Differential Privacy via DRO: GitHub Appendices

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

[1]Imperial College Business School, United Kingdom
[2]Shanghai University of Finance and Economics, China

May 24, 2024

More details on Example 1

Example 1 (cont'd)**.** *Recall that, for the case of $\delta = 0$, the DP constraints required that the adversary should never be confident in her maximum likelihood estimation, because they ensure $\log(\mathbb{P}[f(D_{-n}, d) + \tilde{X} \in \{O\}] / \mathbb{P}[f(D_{-n}, d') + \tilde{X} \in \{O\}]) \leq \varepsilon$ for every $d, d' \in \mathcal{D}$. The quantity $\log(\mathbb{P}[f(D_{-n}, d) + \tilde{X} \in \{O\}] / \mathbb{P}[f(D_{-n}, d') + \tilde{X} \in \{O\}])$, that needs to be bounded by $\pm\varepsilon$ for all $d, d' \in \mathbb{U}$ and all $O \in \mathbb{R}$ is named the* privacy loss *(Dwork and Rothblum 2016). We visualize how the Laplace mechanism ensures privacy loss is always bounded to the desired range if we set its scale carefully. To this end, fix d, d' where $d = D_n$ and d' is an instance with the minimum salary $120k$ INR, and consider the corresponding privacy loss as a function of O (i.e., the adversary is comparing the likelihood of the true D_n value being d or d' upon seeing O). Figure 1 (left) shows how this privacy loss can be bounded in $[-\varepsilon, \varepsilon]$ by increasing the scale of the underlying Laplace mechanism that gave O. The low scale mechanism falls outside this range for some outputs and hence does not satisfy DP. The medium scale ($\frac{170k-120k}{194}\varepsilon^{-1}$ INR) mechanism, on the other hand, bounds the privacy loss to the desired range. However, this mechanism does not satisfy DP either, because this plot is specific to the privacy loss between a* fixed *pair of $D = (D_{-n}, d)$ (salary database) and its neighbor $D' = (D_{-n}, d')$ for which the query differs by $\frac{170k-120k}{194}$ INR; but DP bounds the privacy loss for* any *pair $(D, D') \in \mathcal{N}$. Since the worst case query difference is $\frac{190k-120k}{194}$ INR, the high scale ($\frac{190k-120k}{194}\varepsilon^{-1}$ INR) mechanism, whose scale equals exactly $\Delta f / \varepsilon$, is feasible. The case of $\delta > 0$, on the other hand, allows for violations of bounded privacy loss for negligible events even when the scale is taken correctly. Figure 1 (right) shows how Gaussian mechanisms for (ε, δ)-DP with varying δ attain bounded privacy losses in different ranges.*

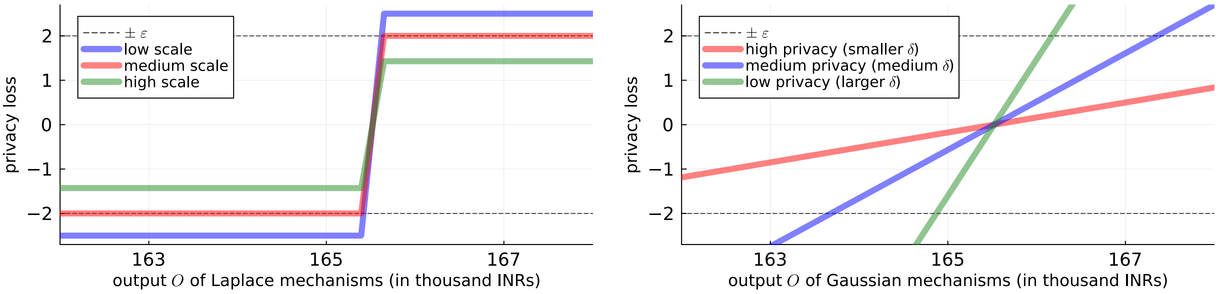

Figure 1: *The privacy loss between the salary database D of Example 1 and (D_{-n}, d') upon sharing O, where d' is an instance with salary $120k$ INR. Laplace mechanisms with varying scale (left) and Gaussian mechanisms for (ε, δ)-DP with varying δ (right) are compared.*

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

Reconstructing Tables 4 and 5 for the ℓ_2-loss

We revise Table 4 (upper bound suboptimality of the best benchmark for ℓ_1-loss) and Table 5 (upper bound suboptimality of the best benchmark for ℓ_1-loss) for ℓ_2-loss in Tables 1 and 2.

