

PRIVATE SELECTION FROM PRIVATE CANDIDATES

JINGCHENG LIU AND KUNAL TALWAR

ABSTRACT. Differentially Private algorithms often need to select the best amongst many candidate options. Classical works on this *selection* problem require that the candidates' goodness, measured as a real-valued score function, does not change by much when one person's data changes. In many applications such as hyperparameter optimization, this stability assumption is much too strong. In this work, we consider the selection problem under a much weaker stability assumption on the candidates, namely that the score functions are differentially private. Under this assumption, we present algorithms that are near-optimal along the three relevant dimensions: privacy, utility and computational efficiency.

Our result can be seen as a generalization of the exponential mechanism and its existing generalizations. We also develop an online version of our algorithm, that can be seen as a generalization of the sparse vector technique to this weaker stability assumption. We show how our results imply better algorithms for hyperparameter selection in differentially private machine learning, as well as for adaptive data analysis.

Jingcheng Liu, Computer Science Division, UC Berkeley. Email: liuexp@berkeley.edu.
Kunal Talwar, Google Brain. Email: kunal@google.com.
Some of this work was done while JL was an intern at Google Brain.

1. INTRODUCTION

Differential Privacy [12] is the standard notion of privacy for statistical databases. It imposes a probabilistic constraint on the behavior of the algorithm on datasets that differ in one person’s input. Formally,

Definition 1.1 (Differential Privacy). Let $\mathcal{M} : \mathcal{D}^n \rightarrow \mathcal{R}$ be a randomized algorithm mapping datasets to some range \mathcal{R} . We say that \mathcal{M} is (ε, δ) -differentially private if for all pairs of adjacent datasets $D, D' \in \mathcal{D}^n$, and for all measurable subsets $S \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{M}(D') \in S] + \delta.$$

Here, two datasets are adjacent if they differ in one person’s input. When $\delta = 0$, we will sometimes say that \mathcal{M} is ε -differentially private.

Differential privacy (DP) satisfies nice post-processing and composition properties, allowing for complex differentially private algorithms to be built out of simpler building blocks. In the last decade or so, differentially private algorithms have been designed and analyzed for numerous statistical and machine learning tasks, in most cases by carefully putting together these building blocks. This approach to the design and analysis of differentially private algorithms has proven surprisingly robust and useful.

One of these fundamental building blocks is *Differentially Private Selection*, which aims to select, based on a dataset, the best of many options. For concreteness, suppose that we have a score function $q : [K] \times \mathcal{D}^n \rightarrow \mathbb{R}$ that maps each of K candidates, and a dataset to a real-valued score. The DP selection problem is to select amongst these K candidates, one that (approximately) maximizes this score on a given dataset $D \in \mathcal{D}^n$, while ensuring differential privacy.

One can only hope to approximately maximize q when single individuals in the dataset cannot change any of the score functions $q(i, \cdot)$ too much. This stability of q under small changes in D is usually codified in an assumption that each score function $q(i, D)$ is Lipschitz with respect to Hamming distance 1 changes to D . The Exponential mechanism [26] is an algorithm for DP selection under this assumption and has found numerous applications to the design of DP mechanisms. Several other mechanisms for the private selection problem have been proposed, that improve the utility guarantee under stronger assumptions [3, 7, 27, 28, 31, 33].

In many settings however, the Lipschitzness assumption is much too strong. In this work, we ask: *Are there weaker versions of the stability assumption that allow for private selection?* We show that one can codify the stability simply as differential privacy: the function q , viewed as a randomized algorithm, satisfies differential privacy. Indeed, one can convert a Lipschitz function q' into an ε -DP random function q by simply adding, say, a noise drawn from the Laplace distribution to q' . We assume oracle access to a randomized function that on input (i, D) computes a sample (\tilde{x}, \tilde{q}) from the i -th candidate $\mathcal{M}_i(D)$, where \tilde{q} is the score and \tilde{x} can be any additional output. Moreover, the output distributions of $\mathcal{M}_i(D)$ and $\mathcal{M}_i(D')$ are promised to be close whenever D and D' are neighbors. Here closeness in distributions is taken to mean ε -DP or (ε, δ) -DP. Motivated by applications, we assume that the scores are bounded, say $\tilde{q} \in [0, 1]$.

To measure the quality of a candidate $\mathcal{M}_i(D)$, one option is by the *median* of the distribution: $\text{Median}(\mathcal{M}_i(D)) := \sup \{ \tau : \Pr_{(\tilde{x}, \tilde{q}) \sim \mathcal{M}_i(D)}[\tilde{q} \geq \tau] \geq \frac{1}{2} \}$. However, even if a candidate \mathcal{M}_i is ε_1 -DP, its median can still be very sensitive to the dataset. Thus one could only hope to approximately maximize the median score. Moreover, in many real world applications, one not only wants to find a “good” candidate, but also get a “good” sample from it, especially because these candidates themselves are randomized algorithms. Therefore, we use the following non-private algorithm as our main benchmark: draw a number of samples $(\tilde{x}_j, \tilde{q}_j)$ from every candidates, and then output the one with the highest score \tilde{q}_j . If one only assumes each candidate is individually ε_1 -DP, however, outputting the best of the \tilde{K} options will only be $\tilde{K}\varepsilon_1$ -DP (see Appendix B.1). We would like to

compete with this naive algorithm, while still preserving $O(\varepsilon_1)$ -DP. Another important resource constraint in applications is the computational efficiency of the procedure. In our setting, we would want to minimize the number of oracle calls to $\mathcal{M}_i(D)$ made by our algorithm.

Our first result is a simple algorithm that given as input a threshold τ , outputs a sample (\tilde{x}, \tilde{q}) with score $\tilde{q} \geq \tau$, under the assumption that at least one candidate has a median score of at least τ . This algorithm makes a near linear number of oracle calls, and improves on the quadratic bound that follows from a reinterpretation of a result in [19]. We show that the loss in privacy, utility and efficiency for this algorithm are all close to optimal. Interestingly, this algorithm can be seen as, starting from a naive differentially private algorithm with a poor utility guarantee (e.g., pick a candidate uniformly at random), and then by repeating it in a private way to boost its utility guarantee. In doing so, we get simple algorithms that are both private and have good utility guarantees.

Theorem 1.2. *Fix any $\varepsilon_1 > 0, \tau \in [0, 1]$. Then given ε_1 -DP algorithms $\mathcal{M}_1, \dots, \mathcal{M}_K$, there is an algorithm \mathcal{M} that on any dataset D , outputs a sample (\tilde{x}, \tilde{q}) such that*

- (a) \mathcal{M} is $(2\varepsilon_1)$ -DP.
- (b) $\tilde{q} \geq \tau$.
- (c) Let \tilde{T} be the number of calls the algorithm makes to any $\mathcal{M}_i(D)$, and suppose that $\exists i : \Pr_{q \sim \mathcal{M}_i(D)}[q \geq \tau] \geq \frac{1}{2}$, then $\mathbb{E}\tilde{T} \leq 2K$.

Can we do this without knowing this target value τ ? We give two algorithms that compete with the best i without knowing the target τ . The first can be seen as modifying the naive non-private algorithm by employing a random stopping strategy. In doing so, it guarantees that “outputting the highest scored sample seen so far” is already private. However it pays a small additional privacy penalty: the final privacy cost is $3\varepsilon_1$ instead of $2\varepsilon_1$.

Theorem 1.3. *Fix any $\varepsilon_1 > 0, \gamma \in [0, 1]$. Then given ε_1 -DP algorithms $\mathcal{M}_1, \dots, \mathcal{M}_K$, there is an algorithm \mathcal{M} that on any dataset D , outputs a sample (\tilde{x}, \tilde{q}) such that*

- (a) \mathcal{M} is $(3\varepsilon_1)$ -DP.
- (b) Let \tilde{T} be the number of calls the algorithm makes to any $\mathcal{M}_i(D)$, then $\mathbb{E}\tilde{T} \leq \frac{1}{\gamma}$.
- (c) \tilde{q} is the highest scored sample among the \tilde{T} samples seen so far.

Our second algorithm keeps the privacy cost to essentially $2\varepsilon_1$, at the cost of a slightly higher runtime and a more complicated algorithm and analysis. This is valuable since in some settings, the utility of the base algorithm is quite sensitive with respect to the privacy parameter ε_1 . In such settings, with a final target privacy parameter of ε_{fin} , the second algorithm can allow us to give each \mathcal{M}_i a privacy budget of $\approx \varepsilon_{fin}/2$, which can lead to a better utility than the $\approx (\varepsilon_{fin}/3)$ -DP \mathcal{M}_i ’s needed for the first simpler algorithm.

Theorem 1.4. *Fix any $\varepsilon_1 > 0, \varepsilon_0 \in [0, 1], \beta > 0, R \in \mathbb{N}$. Suppose that there are ε_1 -DP algorithms $\mathcal{M}_1, \dots, \mathcal{M}_K$ and let $\tau^*(D) = \max_i \text{Median}(\mathcal{M}_i(D))$. There is an algorithm \mathcal{M} that on any dataset D either outputs \perp , or outputs a sample (\tilde{x}, \tilde{q}) such that*

- (a) \mathcal{M} is $(2\varepsilon_1 + \varepsilon_0, \delta)$ -DP.
- (b) Except with probability $\beta + \delta/R$, \tilde{x} has quality at least $\tau^* - \frac{1}{R}$.
- (c) The number of calls \tilde{T} that the algorithm makes to any $\mathcal{M}_i(D)$ satisfies (deterministically)

$$\tilde{T} \leq O\left(K \left(\frac{R+1}{\beta^2}\right)^{6+\frac{12\varepsilon_1}{\varepsilon_0}} \left(\frac{\ln \frac{R}{\delta}}{\varepsilon_0^2} + \frac{\ln \frac{1}{\varepsilon_0}}{\beta}\right)\right).$$

Furthermore, $\Pr[\mathcal{M} \text{ outputs } \perp] \leq \beta + \delta$.

In the process, we develop an online version of our algorithm, which can be seen as a generalization of the sparse vector technique [13] to this privacy-instead-of-Lipschitzness setting. This algorithm takes as input a sequence of mechanisms $\mathcal{M}_i(\cdot)$ and τ_i , and stops at the first i such that $\mathcal{M}_i(\cdot)$ has median score larger than τ_i .

Theorem 1.5. *There is an (ε_3, δ) -DP mechanism \mathcal{M}_{sv} such that, for any $p^* \in (0, 1), \beta \in (0, 1)$, and for any sequence of ε_1 -DP mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_K$ and any sequence of thresholds τ_1, \dots, τ_K :*

- (a) *If there is an i such that $\Pr[\mathcal{M}_i(D) \geq \tau_i] \geq p^*$, then \mathcal{M} outputs i with probability $(1 - \beta)$.*
- (b) *If \mathcal{M} outputs i , then except with probability β , $\Pr[\mathcal{M}_i(D) \geq \tau_i] \geq (\frac{\beta}{K})^{O(\varepsilon_1/\varepsilon_3)} \cdot p^*$.*

Several remarks are in order. First note that the stability assumption that we use, i.e. that of differential privacy, is in some sense the weakest possible. Indeed if we want the final outcome to be differentially private and we treat each mechanism as a blackbox, it is easy to see that each mechanism itself must be differentially private. In other words, we have relaxed the Lipschitzness condition to the weakest possible condition that would allow for differentially private selection. Our algorithm suffers a factor of two loss in the privacy parameter. In Appendix D, we show that this factor of two loss is unavoidable even in simple settings. Note also that our algorithm only makes $\tilde{O}(K)$ oracle calls, whereas even computing the maximum non-privately would require K oracle calls.

We next outline some motivating applications of our work.

Hyperparameter/algorithm Selection: When designing practical machine learning algorithms, one often ends up choosing amongst different algorithms/models, or setting values for common hyperparameters such as the learning rate in an algorithm. This *hyperparameter selection* problem has attracted a lot of interest in recent years [35]. Differentially private ML algorithms such as [1, 30] have many of these hyperparameters, and often add on a few hyperparameters of their own. A common approach in the non-private setting is to try out several (or all) values of the hyperparameters and select the best one based on the performance on a validation set. Doing this with privacy requires more care. Chaudhuri and Vinterbo [9] studied this problem formally under strong assumptions on the algorithm. These assumptions, however, can be hard to enforce and one would like to design an algorithm that works without any additional assumptions. Note that given K choices for the hyperparameters, and an ε -DP learner, one can publish K models and select the best, say using the exponential mechanism. This approach only gives εK -DP, which allows for privacy budget of only ε/K (or $\varepsilon\sqrt{\log \frac{1}{\delta}}/\sqrt{K}$ if using advanced composition) for the learner, which often translates to significantly poorer utility guarantee. In this setting, note also that each oracle call is a run of the DP learner for some hyperparameter setting, that can involve a large computational cost.

Our work shows how to compete with the best choices of hyperparameters in the non-private setting while satisfying $O(\varepsilon)$ -DP, at a small computational overhead.

Adaptive data analysis beyond low-sensitivity queries: One of the applications of DP, beyond privacy itself, is in understanding overfitting in the adaptive setting where the same dataset is used in a sequence of analyses, chosen adaptively based on the results of previous ones. This problem, sometimes referred to as the *garden of forking paths* [17], can lead to a breakdown of standard statistical guarantees. A beautiful recent line of work [2, 10] shows that when these analyses take the form of low-sensitivity queries, using differentially private versions of these analyses allows us to improve the sample complexity quadratically over what would otherwise be possible. Often, however, the forking paths can involve queries that are not low-sensitivity. For example, at some step an analyst may choose the best k for k -means clustering or may choose the clustering algorithm itself amongst one of several. At another step, the analyst may project the data for a carefully chosen target rank, and may choose to use a projection algorithm such as PCA, or an ℓ_p version of

PCA to get outlier robustness, for a carefully chosen p . Making these choices differentially private naively would involve paying for the privacy cost of each of the options considered, even though only one may be used in the subsequent analysis. Our work shows that if one uses a differentially private algorithm to score each of the options, selecting amongst them can be done while paying the adaptivity cost of only one query, essentially independently of the number K of options considered.

Generalizing the Exponential Mechanism: Beyond these applications, our result can be viewed as a generalization of the exponential mechanism. Given a score function q that has sensitivity S , observe that adding Laplace noise of scale S/ϵ to the score gives us an ϵ -DP mechanism. Our algorithm can be then used to select amongst these. We can however relax the assumptions. If we allow the score functions to have different sensitivities, we can still use our framework and recover the generalized exponential mechanism of Raskhodnikova and Smith [31]. If the score functions have small smoothed sensitivity [29], we get a smooth sensitivity version of the exponential mechanism. This last result does not seem to follow from known techniques.

Private amplification for private algorithms: Beyond these applications, our result can be viewed as an extension of the *private amplification scheme* introduced in [19]. Given a private algorithm, which is usually a randomized algorithm, ideally one would like to run it multiple times, and then choose the *best* run so as to obtain an output with a higher quality. Here the quality measure can either be the success probability, or any other utility measure of the output. This is trivial in the non-private setting. Is it possible to compete with such a naive repetition strategy in a differentially private way? In this work, we present an algorithm that can be seen as modifying the naive repetition strategy with a random stopping time, which is arguably almost as competitive as the non-private naive repetition.

1.1. Other Related Work. The Differentially Private Selection problem, often known as differentially private maximization, is a very general algorithmic problem that arises in many applications. Some examples include private PAC learning [21], private frequent itemset mining [4], private PCA [8, 20] and private multiple hypothesis testing [15, 34]. The Sparse Vector Technique can be viewed in hindsight as a novel solution to the online version of the selection problem, under the assumption that the target value τ is known in advance. This technique was introduced by Dwork et al. [13]. We refer the reader to the book by Dwork and Roth [14] for further applications of these techniques.

Several generalization of the exponential mechanisms have been proposed. Smith and Thakurta [33] and Beimel et al. [3] showed that the utility guarantee can be improved using the propose-test-release framework of Dwork and Lei [11] when there is a large margin between the maximum and the rest. Chaudhuri et al. [7] gave an elegant algorithm that can exploit a large margin between the maximum and the k th maximum for any k . Raskhodnikova and Smith [31] proposed the generalized exponential mechanism whose utility depends on the sensitivity of the maximizer, rather than the worst-case sensitivity. Minami et al. [27] show that under certain assumptions on the base distribution, the sensitivity assumptions on the loss function can be significantly relaxed. Our algorithms can also be seen as a natural generalization of the Laplace mechanism. Given a Lipschitz score function q' , one can convert it into an ϵ -DP score function q by adding a Laplace noise. Then the Laplace mechanism says that one can just output the max of the noise-added scores. However, the Laplace relies crucially on the fact that the noise is a Laplace noise. As we will discuss in Appendix B.1, under the mere assumption that the score function is ϵ -DP, outputting the max will inevitably incur a factor of K loss in privacy.

