lambda_{l+1}) \cdot \beta + \beta, \ldots, (\lambda_{l'+1} - \lambda_l) \cdot \beta\} \cap [-\Delta f, \Delta f]$ do
        Initialize $A = \emptyset$ and $V = 0$;
        for $j = 1, \ldots, N$ do
            Let $A_j = \Pi_j(\beta) \setminus [-L \cdot \beta + \varphi, (L+1) \cdot \beta + \varphi)$ and update
                $$A = A \cup A_j, \quad V = V + |A_j| \cdot \frac{p_{\lambda_l}(j)}{|\Pi_j(\beta)|}.$$
            for $j' = 1, \ldots, N$ do
                if $p_{\lambda_l}(j)/|\Pi_j(\beta)| > e^\varepsilon \cdot p_{\lambda_{l'}}(j')/|\Pi_{j'}(\beta)|$ then
                    Let $A_{jj'} = \Pi_j(\beta) \cap (\Pi_{j'}(\beta) + \varphi)$ and update
                        $$A = A \cup A_{jj'}, \quad V = V + |A_{jj'}| \cdot \left[\frac{p_{\lambda_l}(j)}{|\Pi_j(\beta)|} - e^\varepsilon \cdot \frac{p_{\lambda_{l'}}(j')}{|\Pi_{j'}(\beta)|}\right].$$
                end
            end
        end
    end
    if $V > V^\star$ then
        Update $\varphi^\star = \varphi$, $A^\star = A$ and $V^\star = V$.
    end
end
return $(\varphi^\star, A^\star)$ and $V^\star(\varphi, A) = V^\star - \delta$.
```