		δ									
		0.005	**0.010**	**0.020**	**0.050**	**0.100**	**0.200**	**0.250**	**0.300**	**0.500**	**0.750**
ε	**0.005**	0.22%	0.17%	0.08%	0.39%	0.77%	3.41%	4.58%	2.61%	3.35%	1.44%
	0.010	0.58%	0.46%	0.85%	0.81%	0.80%	3.43%	4.61%	2.66%	3.38%	1.46%
	0.020	0.38%	0.58%	0.11%	2.09%	3.26%	3.49%	4.68%	2.77%	3.44%	1.52%
	0.050	0.64%	0.36%	0.53%	2.45%	3.55%	3.82%	3.09%	3.14%	3.64%	1.69%
	0.100	0.46%	0.57%	0.77%	2.96%	4.03%	2.75%	3.51%	3.05%	3.93%	2.09%
	0.200	0.91%	1.05%	1.29%	4.08%	3.02%	2.94%	4.08%	4.74%	4.44%	2.61%
	0.500	1.96%	2.15%	2.48%	7.88%	5.85%	5.74%	6.28%	5.60%	5.22%	3.89%
	1.000	5.26%	5.52%	5.99%	10.36%	9.45%	7.84%	6.85%	5.90%	5.43%	5.90%
	2.000	7.65%	7.65%	7.48%	9.16%	8.38%	6.33%	7.36%	5.91%	7.77%	8.82%
	5.000	5.03%	5.03%	5.03%	5.25%	5.28%	5.37%	5.43%	5.48%	5.71%	6.06%

Table 1: *Upper bound suboptimality of the best performing benchmark mechanisms on synthetic data independent instances with $\Delta f = 1$, ℓ_2-loss and various combinations of ε and δ.*

		δ									
		0.005	**0.010**	**0.020**	**0.050**	**0.100**	**0.200**	**0.250**	**0.300**	**0.500**	**0.750**
ε	**0.005**	2.54%	1.65%	1.63%	8.73%	25.81%	79.89%	62.19%	32.11%	33.24%	16.62%
	0.010	3.72%	1.86%	11.55%	3.01%	23.68%	79.62%	61.94%	31.67%	35.90%	14.27%
	0.020	3.65%	3.81%	0.21%	21.28%	19.50%	79.58%	61.38%	30.82%	35.80%	18.01%
	0.050	1.02%	1.67%	2.33%	25.51%	7.12%	78.23%	59.72%	28.42%	35.41%	16.34%
	0.100	4.33%	3.24%	10.71%	26.09%	43.67%	75.71%	56.96%	24.88%	34.81%	16.03%
	0.200	10.42%	8.57%	7.11%	20.29%	22.66%	70.80%	51.72%	19.27%	33.79%	16.90%
	0.500	19.42%	15.20%	23.34%	33.68%	40.89%	53.18%	28.01%	10.26%	29.54%	14.77%
	1.000	32.14%	33.47%	37.92%	44.84%	18.02%	23.41%	14.03%	6.72%	22.34%	11.17%
	2.000	20.46%	20.95%	19.14%	14.54%	16.29%	10.64%	8.42%	6.96%	13.10%	6.95%
	5.000	2.81%	2.80%	2.76%	2.65%	2.46%	2.10%	1.92%	1.75%	1.48%	1.08%

Table 2: *Lower bound suboptimality of the best performing benchmark mechanisms on synthetic data independent instances with $\Delta f = 1$, ℓ_2-loss and various combinations of ε and δ.*

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 2. 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.