The problem of algorithm selection has also been studied in [22] where the best parameters are learnt from features of the problem. Ligett et al. [24] study the problem of picking from a sequence of algorithms with increasing privacy costs, until one with good utility is found, for a special class of mechanisms.

The problem of private median finding, and more generally private percentile estimation has been studied in several works [5, 11, 29, 32]. While syntactically similar to the threshold estimation problem studied in Section 4, the assumptions on the data in those works are very different from ours and we do not believe that the techniques in those works apply to the setting of interest in this work.

1.2. Organization. The rest of the paper is organized as follows. In Section 3 we present our algorithm for the known threshold case. Section 4 describes our sparse vector and general selection algorithms. We sketch applications of our results in Section 5. The appendices contain some deferred proofs, show why simpler natural approaches do not work for our problem, and show a lower bound on the privacy overhead.

2. PRELIMINARY AND NOTATIONS

For a random variable X and distribution Q , we write $X \sim Q$ if X is distributed according to the law of Q .

Let $D(A\|B)$ be the max-divergence of two random variables defined as follows:

$$D(A\|B) = \max_{S \subseteq \text{Supp}(A)} \left[\ln \frac{\Pr[A \in S]}{\Pr[B \in S]} \right].$$

Then we define $D_\delta(A\|B)$ as:

$$D_\delta(A\|B) = \max_{S \subseteq \text{Supp}(A)} \left[\ln \frac{\Pr[A \in S] - \delta}{\Pr[B \in S]} \right].$$

For convenience, for distributions Q_1 and Q_2 , let q_1, q_2 be random variables distributed as Q_1 and Q_2 respectively, then we will also write $D(Q_1\|Q_2) := D(q_1\|q_2)$, and similarly $D_\delta(Q_1\|Q_2) := D_\delta(q_1\|q_2)$.

For a distribution $Q(D)$ that depends on datasets D , we say that Q satisfies ε -differential privacy (or simply written as ε -DP), if for every two neighboring datasets D_1, D_2 , $D(Q(D_1)\|Q(D_2)) \leq \varepsilon$. And we say Q satisfies (ε, δ) -DP if for every two neighboring datasets D_1, D_2 , $D_\delta(Q(D_1)\|Q(D_2)) \leq \varepsilon$.

Given a function f on dataset D , we say that f is t -Lipschitz if for any two neighboring dataset D, D' , $|f(D) - f(D')| \leq t$.

3. PRIVATE SELECTION

Let $\{M_i(D)\}_{i=1}^K$ be a set of differentially private mechanisms, that is, for every i , M_i is a differentially private mechanism with respect to the dataset D . We will also refer to the set of M_i as *private candidates*. For convenience, we will also treat a randomized mechanism $M_i(D)$ as a distribution, and write $m \sim M_i(D)$ if m follows the output distribution of $M_i(D)$. Let $\{q_i\}$ be scoring functions over the output of these mechanisms, that is, for $m \sim M_i(D)$, $q_i(m)$ is the score for m . We assume that there is a total ordering of the candidates: when two candidates have the same score, we assume that there is an arbitrary tie-breaking rule (e.g., by alphabetical ordering). Given a total ordering of the candidates, without loss of generality we will further assume that each option has a different score.

The goal of *private selection* is to select (m, i) that (approximately) maximizes the score of $q_i(m)$. Naively, a natural algorithm is to draw samples $m_i \sim M_i$ for every i , and then output the pair (m_i, i) with the highest score $q_i(m_i)$. Unfortunately this naive algorithm is not private. The detailed discussion and analysis is deferred to Appendix B.1. The next natural algorithm would be to output the p -th percentile best, which unfortunately is also not private. Again we defer the analysis to Appendix B.3.

In this section, we will start with the following naive algorithm that is guaranteed to be *private* but not very *useful* (has poor utility guarantee): we choose a candidate i uniformly at random and output $M_i(D)$. It is not hard to see that such a choice of candidate is at least as private as the individual candidates. However, the probability of getting a reasonably “good” candidate can be of the order $O(1/K)$. Nevertheless, we will show how to boost its usefulness (utility guarantee) by *thresholding* or *random stopping*. As a result, this leads to simple and practical algorithms that are also able to compete with the *best* candidates in a differentially private way.

Formally, we consider a randomized mechanism $Q(D)$, where every output comes with a utility score q : $(x, q) \in \Omega \times \mathbb{R}$. For convenience, we will abuse notation and also denote the output distribution of the randomized mechanism $Q(D)$ by $Q(D)$, and write $(\tilde{x}, \tilde{q}) \sim Q(D)$ to indicate that (\tilde{x}, \tilde{q}) is obtained by running the randomized mechanism $Q(D)$. Given blackbox access to $Q(D)$, the goal is to find (x, q) that (approximately) *maximizes* the score: e.g., they are the top 1%, that is, $\Pr_{(\tilde{x}, \tilde{q}) \sim Q(D)}[\tilde{q} > q] < 0.01$. When it is clear from the context, we will also simply write $q \sim Q(D)$ for taking only the q part of the pair (x, q) .

To apply this framework to the private selection problem, we define a randomized mechanism $Q(D)$ as follows: we first sample $i \sim \text{Uniform}[K]$, then sample $m \sim M_i(D)$, and evaluate the score $q_i(m)$, and output $((i, m), q_i(m))$. The above sampling process implements an oracle access to the naive algorithm that outputs a candidate uniformly at random. Our goal is to boost the utility of such a naive algorithm. While our algorithms work for more general distribution of $Q(D)$, where the candidate i can be drawn from *any* samplable distribution, we will focus in this work on the case when i is drawn uniformly from a finite set of candidates (e.g., due to the lack of domain knowledge). It is also worth noting that if M_i is ε_1 -DP for every i , then so is Q . Similarly if M_i is $(\varepsilon_1, \delta_1)$ -DP for every i , then so is Q .

3.1. Private selection with a known threshold τ . We consider a thresholding algorithm, which for a given threshold, repeatedly samples from the candidates until we get one that is above the threshold. In addition, we have a small probability γ of stopping at each step. See Algorithm 1 for a more formal description.

Algorithm 1 Thresholding with a known threshold τ .

Input: a threshold τ , a budget $\gamma \leq 1$ and $\varepsilon_0 \leq 1$, number of steps $T \geq \max\left\{\frac{1}{\gamma} \ln \frac{2}{\varepsilon_0}, 1 + \frac{1}{e\gamma}\right\}$, and sampling access to $Q(D)$.

For $j = 1, \dots, T$:

- draw $(x, q) \sim Q(D)$
- if $q \geq \tau$ then output (x, q) and halt;
- flip a γ -biased coin: with probability γ , output \perp and halt;

Output \perp and halt.

We assume that the adversary can only observe the final output of the algorithm. We show that for any choice of parameters, the algorithm is private; and if the given threshold τ is a “good” threshold, the algorithm is unlikely to output \perp .

Theorem 3.1. *Fix any $\varepsilon_1, \delta_1 > 0, \varepsilon_0 \in [0, 1], \gamma \in [0, 1]$. Let T be any integer such that $T \geq \max\left\{\frac{1}{\gamma} \ln \frac{2}{\varepsilon_0}, 1 + \frac{1}{e\gamma}\right\}$. Then Algorithm 1 with these parameters satisfies the following:*

(a) *Let $A_{\text{out}}(D)$ be the output of Algorithm 1, then for $q \geq \tau$,*

$$\Pr[A_{\text{out}}(D) = (x, q)] \propto \Pr_{(\tilde{x}, \tilde{q}) \sim Q(D)}[(\tilde{x}, \tilde{q}) = (x, q)].$$

(b) *If Q is ε_1 -DP, then the output is $(2\varepsilon_1 + \varepsilon_0)$ -DP.*

(c) If Q is $(\varepsilon_1, \delta_1)$ -DP, then the output is $(2\varepsilon_1 + \varepsilon_0, 3e^{2\varepsilon_1 + \varepsilon_0} \cdot \frac{\delta_1}{\gamma})$ -DP.

(d) Let \tilde{T} be the number of iterations of the algorithm, and let $p_1 = \Pr_{q \sim Q(D)}[q \geq \tau]$, then

$$\mathbb{E}\tilde{T} \leq \frac{1}{p_1(1-\gamma) + \gamma} \leq \min \left\{ \frac{1}{p_1}, \frac{1}{\gamma} \right\}.$$

(e) Furthermore, $\Pr[\text{output} \perp] \leq \frac{(1-p_1)(1+\varepsilon_0/2)}{p_1} \gamma$.

Due to space considerations, we defer this proof to Appendix A.1. As a remark, it is clear that in the worst case, the number of iterations of Algorithm 1 is no more than T ; this theorem provides a more average-case guarantee: the larger p_1 (or γ) is, the more likely that the algorithm will terminate (much) sooner than T . Moreover, it is worth noting that the larger the setting of T is, the smaller we can set ε_0, γ , providing more privacy and utility. In particular, the above theorem holds even if we set $\gamma = 0, \varepsilon_0 = 0$ but $T = \infty$, in other words, we run the algorithm till it stops by itself. However this would not be a very practical setting: if one started with a “bad” threshold, the algorithm may never stop. In that case, one may want to stop the algorithm and try a different threshold. Therefore, for practical purposes one may want to set $\gamma > 0$ and $\varepsilon_0 > 0$.

3.2. Random stopping without thresholding. In this subsection, we show that the idea of random stopping leads to a simple private algorithm, even without knowing the threshold. It is similar to Algorithm 1 but without the thresholding part: draw a random number of samples, and then output the best option.

Algorithm 2 Outputting the highest score with random stopping.

Input: a budget $\gamma \leq 1$ and the sampling access to $Q(D)$.

Initialize the list (multiset) $S = \emptyset$.

For $j = 1, \dots, \infty$:

- draw $(x, q) \sim Q(D)$
 - $S \leftarrow S \cup \{(x, q)\}$
 - flip a γ -biased coin: with probability γ , we output the highest scored candidate from S and halt;
-

Theorem 3.2. Fix any $\varepsilon_1 > 0, \gamma \in [0, 1]$. If Q is ε_1 -DP, then the output of Algorithm 2 is $(3\varepsilon_1)$ -DP.

Proof. We first consider the event of getting the output (x, q) from Algorithm 2 on neighboring datasets D and D' . Without loss of generality, we assume that each option has a different score.¹ Then we denote

$$\begin{aligned} p &:= \Pr_{\tilde{q} \sim Q(D)}[\tilde{q} = q] & \text{and} & & p' &:= \Pr_{\tilde{q} \sim Q(D')}[\tilde{q} = q], \\ p_0 &:= \Pr_{\tilde{q} \sim Q(D)}[\tilde{q} > q] & \text{and} & & p'_0 &:= \Pr_{\tilde{q} \sim Q(D')}[\tilde{q} > q], \\ p_1 &:= \Pr_{\tilde{q} \sim Q(D)}[\tilde{q} \geq q] & \text{and} & & p'_1 &:= \Pr_{\tilde{q} \sim Q(D')}[\tilde{q} \geq q]. \end{aligned}$$

Notice that $p = \Pr_{(\tilde{x}, \tilde{q}) \sim Q(D)}[(\tilde{x}, \tilde{q}) = (x, q)]$, $p_1 = p_0 + p$, and $p'_1 = p'_0 + p'$.

We define the highest score for a set (or a multiset) S of tuples (x, q) as

$$\max S := \max_{(x, q) \in S} q.$$

¹Otherwise, whenever we write $q_1 > q$, we break ties using the same total ordering of the candidates.

Let $A_{\text{out}}(D)$ be the output of Algorithm 2 on D , then we have

$$\begin{aligned}
& \Pr[A_{\text{out}}(D) = (x, q)] \\
&= \sum_{j=1}^{\infty} \Pr[A_{\text{out}}(D) = (x, q) \wedge |S| = j] \\
&= \sum_{j=1}^{\infty} \Pr[|S| = j] \cdot \Pr[\max S \leq q, \text{ and } (x, q) \in S \mid |S| = j] \\
&= \sum_{j=1}^{\infty} (1 - \gamma)^{j-1} \gamma \cdot \Pr[\max S \leq q, \text{ and } (x, q) \in S \mid |S| = j].
\end{aligned}$$

Then, observe that

$$\Pr[\max S \leq q \mid |S| = j] = (1 - p_0)^j,$$

and

$$\Pr[(x, q) \in S \mid \max S \leq q, \text{ and } |S| = j] = 1 - \left(1 - \frac{p}{1 - p_0}\right)^j = 1 - \left(\frac{1 - p_1}{1 - p_0}\right)^j.$$

Together we have

$$\begin{aligned}
\Pr[A_{\text{out}}(D) = (x, q)] &= \sum_{j=1}^{\infty} (1 - \gamma)^{j-1} \gamma \cdot (1 - p_0)^j \left(1 - \left(\frac{1 - p_1}{1 - p_0}\right)^j\right) \\
&= \sum_{j=1}^{\infty} (1 - \gamma)^{j-1} \gamma \cdot ((1 - p_0)^j - (1 - p_1)^j) \\
&= \frac{\gamma(1 - p_0)}{1 - (1 - \gamma)(1 - p_0)} - \frac{\gamma(1 - p_1)}{1 - (1 - \gamma)(1 - p_1)} \\
&= \frac{\gamma(p_1 - p_0)}{(p_0(1 - \gamma) + \gamma)(p_1(1 - \gamma) + \gamma)} \\
&= \frac{\gamma p}{(p_0(1 - \gamma) + \gamma)(p_1(1 - \gamma) + \gamma)}.
\end{aligned}$$

Since Q is ε_1 -DP, we have that p, p_0, p_1 are ε_1 -close (in a DP sense) to p', p'_0, p'_1 , respectively. Then,

$$\frac{\Pr[A_{\text{out}}(D) = (x, q)]}{\Pr[A_{\text{out}}(D') = (x, q)]} = \frac{p}{p'} \cdot \frac{p'_0(1 - \gamma) + \gamma}{p_0(1 - \gamma) + \gamma} \cdot \frac{p'_1(1 - \gamma) + \gamma}{p_1(1 - \gamma) + \gamma} \leq \exp(3\varepsilon_1).$$

□

The following utility bound holds for this algorithm.

Theorem 3.3. *For $p > 0$, let $Q^{(p)}(D) = \sup\{z : \Pr[Q(D) \geq z] > p\}$. Then the output of Algorithm 2 has score at least $Q^{(p)}(D)$ except with probability γ/p .*

Proof. Let $A_{\text{out}}(D)$ be the output of Algorithm 2 on D , then we write

$$\begin{aligned}
\Pr[A_{\text{out}}(D) < Q^{(p)}(D)] &= \sum_{j=1}^{\infty} \Pr[|S| = j] \cdot \Pr[\max S < Q^{(p)}(D) \mid |S| = j] \\
&= \sum_{j=1}^{\infty} (1 - \gamma)^{j-1} \gamma \cdot (1 - p)^j \\
&\leq \gamma/p.
\end{aligned}$$

□

Instead of random stopping, one can also design a hard stopping variant of this algorithm similar to that of Algorithm 1, and allow for (ε, δ) -DP input algorithms.

Theorem 3.4. *Fix any $\gamma \in [0, 1]$, $\delta_2 > 0$ and let $T = \frac{1}{\gamma} \log \frac{1}{\delta_2}$. Consider a variant of Algorithm 2 that outputs the highest scored candidate from S if j reaches T . If Q is $(\varepsilon_1, \delta_1)$ -DP, then the output of this algorithm is $(3\varepsilon_1 + 3\sqrt{2\delta_1}, \delta)$ -DP for $\delta = \sqrt{2\delta_1}T + \delta_2$.*

Proof. We simply reduce to Theorem 3.2 using simple properties of (ε, δ) -DP. We give details next, using folklore results proven in Appendix E. Fix a pair of neighboring datasets D and D' . Then we can define an event B such that $\Pr[B] \leq \sqrt{\delta_1}$ and that $Q(D) \mid B^c$ and $Q(D') \mid B^c$ are multiplicatively $\varepsilon_1 + \sqrt{2\delta_1}$ close. Let B_j be the event B in the j th call to Q . Further, let C be the event that the algorithm reaches step T . Conditioned on $(\cup_{j=1}^T B_j \cup C)^c$, the run of this algorithm can be coupled with a run of Algorithm 2 for a pure DP Q . Further, the probability of the event $\cup_j B_j \cup C$ is at most $\sqrt{2\delta_1}T + \delta_2$. The claim follows. □

Since δ_1 is typically smaller than a polynomial, we have not attempted to optimize the δ term in this theorem. We conclude with a remark that, in the case when Q satisfies purely ε_1 -DP, one can show that the hard stopping variant of Algorithm 2 preserves purely $\approx 3\varepsilon_1$ -DP.

Theorem 3.5. *Fix any $\varepsilon_0 \in (0, 1/2)$, $\gamma \in [0, 1]$, $\delta_2 > 0$ and let $T = \left\lceil \frac{1}{\gamma} \left(\ln \frac{2(1+\gamma)^2}{\varepsilon_0 \gamma^2} + \ln \ln \frac{2(1+\gamma)^2}{\varepsilon_0 \gamma^2} \right) \right\rceil$. Consider a variant of Algorithm 2 that outputs the highest scored candidate from S if j reaches T . If Q is ε_1 -DP, then the output of this algorithm is $(3\varepsilon_1 + 3\varepsilon_0)$ -DP.*

The proof of this theorem is quite involved and is deferred to Appendix A.2.

4. SEARCHING FOR A PERCENTILE-THRESHOLD: PRIVACY-PRESERVING SPARSE VECTOR

In this section we consider the problem of searching for a percentile-threshold τ for any given percentile p_* in a differentially private way. We start by defining some notations. Given any sequence of randomized queries $\{Q_i\}$, we write $q_i \sim Q_i(D)$ to indicate that q_i is obtained from running the randomized query Q_i on dataset D . In other words, $q_i \sim Q_i(D)$ means that q_i follows the output distribution of the randomized query Q_i on dataset D . We will treat these $Q_i(D)$ as samplable distributions, where each Q_i is ε_1 -DP. Then for any sequence of thresholds $\{\tau_i\}$, and a target threshold $p_* \in (0, 1)$, we would like to test if $\Pr_{q_i \sim Q_i(D)}[q_i \geq \tau_i] > p_*$ and output the first one that is above the threshold, and in a differentially private way.

It is worth noting that this can be seen as an extension of the standard sparse vector algorithm for Lipschitz queries: given 1-Lipschitz queries f_1, \dots, f_k and a threshold τ_0 , if we set $p_* = \frac{1}{2}$, $Q_i = f_i + \text{Lap}\left(\frac{4}{\varepsilon_1}\right)$ and $\tau_i = \tau_0$, then it is not hard to check that the queries Q_i are now ε_1 -DP, and the first query Q_i above the percentile-threshold is exactly the same as the first query f_i above the query threshold τ_0 (that is, the first f_i with median score at least τ_0). Answering such a percentile query exactly is not private (see Appendix B.3 for an example for $p_* = 1/2$), so we will have to relax the goal of finding the first above percentile-threshold query. Similar to the standard setting, we would like that:

- if a query is much below the threshold, that is, $\Pr_{q_i \sim Q_i(D)}[q_i \geq \tau_i] \ll p_*$, then our algorithm should report “below threshold” (denoted by \perp);
- if a query is much above the threshold, that is, $(1 - \Pr_{q_i \sim Q_i(D)}[q_i \geq \tau_i]) \ll (1 - p_*)$, then our algorithm should report “above threshold” (denoted by \top).

In fact, our algorithm will be a natural extension of the standard sparse vector algorithm.

4.1. Sparse vector for online private queries with the help of a percentile oracle. To illustrate ideas, we will start by assuming that we have access to an *exact* percentile oracle: $p(\tau_i, Q_i) := \Pr_{q_i \sim Q_i}[q_i \geq \tau_i]$. As a remark, such a percentile oracle is available in the standard sparse vector algorithm, which is just the cumulative distribution function of the Laplace distribution. We observe that if the randomized queries Q_i are ε_1 -DP, then both $\ln p(\tau_i, Q_i)$ and $\ln(1 - p(\tau_i, Q_i))$ are ε_1 -Lipschitz. In other words, although we no longer have Lipschitzness in the “answer of a query” space (that is, the quantile space), the fact that each query is ε_1 -DP will ensure that we have Lipschitzness in the logarithm of the percentile space (that is, the log of the CDF). This allows us to adapt the sparse vector algorithm to the log of the percentile space.

Let $\Phi(x) := \frac{x}{1-x}$. Note that $\ln \Phi(p(\tau_i, Q_i))$ is $2\varepsilon_1$ -Lipschitz: since both $\ln p(\tau_i, Q_i)$ and $\ln(1 - p(\tau_i, Q_i))$ are ε_1 -Lipschitz, and $\ln \Phi(p(\tau_i, Q_i))$ is just the difference of two ε_1 -Lipschitz functions. Also notice that Φ is a strictly increasing function for $x \in (0, 1)$. Given access to the oracle $p(\tau_i, Q_i)$, we can then adapt the sparse vector algorithm as in Algorithm 3.

Algorithm 3 AboveThreshold algorithm assuming the oracle $p(\tau_i, Q_i)$.

Input: $\varepsilon_1, \varepsilon_3, p_*$, a stream of thresholds $\{\tau_i\}$ and randomized queries $\{Q_i(D)\}$

Sample $\nu \sim \text{Lap}\left(\frac{4\varepsilon_1}{\varepsilon_3}\right)$;

For $i = 1, \dots$:

- let $\xi_i \sim \text{Lap}\left(\frac{8\varepsilon_1}{\varepsilon_3}\right)$;
 - $e^{\xi_i} \cdot \Phi(p(\tau_i, Q_i)) > e^\nu \Phi(p_*)$, output $a_i = \top$ and halt;
 - otherwise output $a_i = \perp$;
-

Theorem 4.1. *If for every i , Q_i is ε_1 -DP, then*

(a) *Algorithm 3 is ε_3 -DP.*

(b) *Conditional on Algorithm 3 reporting the R -th query Q_R is “above threshold”, we have that*

$\forall \beta \in (0, 1), \Pr \left[\Phi(p(\tau_R, Q_R)) \leq \left(\frac{\beta}{R+1} \right)^{\frac{12\varepsilon_1}{\varepsilon_3}} \Phi(p_*) \right] \leq \beta$. *In other words, the algorithm does not stop too early.*

(c) *Conditional on Algorithm 3 reporting the R -th query Q_R is “above threshold”, we have that*

$\forall \beta \in (0, 1), \Pr \left[\exists i < R : \Phi(p(\tau_i, Q_i)) \geq \left(\frac{R+1}{\beta} \right)^{\frac{12\varepsilon_1}{\varepsilon_3}} \Phi(p_*) \right] \leq \beta$. *In other words, the algorithm does not stop too late.*

(d) $\forall \beta \in (0, 1)$, *if for some i , $\Phi(p(\tau_i, Q_i)) \geq \left(\frac{1}{\beta} \right)^{\frac{12\varepsilon_1}{\varepsilon_3}} \Phi(p_*)$, then $\Pr[a_i = \top | \forall j < i, a_j = \perp] \leq \beta$.*

In other words, on a query that is way above the threshold the algorithm will likely halt.

Proof. (Sketch) Part (a), part (b) and part (c) all follow from the standard sparse vector analysis (see, e.g., [14]), and the fact that $\ln \Phi(p(\tau_i, Q_i))$ is $2\varepsilon_1$ -Lipschitz. Observe that the test $e^{\xi_i} \cdot \Phi(p(\tau_i, Q_i)) > e^\nu \Phi(p_*)$ is equivalent to $\xi_i + \ln \Phi(p(\tau_i, Q_i)) > \nu + \ln \Phi(p_*)$. Therefore, if we view $\ln \Phi(p(\tau_i, Q_i))$ as the i -th query (which is $2\varepsilon_1$ -Lipschitz) and $\ln \Phi(p_*)$ as the threshold, then this is indeed the standard sparse vector setting. The details are omitted here as we will see proofs for stronger claims for the actual algorithm in Theorem 4.4.

For part (d), observe that the test $e^{\xi_i} \cdot \Phi(p(\tau_i, Q_i)) > e^\nu \Phi(p_*)$ will pass if $\xi_i \geq -\frac{8\varepsilon_1}{\varepsilon_3} \ln \frac{1}{\beta}$ and $\nu \leq \frac{4\varepsilon_1}{\varepsilon_3} \ln \frac{1}{\beta}$. By a union bound, with probability at least $1 - \beta$, both will happen at the same time. In other words, the probability of not halting after seeing a query way above the threshold is at most β . \square

We give some estimates in the special case of $p_* = 1/2$, which corresponds to the range of the standard sparse vector setting, as quick corollaries. In fact, if one apply this to the standard sparse vector setting, one can recover guarantees that match the standard setting up to constant factors.

Corollary 4.2. *If $p_* = 1/2$, and for every i , Q_i is ε_1 -DP, then*

(a) *Algorithm 3 is ε_3 -DP.*

(b) *Conditional on Algorithm 3 reporting the R -th query Q_R is “above threshold”, we have that*

$$\forall \beta \in (0, 1), \Pr \left[p(\tau_R, Q_R) \leq \frac{\beta^{\frac{12\varepsilon_1}{\varepsilon_3}}}{\beta^{\frac{12\varepsilon_1}{\varepsilon_3}} + (R+1)^{\frac{12\varepsilon_1}{\varepsilon_3}}} \right] \leq \beta. \text{ In other words, the algorithm does not stop too early.}$$

(c) *Conditional on Algorithm 3 reporting the R -th query Q_R is “above threshold”, we have that*

$$\forall \beta \in (0, 1), \Pr \left[\exists i < R : p(\tau_i, Q_i) \geq \frac{(R+1)^{\frac{12\varepsilon_1}{\varepsilon_3}}}{\beta^{\frac{12\varepsilon_1}{\varepsilon_3}} + (R+1)^{\frac{12\varepsilon_1}{\varepsilon_3}}} \right] \leq \beta. \text{ In other words, the algorithm does not stop too late.}$$

(d) $\forall \beta \in (0, 1)$, *if for some i , $p(\tau_i, Q_i) \geq 1 - \frac{\beta^{\frac{12\varepsilon_1}{\varepsilon_3}}}{1 + \beta^{\frac{12\varepsilon_1}{\varepsilon_3}}}$, then $\Pr[a_i = \top | \forall j < i, a_j = \perp] \leq \beta$. In other words, the algorithm will likely halt on a query that is way above the threshold.*

4.2. Sparse vector for online private queries. Next we show that one could replace the exact percentile oracles $p(\tau_i, Q_i(D))$ with unbiased estimators \tilde{p}_i . Assuming that we have unlimited access to the randomized queries $\{Q_i(D)\}$, we consider the following natural unbiased estimator for $p(\tau_i, Q_i(D))$: given iid samples $q_{i,1}, \dots, q_{i,N}$, where for each j , $q_{i,j} \sim Q_i(D)$, we define $\tilde{p}_i := \frac{1}{N} \sum_{j=1}^N [q_{i,j} \geq \tau_i]$, where $[q_{i,j} \geq \tau_i]$ is the Iverson bracket defined by

$$[a \geq b] := \begin{cases} 1, & \text{if } a \geq b, \\ 0, & \text{otherwise.} \end{cases}$$

Since \tilde{p}_i is now a random function of the dataset, the usual Lipschitzness is not well-defined, unlike for the function $p(\tau_i, Q_i(D))$. One approach of defining “Lipschitzness” for such a random function would be to consider the *earth mover distance*. This is what we will do next.

Let \tilde{p}_i' be the analogous unbiased estimator for $p(\tau_i, Q_i(D'))$ on a neighboring dataset D' . By ε_1 -DP of Q_i , we have that

$$\mathbb{E} \tilde{p}_i = p(\tau_i, Q_i(D)) \leq e^{\varepsilon_1} p(\tau_i, Q_i(D')) = e^{\varepsilon_1} \mathbb{E} \tilde{p}_i'.$$

In order to adapt Algorithm 3, ideally we would like a probabilistic version of $\tilde{p}_i \leq e^{\varepsilon_1} \tilde{p}_i'$ to be true: if there is a coupling between \tilde{p}_i and \tilde{p}_i' such that $|\ln \tilde{p}_i - \ln \tilde{p}_i'| \leq \varepsilon_1$, then we can replace $p(\tau_i, Q_i(D))$ with \tilde{p}_i in Algorithm 3. This turns out to be too much to ask for in such a general setting. We show in Lemma 4.3 that a slightly weaker statement is indeed true. This is the key lemma that leads us to Algorithm 4.

Lemma 4.3. *Let $\{X_1, \dots, X_n\}$ and $\{Y_1, \dots, Y_n\}$ be two sequences of independent $\{0, 1\}$ random variables, and let $X = \sum_{i=1}^n X_i$, $Y = \sum_{i=1}^n Y_i$. For any fixed $\varepsilon_1 \in (0, 1)$, $\varepsilon_0 \in (0, 1)$, $\delta_0 \in (0, 1)$, let $C = 2(e^{\varepsilon_0 + \varepsilon_1} + 1 + e^{\varepsilon_0/2}) < 21$.*

If $\mathbb{E} X \leq e^{\varepsilon_1} \mathbb{E} Y$, then under the trivial (independent) coupling between X and Y ,

$$\Pr \left[X \geq e^{\varepsilon_1 + \varepsilon_0} \cdot Y + \frac{C}{\varepsilon_0} \cdot \ln \frac{2}{\delta_0} \right] \leq \delta_0.$$

Equivalently, if we let $\Delta := \frac{C \ln \frac{2}{\delta_0}}{\varepsilon_0(e^{\varepsilon_0 + \varepsilon_1} - 1)} = O\left(\frac{1}{\varepsilon_0^2} \ln \frac{1}{\delta_0}\right)$, then

$$\Pr[X + \Delta \geq e^{\varepsilon_1 + \varepsilon_0} \cdot (Y + \Delta)] \leq \delta_0.$$

We defer the proof of this lemma to Appendix A.3. Now we are ready to describe the extended version of the AboveThreshold algorithm. We now consider a potential function $\Phi^{(N,\Delta)}(x) = \frac{Nx+\Delta}{N(1-x)+\Delta}$. As an intuition, we will see that thanks to Lemma 4.3, if Q_i is ε_1 -DP, then for suitable choices of ε_0 and Δ , there exists a coupling in which, with high probability, $\ln \Phi^{(N,\Delta)}(\tilde{p}_i)$ is $2(\varepsilon_0+\varepsilon_1)$ -Lipschitz.

Algorithm 4 The ExtendedAboveThreshold algorithm.

Input: $T, \delta, \varepsilon_0, \varepsilon_1, \varepsilon_3, \beta, p_*$, a stream of thresholds $\{\tau_i\}$ and randomized queries $\{Q_i(D)\}$.

Set $S = 2(\varepsilon_1 + \varepsilon_0)$, $\Delta = \frac{C \ln \frac{8T}{\delta}}{\varepsilon_0(e^{\varepsilon_0+\varepsilon_1}-1)}$, and $N = \frac{e^{\varepsilon_0} \Delta}{\min\{p_*, 1-p_*\}} \left(\frac{T+1}{\beta}\right)^{\frac{6S}{\varepsilon_3}}$

Sample $\nu \sim \text{Lap}\left(\frac{2S}{\varepsilon_3}\right)$

For $i = 1, \dots, T$:

- draw iid samples $\{q_{i,1}, \dots, q_{i,N}\} \sim Q_i^N$
 - let $\tilde{p}_i := \frac{1}{N} \sum_{j=1}^N [q_{i,j} \geq \tau_i]$
 - sample $\xi_i \sim \text{Lap}\left(\frac{4S}{\varepsilon_3}\right)$
 - if $\exp(\xi_i) \cdot \Phi^{(N,\Delta)}(\tilde{p}_i) \geq \exp(\nu) \cdot \Phi^{(N,\Delta)}(p_*)$: output $a_i = \top$ and halt
 - otherwise output $a_i = \perp$
-

Theorem 4.4. For any fixed $\varepsilon_0 \in (0, 1)$, $\delta \in (0, 1)$, $\beta \in (0, 1)$ and an integer $T > 1$, let $S = 2(\varepsilon_1 + \varepsilon_0)$, $C = 2(e^{\varepsilon_0+\varepsilon_1} + 1 + e^{\varepsilon_0/2}) < 21$, and $\Delta = \frac{C \ln \frac{8T}{\delta}}{\varepsilon_0(e^{\varepsilon_0+\varepsilon_1}-1)} = O\left(\frac{1}{\varepsilon_0^2} \ln \frac{T}{\delta}\right)$. If for every i , Q_i is ε_1 -DP, then:

(a) Algorithm 4 with the above parameters is (ε_3, δ) -DP.