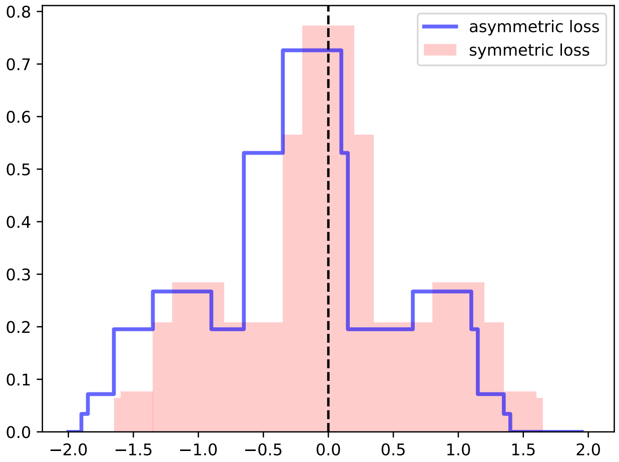

Figure 2: *Optimization-based noise distributions for synthetic data independent instances with* $\Delta f = 1$, $\beta = 0.05$, $\varepsilon = 1$, $\delta = 0.2$, *and two loss functions. The distribution minimizing the symmetric loss* ($c(x) = |x|$) *is shown in red shading, whereas the distribution 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 3, 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.

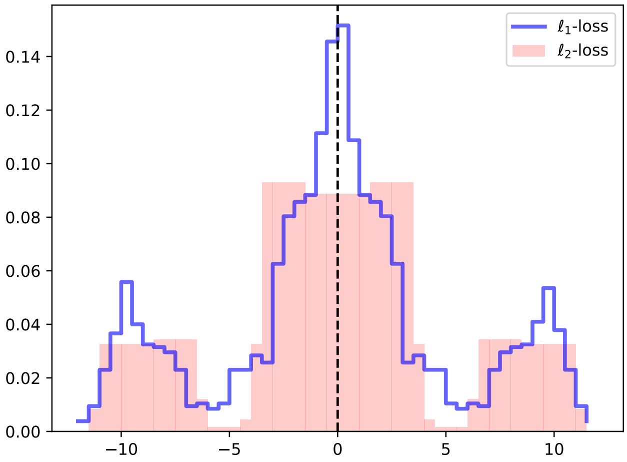

Figure 3: *Optimization-based noise distributions for synthetic data independent instances with $\Delta f = 10$, $\beta = 0.5$, $\varepsilon = 1$, $\delta = 0.4$, and two loss functions. The distribution minimizing the ℓ_1-loss ($c(x) = |x|$) is shown in red shading, whereas the distribution minimizing 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 $F_{\text{TLap}}(F^{-1}_{\text{TLap}}(u)) = u$:

$$\begin{aligned}
&\frac{1}{2} - \underbrace{\text{sign}(F^{-1}_{\text{TLap}}(u))}_{=\text{sign}(u-1/2)} \left[-\frac{1}{2} + B \cdot \lambda \cdot \exp\left(\frac{-|F^{-1}_{\text{TLap}}(u)|}{\lambda}\right) - B \cdot \lambda \cdot \exp\left(\frac{-A}{\lambda}\right)\right] = u \\
\iff & -\frac{1}{2} + B \cdot \lambda \cdot \exp\left(\frac{-|F^{-1}_{\text{TLap}}(u)|}{\lambda}\right) - B \cdot \lambda \cdot \exp\left(\frac{-A}{\lambda}\right) = \frac{u - 1/2}{\underbrace{-\text{sign}(u - 1/2)}_{=-|u-1/2|}} \\
\iff & B \cdot \lambda \cdot \exp\left(\frac{-|F^{-1}_{\text{TLap}}(u)|}{\lambda}\right) - B \cdot \lambda \cdot \exp\left(\frac{-A}{\lambda}\right) = \underbrace{-|u - 1/2| + \frac{1}{2}}_{=\min\{u, 1-u\}} \\
\iff & \exp\left(\frac{-|F^{-1}_{\text{TLap}}(u)|}{\lambda}\right) = \frac{\min\{u, 1-u\}}{B \cdot \lambda} + \exp\left(\frac{-A}{\lambda}\right) \\
\iff & |F^{-1}_{\text{TLap}}(u)| = -\lambda \cdot \log \cdot \left[\frac{\min\{u, 1-u\}}{B \cdot \lambda} + \exp\left(\frac{-A}{\lambda}\right)\right] \\
\iff & F^{-1}_{\text{TLap}}(u) = -\text{sign}(u - 0.5) \cdot \lambda \cdot \log\left[\frac{\min\{u, 1-u\}}{B \cdot \lambda} + \exp\left(\frac{-A}{\lambda}\right)\right].
\end{aligned}$$

We then sample $u \sim [0, 1]$ uniformly at random, and compute $F^{-1}_{\text{TLap}}(u)$ 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 4 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.