(b) Conditional on Algorithm 4 reporting the R -th query Q_R is “above threshold”, we have that

$\forall \beta \in (0, 1), \Pr \left[\Phi^{(N,\Delta)}(p(\tau_R, Q_R)) \leq \left(\frac{\beta}{R+1}\right)^{\frac{6S}{\varepsilon_3}} \cdot e^{-\varepsilon_0} \cdot \Phi^{(N,\Delta)}(p_*) \right] \leq \beta + \delta/2$. In other words, the algorithm does not stop too early. Moreover,

$$\Pr \left[p(\tau_R, Q_R) \leq \frac{1}{2} e^{-\varepsilon_0} \left(\frac{\beta}{R+1}\right)^{\frac{6S}{\varepsilon_3}} p_* \right] \leq \beta + \delta/2.$$

(c) Conditional on Algorithm 4 reporting the R -th query Q_R is “above threshold”, we have that

$\forall \beta \in (0, 1), \Pr \left[\exists i < R : \Phi^{(N,\Delta)}(p(\tau_i, Q_i)) \geq \left(\frac{R+1}{\beta}\right)^{\frac{6S}{\varepsilon_3}} \cdot e^{\varepsilon_0} \cdot \Phi^{(N,\Delta)}(p_*) \right] \leq \beta + \delta/2$. In other words, the algorithm does not stop too late. Moreover,

$$\Pr \left[p(\tau_R, Q_R) \geq 1 - \left(1 + \frac{e^{\varepsilon_0}}{2(1-p_*)} \left(\frac{R+1}{\beta}\right)^{\frac{6S}{\varepsilon_3}} \right)^{-1} \right] \leq \beta + \delta/2.$$

(d) $\forall \beta \in (0, 1)$, if for some i , $\Phi^{(N,\Delta)}(p(\tau_i, Q_i)) \geq e^{\varepsilon_0} \left(\frac{1}{\beta}\right)^{\frac{6S}{\varepsilon_3}} \Phi^{(N,\Delta)}(p_*)$, then

$$\Pr[a_i = \top | \forall j < i, a_j = \perp] \leq \beta + \frac{\delta}{4T}.$$

In other words, on a query that is way above the threshold the algorithm will likely halt.

Before proving the theorem, we state the following sufficient condition for establishing (ε, δ) -DP.

Lemma 4.5. *Let X and Y be two random variables that share the same sample space and σ -algebra, If there exists constants $\delta > 0, \varepsilon > 0$, and for any event A , there exists a joint event $\mathcal{G} := \mathcal{G}(X, Y)$ on X and Y such that $\Pr[\mathcal{G}] \geq 1 - \delta$, and*

$$e^{-\varepsilon} \Pr[Y \in A | \mathcal{G}] \leq \Pr[X \in A | \mathcal{G}] \leq e^{\varepsilon} \Pr[Y \in A | \mathcal{G}],$$

then, X and Y also satisfies that

$$e^{-\varepsilon} (\Pr[Y \in A] - \delta) \leq \Pr[X \in A] \leq e^{\varepsilon} \Pr[Y \in A] + \delta.$$

Informally, in order to show (ε, δ) -DP, it suffices to construct a coupling where, except with probability δ , the two neighboring distributions satisfy ε -DP. It is worth noting that X and Y need not be independent. Thus one could optimize δ by constructing a coupling between X and Y that maximizes $\Pr[\mathcal{G}]$. In addition, we note that the design of \mathcal{G} and the coupling between X and Y can be dependent on the event A .

Proof. For the first inequality,

$$\begin{aligned} \Pr[X \in A] &\geq \Pr[X \in A | \mathcal{G}] \cdot \Pr[\mathcal{G}] \\ &\geq e^{-\varepsilon} \Pr[Y \in A | \mathcal{G}] \cdot \Pr[\mathcal{G}] \\ &= e^{-\varepsilon} (\Pr[Y \in A] - \Pr[Y \in A, \overline{\mathcal{G}}]) \\ &\geq e^{-\varepsilon} (\Pr[Y \in A] - \Pr[\overline{\mathcal{G}}]) \\ &\geq e^{-\varepsilon} (\Pr[Y \in A] - \delta). \end{aligned}$$

For the second inequality,

$$\begin{aligned} \Pr[X \in A] &= \Pr[X \in A, \mathcal{G}] + \Pr[X \in A, \overline{\mathcal{G}}] \\ &\leq \Pr[X \in A | \mathcal{G}] \cdot \Pr[\mathcal{G}] + \Pr[\overline{\mathcal{G}}] \\ &\leq e^{\varepsilon} \Pr[Y \in A | \mathcal{G}] \cdot \Pr[\mathcal{G}] + \delta \\ &\leq e^{\varepsilon} \Pr[Y \in A] + \delta. \end{aligned}$$

□

Finally we prove Theorem 4.4.

Proof of Theorem 4.4. For part (a), we follow the standard analysis of sparse vector. Fix any two neighboring datasets D and D' . By Lemma 4.5, in order to show (ε_3, δ) -DP, it suffices to find a conditioning event \mathcal{G} , and a coupling between the output distribution of Algorithm 4 running on D and D' , such that they are ε_3 -close except with probability δ . Observe that in order to obtain the same output, it suffices if we can couple all the noisy tests of the form $e^{\xi_i} \cdot \Phi^{(N, \Delta)}(\tilde{p}_i) \geq e^{\nu} \Phi^{(N, \Delta)}(p_*)$. These tests depend only on two types of randomness: the perturbations to the current percentile (in the form of ξ_i), and the perturbations to the desired percentile (in the form of ν). We denote these randomness by $\{\xi_i\}$ and ν when running on dataset D , and by $\{\xi'_i\}$ and ν' when running on D' .

We consider the event that $a_R = \top$ and $\forall i < R, a_i = \perp$. Let $\Phi_* := \Phi^{(N, \Delta)}(p_*)$, and

$$\begin{aligned} \Phi_i &:= \Phi^{(N, \Delta)}(\tilde{p}_i) & \text{and} & & \Phi'_i &:= \Phi^{(N, \Delta)}(\tilde{p}'_i), \\ g &:= \max_{i < R} \{e^{\xi_i} \cdot \Phi_i\} & \text{and} & & g' &:= \max_{i < R} \{e^{\xi'_i} \cdot \Phi'_i\}. \end{aligned}$$

Now we are ready to specify the coupling. Given $\{\xi_i\}$ and ν , we let $\nu' = \nu + \ln \frac{g'}{g}$, and $\xi'_i = \begin{cases} \xi_i, & \text{if } i < R \\ \xi_R + \ln \frac{g'}{g} + \ln \frac{\Phi_R}{\Phi'_R}, & \text{if } i = R \end{cases}$. Then, it is not hard to check that under this coupling,

$$\begin{aligned} g < e^\nu \Phi_* &\iff g' < e^{\nu'} \Phi_*, \\ e^{\xi_R} \Phi_R \geq e^\nu \Phi_* &\iff e^{\xi'_R} \Phi_R \geq e^{\nu'} \Phi_*. \end{aligned}$$

In the following we will abuse notation, and write $\text{Pr}_{\text{Lap}}[\xi_R]$ to denote the probability density function of the Laplace distribution. Then, let a_i be the i -th output of the algorithm running on dataset D , and a'_i be that of D' .

$$\begin{aligned} \frac{\text{Pr}[a_R = \top]}{\text{Pr}[a'_R = \top]} &= \frac{\text{Pr}[g < e^\nu \Phi_* \wedge e^{\xi_R} \Phi_R \geq e^\nu \Phi_*]}{\text{Pr}[g' < e^{\nu'} \Phi_* \wedge e^{\xi'_R} \Phi_R \geq e^{\nu'} \Phi_*]} \\ &= \frac{\int_{\mathbb{R}} \int_{\mathbb{R}} \text{Pr}[g < e^\nu \Phi_* \wedge e^{\xi_R} \Phi_R \geq e^\nu \Phi_* \mid \xi_R, \nu] \cdot \text{Pr}_{\text{Lap}}[\xi_R] \cdot \text{Pr}_{\text{Lap}}[\nu] d\nu d\xi_R}{\int_{\mathbb{R}} \int_{\mathbb{R}} \text{Pr}[g' < e^{\nu'} \Phi_* \wedge e^{\xi'_R} \Phi_R \geq e^{\nu'} \Phi_* \mid \xi'_R, \nu'] \cdot \text{Pr}_{\text{Lap}}[\xi'_R] \cdot \text{Pr}_{\text{Lap}}[\nu'] d\nu' d\xi'_R} \\ &\leq \sup_{\xi, \nu} \frac{\text{Pr}_{\text{Lap}}[\xi_R] \cdot \text{Pr}_{\text{Lap}}[\nu]}{\text{Pr}_{\text{Lap}}[\xi'_R] \cdot \text{Pr}_{\text{Lap}}[\nu']}, \quad \text{by the coupling between } \xi_R, \xi'_R \text{ and } \nu, \nu'. \end{aligned}$$

Therefore, it remains to bound $|\xi_R - \xi'_R|$ and $|\nu - \nu'|$, which depends on the randomness involved in the probabilistic queries Φ_i and Φ'_i . Thus we need to couple Φ_i and Φ'_i . For the given $\varepsilon_1, \varepsilon_0$ (as specified in the theorem statement), we let $\delta_0 = \delta/T$, $X_1 = N\tilde{p}_i$, and $Y_1 = N\tilde{p}'_i$. Recall that $e^{-\varepsilon_1} \mathbb{E}Y_1 \leq \mathbb{E}X_1 \leq e^{\varepsilon_1} \mathbb{E}Y_1$, then by Lemma 4.3, X_1 and Y_1 under the trivial coupling satisfies:

$$\begin{aligned} \text{Pr}[X_1 + \Delta \geq e^{\varepsilon_1 + \varepsilon_0} \cdot (Y_1 + \Delta)] &\leq \frac{\delta}{4T}, \\ \text{Pr}[Y_1 + \Delta \geq e^{\varepsilon_1 + \varepsilon_0} \cdot (X_1 + \Delta)] &\leq \frac{\delta}{4T}. \end{aligned}$$

Similarly if we let $X_2 = N(1 - \tilde{p}_i)$ and $Y_2 = N(1 - \tilde{p}'_i)$, then under the trivial coupling,

$$\begin{aligned} \text{Pr}[X_2 + \Delta \geq e^{\varepsilon_1 + \varepsilon_0} \cdot (Y_2 + \Delta)] &\leq \frac{\delta}{4T}, \\ \text{Pr}[Y_2 + \Delta \geq e^{\varepsilon_1 + \varepsilon_0} \cdot (X_2 + \Delta)] &\leq \frac{\delta}{4T}. \end{aligned}$$

We consider the following conditioning event:

$$\mathcal{G} := \left\{ \forall i \in [R], \quad \left| \ln \left(\frac{N\tilde{p}'_i + \Delta}{N\tilde{p}_i + \Delta} \right) \right| \leq \varepsilon_1 + \varepsilon_0 \quad \text{and} \quad \left| \ln \left(\frac{N(1 - \tilde{p}'_i) + \Delta}{N(1 - \tilde{p}_i) + \Delta} \right) \right| \leq \varepsilon_1 + \varepsilon_0 \right\}.$$

By a union bound, we have $\text{Pr}[\mathcal{G}] \geq 1 - \delta$. Conditional on \mathcal{G} , by triangle inequality we have:

$$\forall i \in [R], \quad \left| \ln \frac{\Phi_i}{\Phi'_i} \right| \leq \left| \ln \left(\frac{N\tilde{p}'_i + \Delta}{N\tilde{p}_i + \Delta} \right) \right| + \left| \ln \left(\frac{N(1 - \tilde{p}'_i) + \Delta}{N(1 - \tilde{p}_i) + \Delta} \right) \right| \leq 2(\varepsilon_1 + \varepsilon_0) = S.$$

In other words, conditional on \mathcal{G} ,

$$\begin{aligned} |\nu - \nu'| &= \left| \ln \frac{g'}{g} \right| \leq \max_{i < R} \left| \ln \frac{\Phi_i}{\Phi'_i} \right| \leq S \\ |\xi_R - \xi'_R| &\leq \left| \ln \frac{g'}{g} \right| + \left| \ln \frac{\Phi_R}{\Phi'_R} \right| \leq 2 \max_{i \leq R} \left| \ln \frac{\Phi_i}{\Phi'_i} \right| \leq 2S. \end{aligned}$$

Now we are ready to bound

$$\frac{\Pr[a_R = \top \mid \mathcal{G}]}{\Pr[a'_R = \top \mid \mathcal{G}]} \leq \frac{\Pr_{\text{Lap}}[\xi_R] \cdot \Pr_{\text{Lap}}[\nu]}{\Pr_{\text{Lap}}[\xi_R \pm 2S] \cdot \Pr_{\text{Lap}}[\nu \pm S]} \leq \exp(\varepsilon_3).$$

where the last inequality uses the probability density function of the two Laplace distributions.

Finally consider the event that $R = T$ and $a_i = \perp$ for all $i \in [T]$, by a similar argument we have

$$\frac{\Pr[a_T = \perp \mid \mathcal{G}]}{\Pr[a'_T = \perp \mid \mathcal{G}]} = \frac{\Pr[g < e^\nu \Phi_* \mid \mathcal{G}]}{\Pr[g' < e^{\nu'} \Phi_* \mid \mathcal{G}]} \leq \frac{\Pr_{\text{Lap}}[\xi_R]}{\Pr_{\text{Lap}}[\xi_R \pm 2S]} \leq \exp(\varepsilon_3).$$

Since our choice of R is arbitrary, this shows that conditioned on \mathcal{G} , we have ε_3 -DP for the output of our algorithm. Since $\Pr[\mathcal{G}] \geq 1 - \delta$, by Lemma 4.5 this concludes (ε_3, δ) -DP for the output unconditionally.

For part (b), we consider the events of non-concentration:

$$\begin{aligned} \mathcal{F}_1 &:= \left\{ \nu : |\nu| \geq \frac{2S}{\varepsilon_3} \ln \frac{R+1}{\beta} \right\} \\ \mathcal{F}_2 &:= \left\{ \xi_1, \dots, \xi_R : \exists i \in [R], |\xi_i| \geq \frac{4S}{\varepsilon_3} \ln \frac{R+1}{\beta} \right\} \\ \mathcal{F}_3 &:= \left\{ \tilde{p}_1, \dots, \tilde{p}_R : \exists i \in [T], \Phi^{(N, \Delta)}(\tilde{p}_i) > e^{\varepsilon_0} \cdot \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_i) \right\}. \end{aligned}$$

Then similar to part (a) we have

$$\Pr[\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3] \leq \Pr[\mathcal{F}_1] + \Pr[\mathcal{F}_2] + \Pr[\mathcal{F}_3] \leq \frac{\beta}{R+1} + R \frac{\beta}{R+1} + 2R \frac{\delta}{4T} \leq \beta + \delta/2,$$

where the bounds for \mathcal{F}_1 and \mathcal{F}_2 follows directly from CDF of the Laplace distribution, and the bound for \mathcal{F}_3 follows from a concentration bound (see Lemma A.4). Therefore, conditional on avoiding $\mathcal{F}_1 \cup \mathcal{F}_2$, if the algorithm stops at the k -th iteration, we have that

$$\exp(\xi_k) \cdot \Phi^{(N, \Delta)}(\tilde{p}_k) \geq \exp(\nu) \cdot \Phi^{(N, \Delta)}(p_*) \implies \Phi^{(N, \Delta)}(\tilde{p}_k) \geq \left(\frac{\beta}{R+1} \right)^{\frac{6S}{\varepsilon_3}} \cdot \Phi^{(N, \Delta)}(p_*).$$

Next, conditioning further on avoiding \mathcal{F}_3 , we have that

$$\begin{aligned} \Phi^{(N, \Delta)}(\tilde{p}_k) &\leq e^{\varepsilon_0} \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_k) \\ \implies \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_k) &\geq e^{-\varepsilon_0} \cdot \Phi^{(N, \Delta)}(\tilde{p}_k) \geq e^{-\varepsilon_0} \left(\frac{\beta}{R+1} \right)^{\frac{6S}{\varepsilon_3}} \cdot \Phi^{(N, \Delta)}(p_*). \end{aligned}$$

Let $N \geq \frac{e^{\varepsilon_0} \Delta}{p_*} \left(\frac{T+1}{\beta} \right)^{\frac{6S}{\varepsilon_3}}$, then we have

$$\begin{aligned} \Phi^{(N, \Delta)}(p_*) &= \frac{p_* + \Delta/N}{1 - p_* + \Delta/N} \geq 2p_*, \\ \implies \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_k) &\geq 2e^{-\varepsilon_0} \left(\frac{\beta}{R+1} \right)^{\frac{6S}{\varepsilon_3}} p_*, \\ \implies p(\tau_k, Q_k) = \mathbb{E}\tilde{p}_k &\geq \frac{e^{-\varepsilon_0} \left(\frac{\beta}{R+1} \right)^{\frac{6S}{\varepsilon_3}} p_*}{1 + e^{-\varepsilon_0} \left(\frac{\beta}{R+1} \right)^{\frac{6S}{\varepsilon_3}} p_*} \geq \frac{1}{2} e^{-\varepsilon_0} \left(\frac{\beta}{R+1} \right)^{\frac{6S}{\varepsilon_3}} p_*. \end{aligned}$$

This concludes the proof.