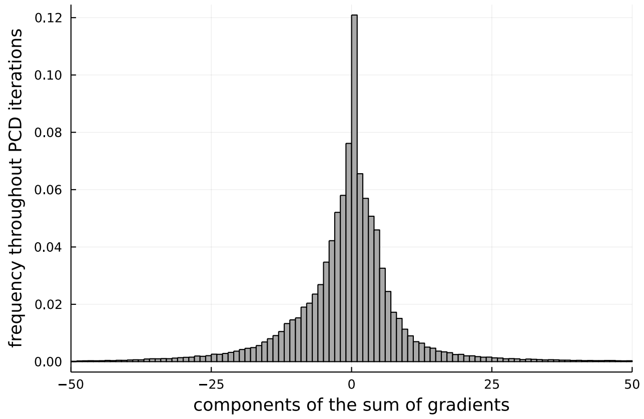

Figure 4: Histogram of the true query answers throughout 100 simulations of a PCD training on the cylinder-bands dataset where each simulation evaluates the query $K \cdot T$ times.

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:

$$\bigcup \{\{(m-k-1) \cdot \beta, (m-k) \cdot \beta, (m-k+1) \cdot \beta\} \ : \ k, m \in [K], \ k \in \Lambda_l, \ m \in \Lambda_{l'}\}$$

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

```
input  : $\boldsymbol{\pi}$, $\boldsymbol{\lambda}$, $\beta$, $p$
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 $V^\star(\varphi, A) = V^\star - \delta$.
```

$$=\{(\lambda_{l'} - \lambda_{l+1}) \cdot \beta, (\lambda_{l'} - \lambda_{l+1}) \cdot \beta + \beta, \ldots, (\lambda_{l'+1} - \lambda_l) \cdot \beta\}.$$

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* $\mathcal{O}(M^2 \cdot N^3)$.

Comment on Theorem 2

In the proof of Theorem 2, we took the strong dual of $\mathrm{D}'(L, \beta)$. Here, we used finite LP duality (the feasibility of the problem was discussed in the main paper). For the completeness of the discussion, note that the dual[1] constraints corresponding to $\boldsymbol{\theta} \in \mathbb{R}^K$ will give us K probability distributions p_k. On the other hand, to be able to represent the dual constraints corresponding to $\psi_k : \mathcal{E}'_k(L, \beta) \mapsto \mathbb{R}_+,\ k \in [K]$ so that they resemble the constraints of $\mathrm{P}'(L, \beta)$, we equivalently

[1] with 'dual' we mean dual of the dual $\mathrm{D}'(L, \beta)$

represent the domain $\mathcal{E}'_k(L,\beta) = [\mathscr{B}(\beta) \cap (\Phi - \underline{\Phi}_k(\beta))] \times \mathcal{F}(L,\beta)$ as

$$\bigcup_{m\in[K]} [\mathscr{B}(\beta) \cap (\Phi_m - \underline{\Phi}_k(\beta))] \times \mathcal{F}(L,\beta),\ k \in [K]$$

and represent the dual constraints with the double index $(k,m) \in [K]^2$, thanks to the additional index used in the above representation. This will let us break down the dual constraints into the "pairs of neighbors" understanding.

Details of DP naïve Bayes Experiments

UCI datasets. The classification datasets are selected according to their popularity from the UCI repository. In our GitHub repository, we share our codes to show how we cleaned and processed data. In summary, we drop the rows with missing values if they comprise less than 5% of the total rows, otherwise, we apply simple missing value imputations (such as the mean value imputation). We also drop erroneous columns, such as those having a single value, or those that are derivatives of the labels. The (ordinal) categorical columns with a large number of categories were encoded so that they are represented with a less (but denser) number of categories. Similarly, if the target variable comprises many labels with few observations, we encode them to have balanced classes. If a dataset has separate training and test sets, we merge them since we will randomly take 100 training set-test set splits as described in the paper. We also randomly permute the rows of each data set before conducting these splits. Moreover, to compute the sensitivities we need to know the minimum and maximum values a feature can obtain; whenever such minima and maxima are not clear, we took the smallest and largest numbers in the columns, respectively.

Optimization parameters. For the data independent noise setting, we solve problem $\mathrm{P}(\boldsymbol{\pi},\beta)$ to obtain a single noise distribution. Here, we determine the length of the support (driven by the limits of $\boldsymbol{\pi}$) by rounding up the support needed by the Truncated Laplace mechanism. We then divide the support into 500 equally sized intervals so that the optimization problem has 500 variables. We minimize the expected value of the noise power. Optimization problems are limited to one-hour runtime, but they terminate within seconds on average due to the performance of the proposed cutting plane method. The same methods are applied for the data dependent noise setting, where we divide the support into 50 equally sized intervals and the function range into 7 intervals (*i.e.*, we optimize 7 noise distributions) that are designed non-uniformly so that the center of the range is partitioned with finer granularity. To speed up the algorithm, we remove the constraints that have slacks larger than 0.08 in every 200 iteration. There are further parameters in the code, documented in the repository, including a binary parameter to enforce monotonicity of distributions (*i.e.*, probabilities of the optimized data independent noise distribution are non-increasing on the sides of the origin), a binary parameter to enforce monotonicity within different distributions (*i.e.*, distributions used for larger function values are skewed left compare), and a value to indicate whether to add all violated constraints whose violations are above a user-specified threshold, instead of adding only the most violated constraint in every iteration.

Statistical significance. In the experiments, we state that our optimization method-based method has a statistically significant improvement over the benchmark methods. We state all the p-values are less than 10^{-7}. The computation of the p-values is as follows. Firstly, we subtract (elementwise) the vector of differentially private naïve Bayes errors attained by using the optimized noise distribution from the same vector of errors attained by the second-best approach. This gives us a sample vector of additional errors arising from using our optimized noise. We then try to reject the hypothesis that the additional error of using optimized noise distributions is non-negative and the improvement of using numerically optimal distributions is not significant compared to the asymptotically optimal mechanisms such as the Truncated Laplace mechanism. To this end, we compute the t-statistic for the mean of the so-called additional errors vector with a hypothesis mean of 0 and sample size of 100. We then report the cumulative probability at this value with a one-sided t-test ($100 - 1$ degrees of freedom) as the p-value.

Details of DP Proximal Coordinate Descent Experiments

The data processing details are identical to that of naïve Bayes setting, except, as discussed in the main paper, here we work on binary classification and hence apply various encoding methods to the output variable. All details of our algorithm are commented on in our codes.

Final note: the piecewise-linear loss function we used for the data dependent noise optimization setting can be found in our C++ codes as one of the loss functions.

Setting Λ and β in Practice

Consider data independent noise mechanisms (our Section 2; extension to Section 3 is analogous). While our upper and lower bounds are valid for any selection of $L \in \mathbb{N}$ and $\beta > 0$, we need both the discretization granularity $\beta = \Delta f / k$ to shrink *and* the support $[-\Lambda \cdot \Delta f, (\Lambda + 1/k) \cdot \Delta f)$ of the noise distribution to grow for our bounds to converge. We illustrate this requirement in Figure 5, where we optimize the expected ℓ_1-loss of a query with sensitivity $\Delta f = 1$ and the privacy parameters $(\varepsilon, \delta) = (1, 0.2)$. Figure 5 (left) shows that for a fixed discretization granularity $\Delta f / k$, the upper and lower bounds improve with an increasing support $[-\Lambda \cdot \Delta f, (\Lambda + 1/k) \cdot \Delta f)$, but that an increasing support alone is not sufficient to guarantee convergence of our bounds. Likewise, Figure 5 (right) shows that for a fixed support $[-\Lambda \cdot \Delta f, (\Lambda + 1/k) \cdot \Delta f)$, refining the discretization granularity $\Delta f / k$ can improve the lower bound, but the upper bounding problem remains infeasible.