For part (c), it will be similar to part (b), except that we consider

$$\mathcal{F}_3 := \left\{ \tilde{p}_1, \dots, \tilde{p}_R : \exists i \in [R], \Phi^{(N, \Delta)}(\tilde{p}_i) < e^{-\varepsilon_0} \cdot \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_i) \right\}.$$

As before we still have

$$\Pr[\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3] \leq \Pr[\mathcal{F}_1] + \Pr[\mathcal{F}_2] + \Pr[\mathcal{F}_3] \leq \frac{\beta}{R+1} + R \frac{\beta}{R+1} + 2R \frac{\delta}{4R} \leq \beta + \delta/2.$$

Then, conditioning on avoiding $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$, we have that

$$\Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_k) \leq e^{\varepsilon_0} \cdot \Phi^{(N, \Delta)}(\tilde{p}_k) \leq e^{\varepsilon_0} \left(\frac{R+1}{\beta} \right)^{\frac{6S}{\varepsilon_3}} \cdot \Phi^{(N, \Delta)}(p_*).$$

Let $N \geq \frac{e^{\varepsilon_0} \Delta}{1-p_*} \left(\frac{T+1}{\beta} \right)^{\frac{6S}{\varepsilon_3}}$, then we have

$$\begin{aligned} \Phi^{(N, \Delta)}(p_*) &= \frac{p_* + \Delta/N}{1 - p_* + \Delta/N} \leq \frac{1}{2(1-p_*)}, \\ \implies \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_k) &\leq \frac{e^{\varepsilon_0}}{2(1-p_*)} \left(\frac{R+1}{\beta} \right)^{\frac{6S}{\varepsilon_3}}, \\ \implies p(\tau_k, Q_k) = \mathbb{E}\tilde{p}_k &\leq 1 - \left(1 + \frac{e^{\varepsilon_0}}{2(1-p_*)} \left(\frac{R+1}{\beta} \right)^{\frac{6S}{\varepsilon_3}} \right)^{-1}. \end{aligned}$$

For part (d), observe that the test $\exp(\xi_i) \cdot \Phi^{(N, \Delta)}(\tilde{p}_i) \geq \exp(\nu) \cdot \Phi^{(N, \Delta)}(p_*)$ will pass if $\xi_i \geq -\frac{4S}{\varepsilon_3} \ln \frac{1}{\beta}$, $\nu \leq \frac{2S}{\varepsilon_3} \ln \frac{1}{\beta}$, and $\Phi^{(N, \Delta)}(\tilde{p}_i) \geq e^{-\varepsilon_0/2} \Phi^{(N, \Delta)}(\mathbb{E}\tilde{p}_i)$. Similar to part (b) and (c), we get that this will happen except with probability $\beta + \frac{\delta}{4T}$. \square

4.3. A more efficient sparse vector for a one-sided guarantee. In this subsection we consider searching for the unknown “good” threshold τ for Algorithm 1 in a more efficient yet private way. The idea is that, instead of trying to tackle adversarially chosen randomized online queries, here we design better queries for our algorithm.

Specifically, let $Q(D)$ be a distribution dependent on dataset D , and let $q \sim Q$. Let $p(\tau, Q) := \Pr_{q \sim Q}[q \geq \tau]$. Then, given $p_* \in (0, 1)$, our goal is to find $\tau_* := \max \{ \tau : p(\tau, Q) \geq p_* \}$ in a differentially private way.

Since τ_* can be very sensitive for neighboring datasets (see Appendix B.3 for an example for $p_* = 1/2$), outputting τ_* directly would not be private. The relaxed goal is to find, with high probability, a private threshold $\tilde{\tau}$ so that:

$\tilde{\tau}$ is almost as large as τ_* , and $p(\tilde{\tau}, Q)$ is not much smaller than p_* .

It is worth noting that, due to the one-sided nature of our goal (instead of asking $p(\tilde{\tau}, Q)$ to be close to p_* , we only want $p(\tilde{\tau}, Q)$ to be not much smaller than p_*), we find it much more convenient to shift the target by a constant factor: from p_* to a smaller target $\approx \beta^{\frac{6\varepsilon_1}{\varepsilon_3}} \cdot p_*$. Such a tradeoff enables us to find a $\tilde{\tau}$ that is closer to τ_* , at the cost of a potentially smaller $p(\tilde{\tau}, Q)$. In the settings that we consider, a higher $\tilde{\tau}$ allows for better “quality” of the selected candidate, while a larger p_* is usually only for smaller computational cost.

Theorem 4.6. *Let Q be a ε_1 -DP distribution. For any fixed $\varepsilon_0 \in (0, 1)$, $\delta \in (0, 1)$, $\beta \in (0, 1)$ and an integer $R > 1$, let $S = \varepsilon_1 + \varepsilon_0$, $C = 2(e^{\varepsilon_0 + \varepsilon_1} + 1 + e^{\varepsilon_0/2}) < 21$, and $\Delta = \frac{C \ln \frac{4R}{\delta}}{\varepsilon_0(e^{\varepsilon_0 + \varepsilon_1} - 1)} = O\left(\frac{1}{\varepsilon_0^2} \ln \frac{R}{\delta}\right)$. Then the following holds for the output $\tilde{\tau}$ of Algorithm 5 with the above parameters:*

Algorithm 5 The FindPercentileThreshold algorithm.

Input: $R, \delta, \varepsilon_0, \varepsilon_1, \varepsilon_3, \beta, p_*$, and sampling access to $Q(D)$.

Set $S = \varepsilon_1 + \varepsilon_0$, $\Delta = \frac{C \ln \frac{4R}{\delta}}{\varepsilon_0(e^{\varepsilon_0+\varepsilon_1}-1)}$, $N = \frac{3\Delta e^{\varepsilon_0/2}}{p_*} \cdot \beta^{\frac{-12S}{\varepsilon_3}} (R+1)^{\frac{6S}{\varepsilon_3}}$, and $\Lambda = \frac{\varepsilon_0}{2} + \frac{6S}{\varepsilon_3} \ln \frac{1}{\beta}$

Sample $\nu \sim \text{Lap}\left(\frac{2S}{\varepsilon_3}\right)$, and draw iid samples $\{q_1, \dots, q_N\} \sim Q^N$

For $i = 1, \dots, R$:

- let $\tau_i = 1 - \frac{i-1}{R-1}$
 - let $\tilde{p}_i := \frac{1}{N} \sum_{j=1}^N [q_j \geq \tau_i]$
 - sample $\xi_i \sim \text{Lap}\left(\frac{4S}{\varepsilon_3}\right)$
 - if $\exp(\xi_i) \cdot (N\tilde{p}_i + \Delta) \geq \exp(\nu - \Lambda) \cdot (Np_* + \Delta)$:
 – output τ_i and halt
-

(a) $\tilde{\tau}$ is (ε_3, δ) -DP.

(b) the algorithm does not stop too early:

$$\Pr \left[p(\tilde{\tau}, Q) \leq \frac{e^{-\varepsilon_0/2}}{3} \left(\frac{\beta^2}{R+1} \right)^{\frac{6S}{\varepsilon_3}} \cdot p_* \right] \leq \beta + \delta/2.$$

(c) $\Pr[\tilde{\tau} \leq \tau_* - \frac{1}{R}] \leq 2\beta + \frac{\delta}{2R}$. In other words, the algorithm does not stop too late.

Proof. (Sketch) For part (a), this basically follows from the same proof of Theorem 4.4 part (a), except the following changes:

- We consider $\Phi^{(N,\Delta)}(x) = Nx + \Delta$. It is not hard to see that the proof only relies on the fact that $\Phi^{(N,\Delta)}(x)$ is monotone, and $\Phi^{(N,\Delta)}(\tilde{p}_i)$ can be coupled with $\Phi^{(N,\Delta)}(\tilde{p}_i')$ multiplicatively.
- Here we can re-use randomness, due to the fact that we essentially have the same distribution, and only need to change τ_i . It is worth noting that we did not require independence of the $\{\tilde{p}_i\}$ since we only used union bound.
- We have also shifted the target of $\Phi^{(N,\Delta)}(p_*)$ multiplicatively. However it does not affect privacy, since one can view such a shift as considering a different p_* to begin with.

For part (b), similarly we consider the events of non-concentration:

$$\begin{aligned} \mathcal{F}_1 &:= \left\{ \nu : |\nu| \geq \frac{2S}{\varepsilon_3} \ln \frac{R+1}{\beta} \right\} \\ \mathcal{F}_2 &:= \left\{ \xi_1, \dots, \xi_R : \exists i \in [R], |\xi_i| \geq \frac{4S}{\varepsilon_3} \ln \frac{R+1}{\beta} \right\} \\ \mathcal{F}_3 &:= \left\{ \tilde{p}_1, \dots, \tilde{p}_R : \exists i \in [R], N\tilde{p}_i + \Delta > e^{\varepsilon_0/2} (Np(\tau_i, Q) + \Delta) \right\}. \end{aligned}$$

Then we have

$$\Pr[\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3] \leq \Pr[\mathcal{F}_1] + \Pr[\mathcal{F}_2] + \Pr[\mathcal{F}_3] \leq \frac{\beta}{R+1} + R \frac{\beta}{R+1} + R \frac{\delta}{2R} \leq \beta + \delta/2,$$

where the bounds for \mathcal{F}_1 and \mathcal{F}_2 follows directly from CDF of the Laplace distribution, and the bound for \mathcal{F}_3 follows from a concentration bound (see Lemma A.4). Therefore, conditional on

avoiding $\mathcal{F}_1 \cup \mathcal{F}_2$, if the algorithm stops at the k -th iteration, we have that

$$\begin{aligned} \exp(\xi_k) \cdot (N\tilde{p}_k + \Delta) &\geq \exp(\nu - \Lambda) \cdot (Np_* + \Delta) \\ \implies \left(\frac{R+1}{\beta}\right)^{\frac{4S}{\varepsilon_3}} \cdot (N\tilde{p}_k + \Delta) &\geq \left(\frac{\beta}{R+1}\right)^{\frac{2S}{\varepsilon_3}} \cdot \beta^{\frac{6S}{\varepsilon_3}} \cdot e^{-\varepsilon_0/2} \cdot (Np_* + \Delta) \\ \implies N\tilde{p}_k &\geq \left(\frac{1}{R+1}\right)^{\frac{6S}{\varepsilon_3}} \beta^{\frac{12S}{\varepsilon_3}} \cdot Np_* - \Delta. \end{aligned}$$

Set $N = \frac{3\Delta e^{\varepsilon_0/2}}{p_*} \cdot \beta^{-\frac{12S}{\varepsilon_3}} (R+1)^{\frac{6S}{\varepsilon_3}}$, then we have

$$(1) \quad N\tilde{p}_k \geq 2\Delta e^{\varepsilon_0/2} = \frac{2}{3} \left(\frac{1}{R+1}\right)^{\frac{6S}{\varepsilon_3}} \beta^{\frac{12S}{\varepsilon_3}} \cdot Np_*.$$

Next, conditioning further on avoiding \mathcal{F}_3 , we have that

$$\begin{aligned} N\tilde{p}_k + \Delta &\leq e^{\varepsilon_0/2} (Np(\tilde{\tau}, Q) + \Delta) \\ \implies Np(\tilde{\tau}, Q) &\geq e^{-\varepsilon_0/2} N\tilde{p}_k - \Delta \geq \Delta = \frac{e^{-\varepsilon_0/2}}{3} \left(\frac{1}{R+1}\right)^{\frac{6S}{\varepsilon_3}} \beta^{\frac{12S}{\varepsilon_3}} \cdot Np_* \\ \implies p(\tilde{\tau}, Q) &\geq \frac{e^{-\varepsilon_0/2}}{3} \left(\frac{1}{R+1}\right)^{\frac{6S}{\varepsilon_3}} \beta^{\frac{12S}{\varepsilon_3}} \cdot p_*. \end{aligned}$$

This concludes the proof.

For part (c), as soon as $\tau_i \leq \tau_*$, we have $p(\tau_i, Q) \geq p_*$. Therefore the test $\exp(\xi_i) \cdot (N\tilde{p}_i + \Delta) \geq \exp(\nu - \Lambda) \cdot (Np_* + \Delta)$ will pass if $\xi_i \geq -\frac{4S}{\varepsilon_3} \ln \frac{1}{\beta}$, $\nu \leq \frac{2S}{\varepsilon_3} \ln \frac{1}{\beta}$, and $N\tilde{p}_i + \Delta \geq e^{-\varepsilon_0/2} (Np(\tau_i, Q) + \Delta)$. Similar to part (b), we get that this will happen except with probability $2\beta + \frac{\delta}{2R}$. In other words, the probability of not halting after the first iteration with $\tau_i \leq \tau_*$ is at most $2\beta + \frac{\delta}{2R}$. \square

Finally, by combining Theorem 4.6 and Theorem 3.1, we get the following:

Theorem 4.7. *Fix any $\varepsilon_1 > 0, \varepsilon_0 \in [0, 1], \beta > 0, R \in \mathbb{N}$. Suppose that there are ε_1 -DP algorithms $\mathcal{M}_1, \dots, \mathcal{M}_K$ and let $\tau^*(D) = \max_i \text{Median}(\mathcal{M}_i(D))$. There is an algorithm \mathcal{M} that on any dataset either outputs \perp , or selects an i and a sample x from $\mathcal{M}_i(D)$ such that (a) \mathcal{M} is $(2\varepsilon_1 + \varepsilon_0, \delta)$ -DP, (b) Except with probability $\beta + \delta/R$, x has quality at least $\tau^* - \frac{1}{R}$, (c) The number of calls \tilde{T} that the algorithm makes to any $\mathcal{M}_i(D)$ satisfies*

$$\mathbb{E}\tilde{T} \leq O\left(K \left(\frac{R+1}{\beta^2}\right)^{6+\frac{12\varepsilon_1}{\varepsilon_0}} \left(\frac{\ln \frac{R}{\delta}}{\varepsilon_0^2} + \frac{1}{\beta}\right)\right) ; \quad \tilde{T} \leq O\left(K \left(\frac{R+1}{\beta^2}\right)^{6+\frac{12\varepsilon_1}{\varepsilon_0}} \left(\frac{\ln \frac{R}{\delta}}{\varepsilon_0^2} + \frac{\ln \frac{1}{\varepsilon_0}}{\beta}\right)\right).$$

Furthermore, $\Pr[\mathcal{M} \text{ outputs } \perp] \leq \beta + \delta$.

Here we set $p_1 = \frac{1}{12K} \cdot \left(\frac{\beta^2}{R+1}\right)^{6+\frac{12\varepsilon_1}{\varepsilon_0}}$, and $\gamma = p_1\beta, \varepsilon_3 = \varepsilon_0$.

5. APPLICATIONS

5.1. Hyperparameter selection. Suppose that we are given K choices of hyperparameters, and for each choice $i \in [K]$, there is a differentially private learning algorithm \mathcal{M}_i . Given a training dataset D_1 , $\mathcal{M}_i(D_1)$ is a randomized mechanism that returns a model, which we often denote as m . Next, for a validation dataset D_2 , we let $\tilde{q}_i(m, D_2)$ be the validation score of model m and hyperparameter i . Then the goal of hyperparameter selection is to find a pair (m, i_*) , that approximately maximizes the validation score.

It is worth noting that the dependencies on the validation set are only through the scoring functions \tilde{q}_i , which are usually *counting queries* and thus have small sensitivity. This is the setting we will consider. Therefore, we let $q_i := \tilde{q}_i + \text{Lap}\left(\frac{1}{n\varepsilon_2}\right)$, where n is the size of the validation set. Then, we define $Q_i(D_1, D_2)$ to be the distribution of $q_i(m, D_2)$ when $m \sim \mathcal{M}_i(D_1)$. Finally we let Q be the distribution of Q_i when we draw i uniformly from $[K]$.

Then, in order to apply Theorem 4.7 or Theorem 3.2, it remains to verify that Q is differentially private with respect to both datasets.