To ensure that our upper bound is *finite* and thus provides an implementable mechanism, we select $\Lambda \geq \lceil \frac{1}{\varepsilon} \log(1 + (\exp(\varepsilon) - 1)/2\delta) \rceil$ and $\beta = \Delta f / k$ with $k \geq 2$. Indeed, it follows from Geng et al. (2020, Definition 3) that there is a truncated Laplace mechanism with support $[-A, A]$, where $A = \dfrac{\Delta f}{\varepsilon} \cdot \log(1 + (\exp(\varepsilon) - 1)/2\delta)$, that satisfies (ε, δ)-DP. Soria-Comas and Domingo-Ferrer (2013, Section 4) show that any (untruncated) Laplace mechanism satisfying $(\varepsilon, 0)$-DP can be used to construct a feasible noise distribution that is supported on $\mathbb{R}$ and that is piecewise constant over intervals of length $\Delta f / 2$. We can adapt their argument to show that the aforementioned truncated Laplace mechanism with support $[-A, A]$ can be used to construct a feasible noise distribution that is supported on $[-\Lambda \cdot \Delta f, (\Lambda + 1/2) \cdot \Delta f)$ and that is piecewise

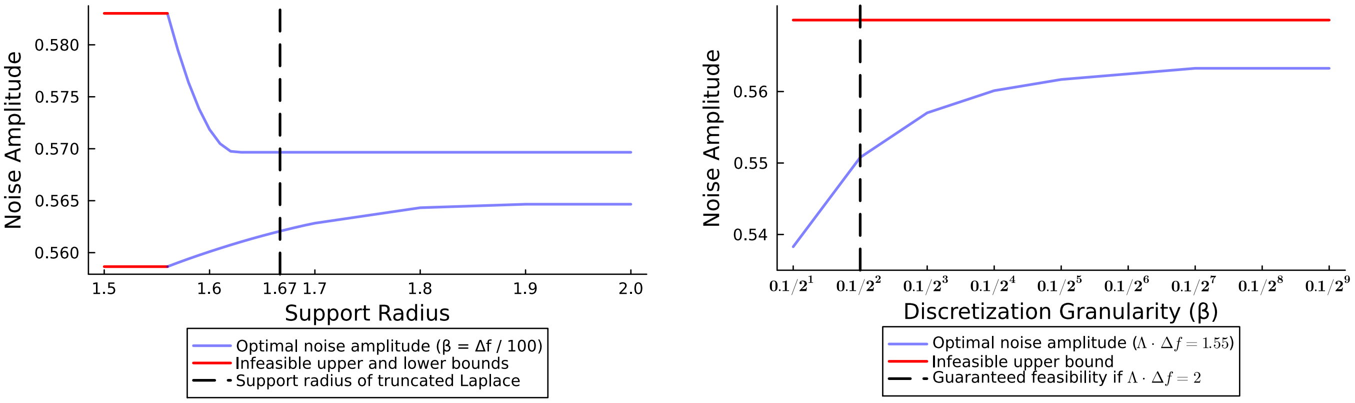

Figure 5: Upper and lower bounds for fixed discretization granularity and varying support size (left) and fixed support size and varying discretization granularity (right).

constant over the intervals $[(\Delta f/2) \cdot i, (\Delta f/2) \cdot (i+1))$, $i \in [\pm 2 \cdot \Lambda]$, of length $\Delta f/2$. In other words, choosing $\Lambda \geq \lceil \frac{1}{\varepsilon} \log(1 + (\exp(\varepsilon) - 1)/2\delta) \rceil$ and $\beta = \Delta f/k$ with $k \geq 2$ ensures that we obtain an implementable mechanism.

While the above parameter choice ensures that we obtain an implementable mechanism, its suboptimality (as measured by the difference of upper and lower bounds) may be large. To address this issue, one can either solve a sequence of refined upper and lower bound approximations, which can be implemented efficiently through warm-starting, or one can solve a single pair of bounding problems for sufficiently large values of L and k. In our numerical experiments, we implemented the following rule-of-thumb: We fix $\Lambda = \lceil \frac{1}{\varepsilon} \log(1 + (\exp(\varepsilon) - 1)/2\delta) \rceil$ as discussed above, and we select $k \in \mathbb{N}$ such that $\beta = \Delta f/k$ results in an optimization problem with approximately $2,000$ variables. This rule-of-thumb consistently resulted in optimality gaps less than 1% in all of our experiments.