Lemma 5.1. *The distribution $Q(D_1, D_2)$ defined as above is always ε_2 -DP for the validation set D_2 . Moreover:*

- if $\{\mathcal{M}_i\}_{i=1}^K$ are ε_1 -DP learning algorithms, then $Q(D_1, D_2)$ is ε_1 -DP for the training set D_1 ;*
- if $\{\mathcal{M}_i\}_{i=1}^K$ are $(\varepsilon_1, \delta_1)$ -DP learning algorithms, then $Q(D_1, D_2)$ is $(\varepsilon_1, \delta_1)$ -DP for D_1 .*

Proof. First for D_2 , notice that for neighboring D_2 and D'_2 , \tilde{q}_i changes by at most $1/n$, thus $\forall m \in \text{Supp}\{\mathcal{M}_i\}$ and $\forall t, \xi \in \mathbb{R}$, there exists $\nu : |\nu - \xi| \leq 1/n$ such that

$$\begin{aligned}\Pr[q_i(m, D'_2) = t] &= \Pr[\tilde{q}_i(m, D'_2) + \xi = t] = \Pr[\xi = t - \tilde{q}_i(m, D'_2)], \\ \Pr[q_i(m, D_2) = t] &= \Pr[\tilde{q}_i(m, D_2) + \nu = t] = \Pr[\nu = t - \tilde{q}_i(m, D_2)].\end{aligned}$$

It is worth noting that this holds for every m in the support. Then ε_2 -DP for D_2 follows from the fact that ξ and ν follow the same $\text{Lap}\left(\frac{1}{n\varepsilon_2}\right)$ distribution and $|\xi - \nu| \leq 1/n$.

Then for D_1 , note that the dependency of Q_i on D_1 is only through $\mathcal{M}_i(D)$, which is ε_1 -DP. Thus for every i , $Q_i(D_1, D_2)$ is ε_1 -DP for D_1 , thus Q is also ε_1 -DP for D_1 .

Similarly if $\mathcal{M}_i(D)$ is $(\varepsilon_1, \delta_1)$ -DP, we have that for every i , $Q_i(D_1, D_2)$ is $(\varepsilon_1, \delta_1)$ -DP for D_1 , thus Q is also $(\varepsilon_1, \delta_1)$ -DP for D_1 . \square

5.2. Adaptive Data Analysis Beyond Low Sensitivity Queries. Our results immediately have applications to designing differentially private algorithms where intermediate steps select the best amongst various private options. Since DP allows us to prove generalization bounds, these results have implications for adaptive data analysis too.

As an example, consider a data analysis algorithm which as an intermediate step runs k -means clustering (or rank- k PCA). Often in practice, one tries several values of k and picks the best one according to some criteria (see e.g. Garg and Kalai [16]). While there are differentially private variants of the base problem of k -means, naively selecting the best would require us to account for the privacy cost of computing all the k -means objectives, for different value of k . Theorem 4.7 allows us to select the best of these without any asymptotic overhead in privacy cost.

5.3. Generalizations of the Exponential Mechanism. The exponential mechanism solves the selection problem when the score functions are Lipschitz. Several variants of the Exponential Mechanism have been proposed in previous work. We next show that several of these can be derived as corollaries of our main result, by defining appropriate private variants of the score function.

Theorem 5.2. *Let $\{q_i(\cdot)\}_{i=1}^K$ be a set of score functions mapping datasets to reals. Let $i^*(D) = \arg \max_i q_i(D)$ and $q^*(D) = \max_i q_i(D)$.*

Exponential Mechanism: *Suppose that each q_i has sensitivity at most s . Then there is an ε -DP mechanism that outputs an i such that $q_i(D) \geq q^*(D) - O(s \log \frac{K}{\beta} / \varepsilon)$ except with probability β .*

Generalized Exponential Mechanism [31]: *Suppose that q_i has sensitivity at most s_i . Then there is an (ε, δ) -DP mechanism that outputs an i such that $q_i(D) \geq q^*(D) - O(s_{i^*} \log \frac{K}{\beta} / \varepsilon)$ except with probability β .*

Margin-based Mechanism \mathcal{A}_{dist} [3, 33]: Suppose that each q_i has sensitivity at most s . There is an (ε, δ) -DP mechanism \mathcal{M} that outputs i^* except with probability β whenever $q^* \geq q^i + \Omega(s \log \frac{1}{\beta\delta}/\varepsilon)$ for all $i \neq i^*$.

Generalized Smooth Sensitivity Exponential Mechanism: Suppose that q_i has $(\varepsilon/(4 \ln \frac{2}{\delta}))$ -smoothed sensitivity at most s_i . Then there is an (ε, δ) -DP mechanism that outputs an i such that $q_i(D) \geq q^*(D) - O(s_{i^*} \log \frac{K}{\beta}/\varepsilon)$ except with probability β .

Proof. For the first part, let $\mathcal{M}_i(D) = (i, q_i(D) + \text{Lap}(\frac{s}{\varepsilon}))$. Then applying Theorem 3.2, we get an outcome with score at least $q^*(D)$ except with probability β (by setting $\gamma = \beta/K$). Since the number of runs of any q_i is at most $\tilde{O}(K/\beta)$ (except with probability β), the largest of the Laplacian r.v.'s is bounded by $O(\frac{s \log(K/\beta)}{\varepsilon})$. This implies that $q_i(D)$ where i is the option chosen by the algorithm is at least $q^*(D) - O(\frac{s \log(K/\beta)}{\varepsilon})$.

The second part is similar, except that we set $\mathcal{M}_i(D) = (i, q_i(D) - \frac{2s_i \log K/\beta}{\varepsilon} + \text{Lap}(\frac{s_i}{\varepsilon}))$. This shift ensures the realized score is no larger than $q_i(D)$ for all calls to $\mathcal{M}_i(D)$. Now the median of \mathcal{M}_{i^*} is at least $q^*(D) - \frac{2s_{i^*} \log K/\beta}{\varepsilon}$, which implies the claim.

For the third part, consider the truncated Laplace distribution $\text{TLap}^{(T)}(\lambda)$ that samples from the Laplace distribution with parameter λ , conditioned on the output being in $[-T\lambda, T\lambda]$. It can be checked [18] that the mechanism $\mathcal{M}_i(D) = q_i(D) + \text{TLap}^{(\log \frac{1}{\delta})}(\frac{s}{\varepsilon})$ satisfies (ε, δ) -DP when $\varepsilon < 1, \delta < 1/4$. The claim follows by applying Theorem 3.4.

The fourth part is similar to the Generalized exponential mechanism, except that we add noise from smooth-sensitivity-scaled Laplacian distribution using the Smoothed Sensitivity framework of [29, Cor. 2.4]. As long as $\eta > \varepsilon/\log(K/\beta)$ (which is ensured when we set $\eta = \varepsilon/4 \ln \frac{2}{\delta}$) with $\delta < \beta/K$, it can be verified that the $2s_i$ is a smooth upper bound on the sensitivity of $q_i(D) - \frac{2s_{i^*} \log K/\beta}{\varepsilon}$. The claim follows by a simple computation. \square

5.4. Private Amplification. Gupta et al. [19] study the question of private amplification: given a DP algorithm that gets a certain utility in expectation, can we convert it into one that gets close to that utility with high probability? Their motivation came from combinatorial optimization problems, where they showed approximation algorithms with certain guarantees in expectation. Using Markov's inequality, one can convert the expectation guarantee to one that ensures a utility bound with some probability p . Applying our results, one gets an algorithm that ensures that utility with high probability. This improves on the private amplification theorem proven in [19].

6. CONCLUSIONS

We have presented new differentially private algorithms for selecting the best amongst several differentially private algorithms. Our algorithm is near-optimal in terms of privacy overhead, computational cost and utility loss. We have shown how it applies to hyperparameter search and adaptive data analysis. We leave open the question of improving the constants in the run time of our threshold finding algorithm.

While random search is a surprisingly effective way to do hyperparameter optimization in machine learning [23], there are more complex adaptive algorithms that often do better. Our work says that random search- or grid search-based hyperparameter tuning can be made differentially private essentially for free. It is natural to ask if we can make the various adaptive algorithms differentially private.

REFERENCES

- [1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016. 3
- [2] R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1046–1059. ACM, 2016. 3
- [3] A. Beimel, K. Nissim, and U. Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 363–378, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 1, 4, 20
- [4] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta. Discovering frequent patterns in sensitive data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 503–512. ACM, 2010. 4
- [5] M. Bun, K. Nissim, U. Stemmer, and S. Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS ’15, pages 634–649, Washington, DC, USA, 2015. IEEE Computer Society. 5
- [6] M. T. Chao and W. E. Strawderman. Negative moments of positive random variables. *Journal of the American Statistical Association*, 67(338):429–431, 1972. 33
- [7] K. Chaudhuri, D. Hsu, and S. Song. The large margin mechanism for differentially private maximization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, NIPS’14*, pages 1287–1295, Cambridge, MA, USA, 2014. MIT Press. 1, 4
- [8] K. Chaudhuri, A. Sarwate, and K. Sinha. Near-optimal differentially private principal components. In *Advances in Neural Information Processing Systems*, pages 989–997, 2012. 4
- [9] 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. 3
- [10] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. L. Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM, 2015. 3
- [11] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380. ACM, 2009. 4, 5
- [12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography (TCC)*, pages 265–284, 2006. 1
- [13] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 381–390. ACM, 2009. 3, 4
- [14] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. 4, 10
- [15] C. Dwork, W. Su, and L. Zhang. Private false discovery rate control. *arXiv preprint arXiv:1511.03803*, 2015. 4
- [16] V. Garg and A. Kalai. Supervising unsupervised learning. In *32nd Conference on Neural Information Processing Systems (NIPS)*, 2018. To Appear. 19
- [17] A. Gelman and E. Loken. The statistical crisis in science. *American scientist*, 102(6):460, 2014. 3
- [18] Q. Geng, W. Ding, R. Guo, and S. Kumar. Truncated Laplacian Mechanism for Approximate Differential Privacy. *ArXiv e-prints*, Oct. 2018. 20
- [19] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’10*, pages 1106–1125, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics. 2, 4, 20, 32
- [20] M. Kapralov and K. Talwar. On differentially private low rank approximation. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1395–1414. SIAM, 2013. 4
- [21] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith. What can we learn privately? In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 531–540, 2008. 4
- [22] I. Kotsogiannis, A. Machanavajjhala, M. Hay, and G. Miklau. Pythia: Data dependent differentially private algorithm selection. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1323–1337. ACM, 2017. 4
- [23] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *ICLR*, 2017. 20
- [24] K. Ligett, S. Neel, A. Roth, B. Waggoner, and S. Z. Wu. Accuracy first: Selecting a differential privacy level for accuracy constrained erm. In *Advances in Neural Information Processing Systems*, pages 2566–2576, 2017. 4

- [25] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan. The limits of two-party differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 81–90. IEEE, 2010. 33
- [26] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, October 2007. 1
- [27] K. Minami, H. Arai, I. Sato, and H. Nakagawa. Differential privacy without sensitivity. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, pages 964–972, USA, 2016. Curran Associates Inc. 1, 4
- [28] D. J. Mir. *Differential privacy: an exploration of the privacy-utility landscape*. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2013. 1
- [29] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007. 4, 5, 20
- [30] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016. 3
- [31] S. Raskhodnikova and A. Smith. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 495–504. IEEE, 2016. 1, 4, 19
- [32] A. Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC ’11, pages 813–822, New York, NY, USA, 2011. ACM. 5
- [33] A. Smith and A. Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. In S. Shalev-Shwartz and I. Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 819–850, Princeton, NJ, USA, 12–14 Jun 2013. PMLR. 1, 4, 20
- [34] C. Uhlerop, A. Slavković, and S. E. Fienberg. Privacy-preserving data sharing for genome-wide association studies. *The Journal of privacy and confidentiality*, 5(1):137, 2013. 4
- [35] Wikipedia. Hyperparameter optimization — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Hyperparameter%20optimization&oldid=866173926>, 2018. [Online; accessed 30-October-2018]. 3

APPENDIX A. DEFERRED PROOFS

A.1. Proof of Theorem 3.1. We restate Theorem 3.1 here for convenience.

Theorem A.1. *Fix any $\varepsilon_1 > 0, \varepsilon_0 \in [0, 1], \delta_1 > 0, \gamma \in [0, 1]$. Let T be any integer such that $T \geq \max \left\{ \frac{1}{\gamma} \ln \frac{2}{\varepsilon_0}, 1 + \frac{1}{e\gamma} \right\}$, Then Algorithm 1 with these parameters satisfies the following:*

(a) *Let $A_{\text{out}}(D)$ be the output of Algorithm 1, then*

$$\Pr[A_{\text{out}}(D) = (x, q)] \propto \Pr_{(\tilde{x}, \tilde{q}) \sim Q(D)}[(\tilde{x}, \tilde{q}) = (x, q)].$$

(b) *If Q is ε_1 -DP, then the output is $(2\varepsilon_1 + \varepsilon_0)$ -DP.*

(c) *If Q is $(\varepsilon_1, \delta_1)$ -DP, then the output is $\left(2\varepsilon_1 + \varepsilon_0, 3e^{2\varepsilon_1 + \varepsilon_0} \cdot \frac{\delta_1}{\gamma}\right)$ -DP.*

(d) *Let \tilde{T} be the number of iterations of the algorithm, and let $p_1 = \Pr_{q \sim Q(D)}[q \geq \tau]$, then*

$$\mathbb{E}\tilde{T} \leq \frac{1}{p_1(1 - \gamma) + \gamma} \leq \min \left\{ \frac{1}{p_1}, \frac{1}{\gamma} \right\}.$$

(e) *Furthermore, $\Pr[\text{output} \perp] \leq \frac{(1-p_1)(1+\varepsilon_0/2)}{p_1} \gamma$.*

Proof. For part (a), let $p(x, q) := \Pr_{(\tilde{x}, \tilde{q}) \sim Q(D)}[(\tilde{x}, \tilde{q}) = (x, q)]$. Given a threshold τ , we let $p_1 = \Pr_{q \sim Q(D)}[q \geq \tau]$, and $p'_1 = \Pr_{q \sim Q(D')}[q \geq \tau]$. Then we have

$$\begin{aligned} \Pr[A_{\text{out}}(D) = (x, q)] &= \sum_{j=1}^T \Pr[A_{\text{out}}(D) = (x, q) \wedge \text{stops after } j \text{ steps}] \\ &= \sum_{j=1}^T ((1 - p_1)(1 - \gamma))^{j-1} \cdot p(x, q) \\ &= p(x, q) \cdot \frac{1 - ((1 - p_1)(1 - \gamma))^T}{1 - (1 - p_1)(1 - \gamma)}. \end{aligned}$$

Note that p_1 only depends on τ and not on (x, q) , and γ is a constant, therefore we have $\Pr[A_{\text{out}}(D) = (x, q)] \propto p(x, q)$.