Extension to Multi-Dimensional Queries

In contrast to our original submission, which merely identified extension to multi-dimensional queries as a promising research direction in the conclusions, here we contain a numerical experiment that compares two multi-dimensional generalizations of our optimization-based DP approach with state-of-the-art methods from the literature. In particular, we study a two-dimensional query $f : \mathcal{D} \to \mathbb{R}^2$ with ℓ_2-sensitivity $\Delta f := \sup_{(D,D') \in \mathcal{N}} \|f(D) - f(D')\|_2 = 1$, and we aim to minimize the expected ℓ_1-loss (*i.e.*, the *amplitude*) of the noise. We compare the following approaches:

(i) **Truncated Laplace Mechanism.** The classical truncated Laplace mechanism is restricted to the one-dimensional setting. To generalize it to multi-dimensional queries, we make use of the composition theorem to optimally combine two one-dimensional truncated Laplace distributions. In particular, we construct truncated Laplace distributions for all $(\varepsilon, \delta) \in \{(i/20) \cdot \varepsilon \ : \ i \in [20]\} \times \{(i/20) \cdot \delta \ : \ i \in [20]\}$, we consider all pairs $([\varepsilon_1, \delta_1], [\varepsilon_2, \delta_2])$ of truncated Laplace distributions satisfying $\varepsilon_1 + \varepsilon_2 \leq \varepsilon$ and $\delta_1 + \delta_2 \leq \delta$, and we choose the

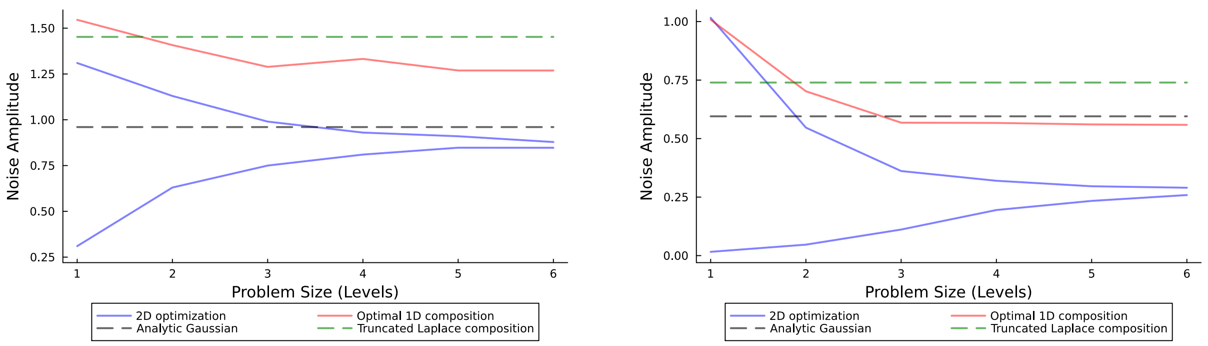

Figure 6: Noise amplitudes of different DP mechanisms in a high-privacy (left; $(\varepsilon, \delta) = (2, 0.2)$) and a low-privacy (right; $(\varepsilon, \delta) = (5, 0.2)$) regime. In both graphs, the abscissae measure the discretization granularity of the optimal 1D composition as well as the 2D obptimization (from coarse, left, to fine, right).

distribution pair that achieves the minimum expected ℓ_1-loss. We refer to this approach as 'Truncated Laplace composition' in the figures.

(ii) **Analytic Gaussian Mechanism.** The analytic Gaussian mechanism (referred to as 'Analytic Gaussian' in the figures) generalizes to the multi-dimensional setting in a well-documented and straightforward way.

(iii) **1D Optimization-Based Approach.** In analogy to the truncated Laplace mechanism, we can generalize the one-dimensional optimization-based approach from our paper to the multi-dimensional setting. To this end, we construct optimization-based distributions for all $(\varepsilon, \delta) \in \{(i/20) \cdot \varepsilon \ : \ i \in [20]\} \times \{(i/20) \cdot \delta \ : \ i \in [20]\}$, we consider all pairs $([\varepsilon_1, \delta_1], [\varepsilon_2, \delta_2])$ of optimization-based distributions whose independent coupling satisfies (ε, δ)-DP, and we choose the coupled distribution pair that achieves the minimum expected ℓ_1-loss. Instead of making use of the composition theorem, we verify (ε, δ)-DP of the independent couplings by solving a two-dimensional generalization of our optimization-based approach from the paper where we fix the decision variables p to the independent coupling and where we verify (ε, δ)-DP by determining the constraint in $\mathrm{P}(\boldsymbol{\pi}, \beta)$ with maximum privacy shortfall (*cf.* Algorithm 2 in the main paper). This amounts to running one iteration of our cutting plane algorithm. We refer to this approach as 'Optimal 1D composition' in the figures.

(iv) **2D Optimization-Based Approach.** We employ a two-dimensional generalization of our cutting plane method that optimizes over noise rectangles instead of noise intervals. We refer to this approach as '2D optimization' in the figures.

Figure 6 compares the objective values (as well as, in '2D optimization', the lower bound on the best achievable objective value) as a function of the discretization granularity, in a high-privacy (left graph) as well as a low-privacy (right graph) regime. We observe that our optimization-based DP approach continues to converge to the optimal noise distribution, and that is enjoys performance guarantees thanks to the dual bounds (which are not available for any of the other approaches). The analytic Gaussian approach performs well in high-privacy regimes, but it is dominated by our 1D optimization-based approach in low-privacy regimes.

Composing truncated Laplace distributions, on the other hand, does not constitute a viable approach in our experiments.

The above naïve extension of our single-dimensional optimization-based DP method to the multi-dimensional setting ('2D optimization') results in an exponential scaling of our optimization problem. This implies that two-dimensional and three-dimensional noise distributions are amenable to optimization with our approach, but we would not expect this strategy to remain viable for higher-dimensional distributions. To obtain a more efficient generalization of our approach to higher-dimensional distributions, we envisage that the following strategies may be promising:

(i) **Structural Properties of Worst-Case Events.** Our implementations of 'Optimal 1D composition' and '2D optimization' rely on a naïve generalization of Algorithm 2 (identification of constraints with maximum privacy shortfall) that confirms for every hyperrectangle $A_{jj'}$ individually whether or not it is contained in a worst-case event. Intuitively, we would expect worst-case events in higher dimensions to exhibit some structure that allows for a more efficient identification (such as connectedness, convexity, etc.). This is particularly likely if we compose marginal noise distributions, for example using an independence copula (as in 'Optimal 1D composition').

(ii) **Basis Function Approximations.** Approximate dynamic programming has been exceptionally successful in approximating challenging high-dimensional problems through optimal linear combinations of moderate numbers of basis functions. We can imagine that in our problem, too, a linear combination of different 'basis distributions' could result in an approximation that is both tractable and (oftentimes) close to optimal.

(iii) **Constraint Sampling Approaches.** Instead of modeling the semi-infinite DP constraints in $\mathrm{P}(\boldsymbol{\pi}, \beta)$ and $\mathrm{D}(\boldsymbol{\pi}, \beta)$ as 'hard constraints' and invoking robust optimization techniques to equivalently reformulate them, we could consider sampling events $(\varphi, A) \in \mathcal{E}(L, \beta)$ from some distribution. There is a rich body of literature which has shown that in many cases, a polynomial number of sampled constraints is sufficient for the original semi-infinite constraint to be satisfied with high probability (De Farias and Van Roy 2004, Calafiore and Campi 2006). This approach could be combined with the basis function approximation from the previous bullet point to reduce both the number of variables and the number of constraints in our optimization problem; alternatively, the basis function approximation may allow us to conclude by itself that a large number of the constraints has become redundant.

All the codes of these experiments are available in our GitHub repository.

References

Calafiore G, Campi MC (2006) The scenario approach to robust control design. *IEEE Transactions on Automatic Control* 51(5):742–753.

De Farias DP, Van Roy B (2004) On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* 29(3):462–478.

Dwork C, Rothblum GN (2016) Concentrated differential privacy. *arXiv preprint 1603.01887* .

Geng Q, Ding W, Guo R, Kumar S (2020) Tight analysis of privacy and utility tradeoff in approximate differential privacy. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 89–99.

Soria-Comas J, Domingo-Ferrer J (2013) Optimal data-independent noise for differential privacy. *Information Sciences* 250:200–214.