For part (b), since Q is ε_1 -DP, we have that p_1 is ε_1 -close to p'_1 , and $1 - p_1$ is also ε_1 -close to $1 - p'_1$. Let $p'(x, q) := \Pr_{(\tilde{x}, \tilde{q}) \sim Q(D')}[(\tilde{x}, \tilde{q}) = (x, q)]$, then we also have $p(x, q)$ is ε_1 -close to $p'(x, q)$.

$$\begin{aligned} \frac{\Pr[A_{\text{out}}(D) = (x, q)]}{\Pr[A_{\text{out}}(D') = (x, q)]} &= \frac{p(x, q)}{p'(x, q)} \cdot \frac{1 - ((1 - p_1)(1 - \gamma))^T}{1 - ((1 - p'_1)(1 - \gamma))^T} \cdot \frac{1 - (1 - p'_1)(1 - \gamma)}{1 - (1 - p_1)(1 - \gamma)} \\ &\leq \exp(\varepsilon_1) \cdot \frac{1}{1 - (1 - \gamma)^T} \cdot \frac{p'_1(1 - \gamma) + \gamma}{p_1(1 - \gamma) + \gamma} \\ &\leq \exp(\varepsilon_1) \cdot \frac{1}{1 - \varepsilon_0/2} \cdot \exp(\varepsilon_1), \quad \text{if } T \geq \frac{1}{\gamma} \ln \frac{2}{\varepsilon_0} \\ &\leq \exp(2\varepsilon_1 + \varepsilon_0), \quad \text{if } \varepsilon_0 \leq 1. \end{aligned}$$

Next we consider the event of outputting \perp on dataset D .

$$\begin{aligned} \Pr[A_{\text{out}}(D) = \perp] &= \left(\sum_{j=1}^T \Pr[A_{\text{out}}(D) = \perp \wedge \text{stops after } j \text{ steps}] \right) + \Pr[\text{not stopping after } T \text{ steps}] \\ &= \left(\sum_{j=1}^T ((1 - p_1)(1 - \gamma))^{j-1} \cdot (1 - p_1)\gamma \right) + ((1 - p_1)(1 - \gamma))^T \\ &= (1 - p_1)\gamma \cdot \frac{1 - ((1 - p_1)(1 - \gamma))^T}{1 - (1 - p_1)(1 - \gamma)} + ((1 - p_1)(1 - \gamma))^T \\ &= \frac{(1 - p_1)\gamma - (1 - p_1)^{T+1}(1 - \gamma)^T\gamma + (1 - p_1)^T(1 - \gamma)^T - (1 - p_1)^{T+1}(1 - \gamma)^{T+1}}{1 - (1 - p_1)(1 - \gamma)} \\ &= \frac{(1 - p_1)\gamma + p_1(1 - p_1)^T(1 - \gamma)^T}{1 - (1 - p_1)(1 - \gamma)} \\ &= (1 - p_1)\gamma \cdot \frac{1 + \frac{p_1}{\gamma}(1 - p_1)^{T-1}(1 - \gamma)^T}{p_1(1 - \gamma) + \gamma}. \end{aligned}$$

Similarly, we have,

$$\begin{aligned}
\frac{\Pr[A_{\text{out}}(D) = \perp]}{\Pr[A_{\text{out}}(D') = \perp]} &= \frac{1 - p_1}{1 - p'_1} \cdot \frac{1 + \frac{p_1}{\gamma}(1 - p_1)^{T-1}(1 - \gamma)^T}{1 + \frac{p'_1}{\gamma}(1 - p'_1)^{T-1}(1 - \gamma)^T} \cdot \frac{p'_1(1 - \gamma) + \gamma}{p_1(1 - \gamma) + \gamma} \\
&\leq \exp(\varepsilon_1) \cdot \left(1 + p_1(1 - p_1)^{T-1} \cdot \frac{1}{\gamma}(1 - \gamma)^T\right) \cdot \exp(\varepsilon_1) \\
&\stackrel{(\dagger)}{\leq} \exp(\varepsilon_1) \cdot \left(1 + \frac{1}{e(T-1)\gamma} \cdot (1 - \gamma)^T\right) \cdot \exp(\varepsilon_1), \quad \text{by AM-GM inequality} \\
&\leq \exp(2\varepsilon_1) \cdot (1 + \varepsilon_0/2), \quad \text{if } T \geq \max\left\{\frac{1}{\gamma} \ln \frac{2}{\varepsilon_0}, 1 + \frac{1}{e\gamma}\right\} \\
&\leq \exp(2\varepsilon_1 + \varepsilon_0),
\end{aligned}$$

where (\dagger) follows from AM-GM inequality: recall that $T > 1$ is an integer, and $0 \leq p_1 \leq 1$, then

$$(T-1)p_1(1-p_1)^{T-1} \leq \left(\frac{(T-1)p_1 + (T-1)(1-p_1)}{T}\right)^T = \left(1 - \frac{1}{T}\right)^T \leq e^{-1}.$$

This concludes part (b). For part (c), it is worth noting that the privacy does not degrade as we increase T (the number of iterations).

We consider any event E on the output of Algorithm 1. Note that E will be a set of tuples (x, q) , and possibly contain \perp . Let $A_{\text{out}}(D)$ be the output of Algorithm 1 on dataset D , and $A_{\text{out}}(D')$ be the output on a neighboring dataset D' . If $\perp \in E$, then clearly $\Pr[A_{\text{out}}(D) \in E] = \Pr[A_{\text{out}}(D) \in E \setminus \{\perp\}] + \Pr[A_{\text{out}}(D) = \perp]$. In the following we will bound the two terms separately. For the first term, we consider any event F that does not contain \perp . Let

$$\begin{aligned}
p &:= \Pr_{(x,q) \sim Q(D)}[(x, q) \in F] & \text{and} & & p' &:= \Pr_{(i,m,q) \sim Q(D')}[(x, q) \in F], \\
p_1 &:= \Pr_{q \sim Q(D)}[q \geq \tau] & \text{and} & & p'_1 &:= \Pr_{q \sim Q(D')}[q \geq \tau].
\end{aligned}$$

If Q is $(\varepsilon_1, \delta_1)$ -DP, then we know that $p \leq e^{\varepsilon_1}p' + \delta_1$, and $p'_1 \leq e^{\varepsilon_1}p_1 + \delta_1$, or equivalently that $p_1 \geq \max\{0, p'_1 - \delta\} e^{-\varepsilon_1}$. Also notice that $p \leq p_1$ and $p' \leq p'_1$. Then, by calculations in part (a), we have the following upperbound:

$$\begin{aligned}
\Pr[A_{\text{out}}(D) \in F] &= p \cdot \frac{1 - ((1 - p_1)(1 - \gamma))^T}{1 - (1 - p_1)(1 - \gamma)} \\
&\leq \frac{p}{p_1(1 - \gamma) + \gamma} \\
&\leq \frac{e^{\varepsilon_1} \cdot p' + \delta_1}{\max\{0, p'_1 - \delta_1\} \cdot e^{-\varepsilon_1}(1 - \gamma) + \gamma} \\
&\leq \frac{e^{2\varepsilon_1} \cdot p' + e^{\varepsilon_1} \delta_1}{\max\{0, p'_1 - \delta_1\} \cdot (1 - \gamma) + \gamma}.
\end{aligned}$$

Furthermore we have the following lowerbound:

$$\begin{aligned}
\Pr[A_{\text{out}}(D') \in F] &= p' \frac{1 - ((1 - p'_1)(1 - \gamma))^T}{1 - (1 - p'_1)(1 - \gamma)} \\
&\geq p' \frac{1 - (1 - \gamma)^T}{p'_1(1 - \gamma) + \gamma} \\
&\geq p' \frac{1 - \varepsilon_0/2}{p'_1(1 - \gamma) + \gamma}, \quad \text{if } T \geq \frac{1}{\gamma} \ln \frac{2}{\varepsilon_0} \\
&\geq \frac{e^{-\varepsilon_0} \cdot p'}{p'_1(1 - \gamma) + \gamma}, \quad \text{if } \varepsilon_0 \leq 1
\end{aligned}$$

Then for an event $F = E \setminus \{\perp\}$ (that is, F does not contain \perp), we have

$$\begin{aligned}
&\Pr[A_{\text{out}}(D) \in F] - e^{2\varepsilon_1 + \varepsilon_0} \cdot \Pr[A_{\text{out}}(D') \in F] \\
&\leq e^{2\varepsilon_1} p' \cdot \left(\frac{1}{\max\{0, p'_1 - \delta_1\} \cdot (1 - \gamma) + \gamma} - \frac{1}{p'_1(1 - \gamma) + \gamma} \right) + \frac{e^{\varepsilon_1} \delta_1}{\max\{0, p'_1 - \delta_1\} (1 - \gamma) + \gamma} \\
&\leq \frac{e^{2\varepsilon_1} \delta_1 p' (1 - \gamma) + e^{\varepsilon_1} \delta_1 (p'_1(1 - \gamma) + \gamma)}{(\max\{0, p'_1 - \delta_1\} (1 - \gamma) + \gamma) (p'_1(1 - \gamma) + \gamma)} \\
&\leq \frac{e^{2\varepsilon_1} \delta_1 (2p'_1(1 - \gamma) + \gamma)}{(\max\{0, p'_1 - \delta_1\} (1 - \gamma) + \gamma) (p'_1(1 - \gamma) + \gamma)}, \quad \text{by } p' \leq p'_1 \\
&\leq \frac{2e^{2\varepsilon_1} \delta_1}{\gamma}.
\end{aligned}$$

Next, notice that we also have $1 - p_1 \leq e^{\varepsilon_1} (1 - p'_1) + \delta_1$, then for the output \perp we can upperbound

$$\begin{aligned}
\Pr[A_{\text{out}}(D) = \perp] &= (1 - p_1) \gamma \cdot \frac{1 + \frac{p_1}{\gamma} (1 - p_1)^{T-1} (1 - \gamma)^T}{p_1(1 - \gamma) + \gamma} \\
&\leq (1 - p_1) \gamma \cdot \frac{1 + \frac{1}{e^{(T-1)\gamma}} (1 - \gamma)^T}{p_1(1 - \gamma) + \gamma}, \quad \text{by AM-GM inequality} \\
&\leq (e^{\varepsilon_1} (1 - p'_1) + \delta_1) \gamma \cdot \frac{1 + \frac{1}{e^{(T-1)\gamma}} (1 - \gamma)^T}{e^{-\varepsilon_1} \max\{0, p'_1 - \delta_1\} (1 - \gamma) + \gamma}, \quad \text{by } (\varepsilon_1, \delta_1)\text{-DP} \\
&\leq (e^{\varepsilon_1} (1 - p'_1) + \delta_1) \gamma \cdot \frac{1 + \varepsilon_0/2}{e^{-\varepsilon_1} \max\{0, p'_1 - \delta_1\} (1 - \gamma) + \gamma}, \quad \text{by the choice of } T \\
&\leq \frac{e^{2\varepsilon_1 + \varepsilon_0} (1 - p'_1) \gamma + e^{\varepsilon_1 + \varepsilon_0} \delta_1 \gamma}{\max\{0, p'_1 - \delta_1\} (1 - \gamma) + \gamma}.
\end{aligned}$$

And we lowerbound

$$\begin{aligned}
\Pr[A_{\text{out}}(D') = \perp] &= (1 - p'_1) \gamma \cdot \frac{1 + \frac{p'_1}{\gamma} (1 - p'_1)^{T-1} (1 - \gamma)^T}{p'_1(1 - \gamma) + \gamma} \\
&\geq (1 - p'_1) \gamma \cdot \frac{1}{p'_1(1 - \gamma) + \gamma}
\end{aligned}$$

Therefore we have

$$\begin{aligned}
& \Pr[A_{\text{out}}(D) = \perp] - e^{2\varepsilon_1 + \varepsilon_0} \cdot \Pr[A_{\text{out}}(D') = \perp] \\
& \leq e^{2\varepsilon_1 + \varepsilon_0} (1 - p'_1) \gamma \left(\frac{1}{\max\{0, p'_1 - \delta_1\}(1 - \gamma) + \gamma} - \frac{1}{p'_1(1 - \gamma) + \gamma} \right) + \frac{e^{\varepsilon_1 + \varepsilon_0} \delta_1 \gamma}{\max\{0, p'_1 - \delta_1\}(1 - \gamma) + \gamma} \\
& = \frac{e^{2\varepsilon_1 + \varepsilon_0} \delta_1 \gamma \cdot (1 - p'_1)(1 - \gamma)}{(\max\{0, p'_1 - \delta_1\}(1 - \gamma) + \gamma)(p'_1(1 - \gamma) + \gamma)} + \frac{e^{\varepsilon_1 + \varepsilon_0} \delta_1 \gamma}{\max\{0, p'_1 - \delta_1\}(1 - \gamma) + \gamma} \\
& \leq \frac{e^{2\varepsilon_1 + \varepsilon_0} \delta_1 \gamma \cdot ((1 - p'_1)(1 - \gamma) + p'_1(1 - \gamma) + \gamma)}{(\max\{0, p'_1 - \delta_1\}(1 - \gamma) + \gamma)(p'_1(1 - \gamma) + \gamma)} \\
& \leq \frac{e^{2\varepsilon_1 + \varepsilon_0} \cdot \delta_1}{\gamma}.
\end{aligned}$$

Finally, for an event E that contains \perp , we let $F = E \setminus \{\perp\}$, and then

$$\begin{aligned}
& \Pr[A_{\text{out}}(D) \in E] - e^{2\varepsilon_1 + \varepsilon_0} \cdot \Pr[A_{\text{out}}(D') \in E] \\
& = \Pr[A_{\text{out}}(D) \in F] + \Pr[A_{\text{out}}(D) = \perp] - e^{2\varepsilon_1 + \varepsilon_0} \cdot (\Pr[A_{\text{out}}(D') \in F] + \Pr[A_{\text{out}}(D') = \perp]) \\
& \leq \frac{2e^{2\varepsilon_1} \delta_1}{\gamma} + \frac{e^{2\varepsilon_1 + \varepsilon_0} \cdot \delta_1}{\gamma} \\
& \leq 3e^{2\varepsilon_1 + \varepsilon_0} \cdot \frac{\delta_1}{\gamma}.
\end{aligned}$$

For part (d), notice that in each iteration, in order to not halt, q has to be below τ , and the γ -biased coin test did not pass. In other words, for each iteration, $\Pr[\text{halting in any iteration}] = 1 - (1 - p_1)(1 - \gamma) = p_1(1 - \gamma) + \gamma$. Therefore this can be stochastically dominated by a geometric distribution (which corresponds to setting $T = \infty$), with expected number of trials being at most $\frac{1}{p_1(1 - \gamma) + \gamma}$.

For part (e), by direct calculations,

$$\begin{aligned}
\Pr[\text{output } \perp] &= (1 - p_1) \gamma \cdot \frac{1 + \frac{p_1}{\gamma}(1 - p_1)^{T-1}(1 - \gamma)^T}{p_1(1 - \gamma) + \gamma} \\
&\leq (1 - p_1) \gamma \cdot \frac{1 + \frac{1}{e^{(T-1)\gamma}}(1 - \gamma)^T}{p_1}, \quad \text{by AM-GM inequality} \\
&\leq \frac{(1 - p_1)(1 + \varepsilon_0/2)}{p_1} \gamma.
\end{aligned}$$

□

A.2. Proof of Theorem 3.5. For convenience we restate Theorem 3.5.

Theorem A.2. Fix any $\varepsilon_0 \in (0, 1/2)$, $\gamma \in [0, 1]$, $\delta_2 > 0$ and let $T = \left\lceil \frac{1}{\gamma} \left(\ln \frac{2(1+\gamma)^2}{\varepsilon_0 \gamma^2} + \ln \ln \frac{2(1+\gamma)^2}{\varepsilon_0 \gamma^2} \right) \right\rceil$. Consider a variant of Algorithm 2 that outputs the highest scored candidate from S if j reaches T . If Q is ε_1 -DP, then the output of this algorithm is $(3\varepsilon_1 + 3\varepsilon_0)$ -DP.

Proof. Similar to the proof of Theorem 3.2: let $A_{\text{out}}(D)$ be the output of Algorithm 2 on D , then we have

$$\begin{aligned}
& \Pr[A_{\text{out}}(D) = (x, q)] \\
&= \sum_{j=1}^T \Pr[A_{\text{out}}(D) = (x, q) \wedge |S| = j] \\
&= \sum_{j=1}^T \Pr[|S| = j] \cdot \Pr[\max S \leq q, \text{ and } (x, q) \in S \mid |S| = j] \\
&= \sum_{j=1}^T (1 - \gamma)^{j-1} \gamma \cdot \Pr[\max S \leq q, \text{ and } (x, q) \in S \mid |S| = j] \\
&\quad + (1 - \gamma)^T \cdot \Pr[\max S \leq q, \text{ and } (x, q) \in S \mid |S| = T].
\end{aligned}$$

Then, observe that

$$\Pr[\max S \leq q \mid |S| = j] = (1 - p_0)^j,$$

and

$$\Pr[(x, q) \in S \mid \max S \leq q, \text{ and } |S| = j] = 1 - \left(1 - \frac{p}{1 - p_0}\right)^j = 1 - \left(\frac{1 - p_1}{1 - p_0}\right)^j.$$

Together we have

$$\begin{aligned}
(2) \quad \Pr[A_{\text{out}}(D) = (x, q)] &= \sum_{j=1}^T (1 - \gamma)^{j-1} \gamma \cdot ((1 - p_0)^j - (1 - p_1)^j) \\
&\quad + (1 - \gamma)^T ((1 - p_0)^T - (1 - p_1)^T).
\end{aligned}$$

We denote $a := (1 - \gamma)(1 - p_0)$, $b := (1 - \gamma)(1 - p_1)$, then

$$\begin{aligned}
& \Pr[A_{\text{out}}(D) = (x, q)] \\
&= \frac{\gamma(1 - p_0)(1 - a^T)}{1 - a} - \frac{\gamma(1 - p_1)(1 - b^T)}{1 - b} + a^T - b^T \\
&= \frac{\gamma p + ((1 - p_1)(1 - p_0)\gamma^2 - p_0 p_1 \gamma)(b^T - a^T) - \gamma(p_1 a^T - p_0 b^T)}{((1 - p_0)\gamma + p_0)((1 - p_1)\gamma + p_1)} + a^T - b^T.
\end{aligned}$$

Observe that $a - b = (1 - \gamma)p$, and we have

$$a^T - b^T = \sum_{i=0}^{T-1} a^{T-i} b^i - a^{T-i-1} b^{i+1} = (a - b) \sum_{i=0}^{T-1} a^{T-i-1} b^i \leq T(a - b)a^{T-1}.$$

Using the upper bound on a , this also implies that

$$a^T - b^T \leq Tp(1 - \gamma)^T.$$

Furthermore,

$$\begin{aligned}
|p_1 a^T - p_0 b^T| &\leq |p_1 a^T - p_0 a^T| + |p_0 a^T - p_0 b^T| \\
&\leq p a^T + Tp(1 - \gamma)^T \cdot p_0(1 - p_0)^{T-1} \\
&\leq p(1 - \gamma)^T + \frac{T}{e(T-1)} p(1 - \gamma)^T, \text{ by AM-GM inequality} \\
&\leq 2p(1 - \gamma)^T.
\end{aligned}$$

And we also have

$$\begin{aligned} ((1-p_0)\gamma + p_0)((1-p_1)\gamma + p_1) &\leq (1+\gamma)^2 \\ |(1-p_1)(1-p_0)\gamma^2 - p_0p_1\gamma| &\leq \gamma(1+\gamma). \end{aligned}$$

Now, if $T \geq \frac{1}{\gamma} \left(\ln \frac{2(1+\gamma)^2}{\varepsilon_0\gamma^2} + \ln \ln \frac{2(1+\gamma)^2}{\varepsilon_0\gamma^2} \right)$, then we have $T(1-\gamma)^T \leq \frac{\varepsilon_0\gamma}{(1+\gamma)^2}$. Therefore we can upperbound

$$\begin{aligned} &\Pr[A_{\text{out}}(D) = (x, q)] \\ &= \frac{\gamma(1-p_0)(1-a^T)}{1-a} - \frac{\gamma(1-p_1)(1-b^T)}{1-b} + a^T - b^T \\ &\leq \frac{\gamma(1-p_0)}{1-a} - \frac{\gamma(1-p_1)}{1-b} + \frac{\varepsilon_0\gamma p}{(1+\gamma)^2} \\ &\leq \frac{\gamma p(1+\varepsilon_0)}{((1-p_0)\gamma + p_0)((1-p_1)\gamma + p_1)}. \end{aligned}$$

The first inequality above is a consequence of upper bounding the sum of the first T terms in eq. (2) by the sum to infinity. Then we lowerbound

$$\begin{aligned} &\Pr[A_{\text{out}}(D) = (x, q)] \\ &= \frac{\gamma p + ((1-p_1)(1-p_0)\gamma^2 - p_0p_1\gamma)(b^T - a^T) - \gamma(p_1a^T - p_0b^T)}{((1-p_0)\gamma + p_0)((1-p_1)\gamma + p_1)} + a^T - b^T \\ &\geq \frac{\gamma p - \frac{\varepsilon_0\gamma^2}{1+\gamma}p - \frac{2\gamma^2\varepsilon_0}{T(1+\gamma)^2}p}{((1-p_0)\gamma + p_0)((1-p_1)\gamma + p_1)} \\ &\geq \frac{\gamma p - \frac{\gamma+3\gamma^2}{(1+\gamma)^2}\varepsilon_0\gamma p}{((1-p_0)\gamma + p_0)((1-p_1)\gamma + p_1)} \\ &\geq \frac{\gamma p(1-\varepsilon_0)}{((1-p_0)\gamma + p_0)((1-p_1)\gamma + p_1)}. \end{aligned}$$

Finally,

$$\frac{\Pr[A_{\text{out}}(D) = (x, q)]}{\Pr[A_{\text{out}}(D') = (x, q)]} \leq \frac{p}{p'} \cdot \frac{p'_0(1-\gamma) + \gamma}{p_0(1-\gamma) + \gamma} \cdot \frac{p'_1(1-\gamma) + \gamma}{p_1(1-\gamma) + \gamma} \cdot \frac{1+\varepsilon_0}{1-\varepsilon_0} \leq \exp(3\varepsilon_1 + 3\varepsilon_0).$$

□

A.3. Proof of Lemma 4.3. We re-state Lemma 4.3 below for convenience.

Lemma A.3. *Let $\{X_1, \dots, X_n\}$ and $\{Y_1, \dots, Y_n\}$ be two sequences of independent $\{0, 1\}$ random variables, and let $X = \sum_{i=1}^n X_i$, $Y = \sum_{i=1}^n Y_i$. For any fixed $\varepsilon_1 \in (0, 1)$, $\varepsilon_0 \in (0, 1)$, $\delta_0 \in (0, 1)$, let $C = 2(e^{\varepsilon_0+\varepsilon_1} + 1 + e^{\varepsilon_0/2}) < 21$.*

If $\mathbb{E}X \leq e^{\varepsilon_1}\mathbb{E}Y$, then under the trivial (independent) coupling between X and Y ,

$$\Pr\left[X \geq e^{\varepsilon_1+\varepsilon_0} \cdot Y + \frac{C}{\varepsilon_0} \cdot \ln \frac{2}{\delta_0}\right] \leq \delta_0.$$

Equivalently, if we let $\Delta := \frac{C \ln \frac{2}{\delta_0}}{\varepsilon_0(e^{\varepsilon_0+\varepsilon_1}-1)} = O\left(\frac{1}{\varepsilon_0^2} \ln \frac{1}{\delta_0}\right)$, then

$$\Pr[X + \Delta \geq e^{\varepsilon_1+\varepsilon_0} \cdot (Y + \Delta)] \leq \delta_0.$$

Before proving this lemma, it will be useful to show the following concentration bounds.

Lemma A.4. Let X be a sum of independent $\{0, 1\}$ random variables: $X = \sum_{i=1}^n X_i$ as defined in Lemma 4.3, then $\forall \varepsilon \in (0, 1), \delta \in (0, 1)$,

$$(3) \quad \Pr \left[X \geq e^\varepsilon \mathbb{E}X + \frac{e^\varepsilon + 1}{\varepsilon} \ln \frac{1}{\delta} \right] \leq \delta.$$

Proof. Let $\Delta_1 = \frac{e^\varepsilon + 1}{(e^\varepsilon - 1)^2} \ln \frac{1}{\delta}$, then we apply the standard Chernoff bound to $X + \Delta_1$:

$$\Pr[X + \Delta_1 \geq e^\varepsilon \cdot \mathbb{E}(X + \Delta_1)] \leq \exp \left(\frac{-(e^\varepsilon - 1)^2 \mathbb{E}(X + \Delta_1)}{e^\varepsilon + 1} \right) \leq \exp \left(\frac{-(e^\varepsilon - 1)^2 \Delta_1}{e^\varepsilon + 1} \right) = \delta.$$

By re-arranging, we get that

$$\Pr \left[X \geq e^\varepsilon \mathbb{E}X + \frac{e^\varepsilon + 1}{\varepsilon} \ln \frac{1}{\delta} \right] \leq \Pr \left[X \geq e^\varepsilon \mathbb{E}X + \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \ln \frac{1}{\delta} \right] = \Pr[X \geq e^\varepsilon \mathbb{E}X + (e^\varepsilon - 1)\Delta_1] \leq \delta.$$

□

Lemma A.5. Let Y be a sum of independent $\{0, 1\}$ random variables: $Y = \sum_{i=1}^n X_i$ as defined in Lemma 4.3, then $\forall \varepsilon \in (0, 1), \delta \in (0, 1)$,

$$(4) \quad \Pr \left[Y \leq e^{-\varepsilon} \mathbb{E}Y - \frac{\ln \frac{1}{\delta}}{\varepsilon} \right] \leq \delta.$$

Proof. Note that by a direct application of Chernoff bound, it holds that $\forall \delta \in (0, 1)$,

$$\Pr \left[Y \leq \left(1 - \sqrt{\frac{2 \ln \frac{1}{\delta}}{\mathbb{E}Y}} \right) \cdot \mathbb{E}Y \right] \leq \exp \left(- \left(\frac{2 \ln \frac{1}{\delta}}{\mathbb{E}Y} \right) \cdot \frac{\mathbb{E}Y}{2} \right) = \delta.$$

Then, by AM-GM inequality: $\forall \varepsilon$, we have $\frac{\varepsilon \mathbb{E}Y}{2} + \frac{\ln \frac{1}{\delta}}{\varepsilon} \geq \sqrt{2 \mathbb{E}Y \cdot \ln \frac{1}{\delta}}$. Moreover, by standard estimates: for $\varepsilon \in (0, 1)$, we have $1 - \varepsilon \leq e^{-\varepsilon} \leq 1 - \varepsilon/2$. Therefore, $\forall \varepsilon \in (0, 1), \delta \in (0, 1)$,

$$\begin{aligned} \Pr \left[Y \leq e^{-\varepsilon} \mathbb{E}Y - \frac{\ln \frac{1}{\delta}}{\varepsilon} \right] &\leq \Pr \left[Y \leq \mathbb{E}Y - \left(\frac{\varepsilon \mathbb{E}Y}{2} + \frac{\ln \frac{1}{\delta}}{\varepsilon} \right) \right] \\ &\leq \Pr \left[Y \leq \mathbb{E}Y - \sqrt{2 \mathbb{E}Y \cdot \ln \frac{1}{\delta}} \right], \quad \text{by AM-GM inequality} \\ &= \Pr \left[Y \leq \left(1 - \sqrt{\frac{2 \ln \frac{1}{\delta}}{\mathbb{E}Y}} \right) \cdot \mathbb{E}Y \right] \\ &\leq \delta. \end{aligned}$$

□

Finally, we are ready to prove Lemma 4.3.

Proof of Lemma 4.3. For any given ε_0, δ_0 , we set $\varepsilon = \varepsilon_0/2$, and $\delta = \delta_0/2$. Then we consider the following events, G_X and G_Y on the probability space of X and Y respectively:

$$\begin{aligned} G_X &:= \left\{ X : X < e^\varepsilon \mathbb{E}X + \frac{e^\varepsilon + 1}{\varepsilon} \ln \frac{1}{\delta} \right\}; \\ G_Y &:= \left\{ Y : Y > e^{-\varepsilon} \mathbb{E}Y - \frac{\ln \frac{1}{\delta}}{\varepsilon} \right\}. \end{aligned}$$

As discussed in Lemmas A.4 and A.5, we have

$$\Pr \left[\overline{G_X} \cup \overline{G_Y} \right] \leq \Pr \left[\overline{G_X} \right] + \Pr \left[\overline{G_Y} \right] \leq \delta + \delta = \delta_0.$$

On the other hand, conditional on G_X and G_Y , we must have

$$\begin{aligned} X &< e^\varepsilon \mathbb{E}X + \frac{e^\varepsilon + 1}{\varepsilon} \ln \frac{1}{\delta}, \quad \text{by } G_X \\ &\leq e^{\varepsilon + \varepsilon_1} \mathbb{E}Y + \frac{e^\varepsilon + 1}{\varepsilon} \ln \frac{1}{\delta} \\ &\leq e^{2\varepsilon + \varepsilon_1} \left(Y + \frac{\ln \frac{1}{\delta}}{\varepsilon} \right) + \frac{e^\varepsilon + 1}{\varepsilon} \ln \frac{1}{\delta}, \quad \text{by } G_Y \\ &= e^{\varepsilon_0 + \varepsilon_1} \cdot Y + \frac{2(e^{\varepsilon_0 + \varepsilon_1} + 1 + e^{\varepsilon_0/2})}{\varepsilon_0} \ln \frac{2}{\delta_0} \end{aligned}$$

Therefore, let $C = 2(e^{\varepsilon_0 + \varepsilon_1} + 1 + e^{\varepsilon_0/2})$, then

$$\Pr \left[X \geq e^{\varepsilon_0 + \varepsilon_1} \cdot Y + \frac{C \cdot \ln \frac{2}{\delta_0}}{\varepsilon_0} \right] \leq \Pr \left[\overline{G_X} \cup \overline{G_Y} \right] \leq \delta_0.$$

□

APPENDIX B. NAIVE ALGORITHMS: TIGHT EXAMPLES AND ANALYSIS

B.1. Outputting the best candidate. In this subsection we consider a naive algorithm where, one simply chooses the best candidate (with the highest score, e.g., in the hyperparameter selection setting, among the trained models one outputs the best performing model and its corresponding hyperparameter).

What is the best ε -DP bound, or (ε, δ) -DP bound that we can hope for? Basic composition theorem says that if there are K candidates, and each candidate is ε -DP, then, outputting the best of the K candidates is $(K\varepsilon)$ -DP. This is actually tight, thanks to the following example.

$$\begin{aligned} \text{let } m \sim M_i(d_1), \text{ and } q_i(m) &= \begin{cases} 0.9, & \text{if } i = 0 \\ 0.8, & \text{if } i \neq 0 \text{ and with probability } \frac{1}{2} \\ 0.95, & \text{if } i \neq 0 \text{ and with probability } \frac{1}{2} \end{cases} \\ \text{let } m \sim M_i(d'_1), \text{ and } q_i(m) &= \begin{cases} 0.9, & \text{if } i = 0 \\ 0.8, & \text{if } i \neq 0 \text{ and with probability } \frac{e^\varepsilon}{2} \\ 0.95, & \text{if } i \neq 0 \text{ and with probability } \frac{1-e^\varepsilon}{2} \end{cases} \end{aligned}$$

Here the probability are with respect to the randomness in the ε -DP candidate M . Then for neighboring datasets d_1 and d'_1 , we get K samples of the candidates (e.g. for each of the K candidates, we draw a sample), and then we compare the event of choosing $i = 0$ as the best hyperparameter. It is easy to see that $\Pr[i_*(d_1) = 0] = 2^{-K}$ and $\Pr[i_*(d'_1) = 0] = \exp(K\varepsilon) \cdot 2^{-K}$, therefore,

$$\ln \left| \frac{\Pr[i_*(d'_1) = 0]}{\Pr[i_*(d_1) = 0]} \right| = K\varepsilon.$$

What about (ε, δ) -DP bound? We show that outputting the maximum cannot do better than $(\Theta(\ln \frac{1}{\delta})\varepsilon, \delta)$ -DP. Fix an integer K , and $\delta \in (0, 1)$.

$$\begin{aligned} \text{let } m \sim M_i(d_1), \text{ and } q_i(m) &= \begin{cases} 0.9, & \text{if } i = 0 \\ 0.8, & \text{if } i \neq 0 \text{ and with probability } 1 - \frac{\ln 1/\delta}{K} \\ 0.95, & \text{if } i \neq 0 \text{ and with probability } \frac{\ln 1/\delta}{K} \end{cases} \\ \text{let } m \sim M_i(d'_1), \text{ and } q_i(m) &= \begin{cases} 0.9, & \text{if } i = 0 \\ 0.8, & \text{if } i \neq 0 \text{ and with probability } 1 - \frac{e^{-\varepsilon} \ln 1/\delta}{K} \\ 0.95, & \text{if } i \neq 0 \text{ and with probability } \frac{e^{-\varepsilon} \ln 1/\delta}{K} \end{cases} \end{aligned}$$

Again for neighboring datasets d_1 and d'_1 , we get K samples of the candidates (e.g. for each of the K candidates, we draw a sample), and then we compare the event of choosing $i = 0$ as the best hyperparameter. It is easy to see that $\Pr[i_*(d_1) = 0] \approx \delta$ and $\Pr[i_*(d'_1) = 0] \approx \delta^{1-\varepsilon}$, therefore,

$$\ln \left| \frac{\Pr[i_*(d'_1) = 0] - \delta}{\Pr[i_*(d_1) = 0]} \right| \approx \ln(\delta^{-\varepsilon} - 1) \approx (\ln 1/\delta) \varepsilon.$$

B.2. Thresholding with decreasing thresholds. In this subsection we consider a natural variant of Algorithm 1: in each iteration, instead of halting (and output \perp) with probability γ , what if we decrease the threshold? In particular, we will try a lower threshold with probability at least γ in each step. Is this good enough, so that we can avoid paying the privacy cost for the different thresholds that we tried along the way? See Algorithm 6 for formal description. For simplicity, we consider the special case where we do not stop the algorithm after some finite number of T steps. The algorithm could run forever *in the worst case*. Note that in Algorithm 1, running the algorithm longer only helps in privacy (recall that in Theorem 3.1, the larger T is, the smaller ε_0 we can choose).

Algorithm 6 Thresholding with decreasing thresholds.

Input: a budget $\gamma \leq 1$, an integer R for how many thresholds to try, and the sampling access to $Q(D)$.

Let $\tau = 1$;

While $\tau \geq 0$:

- draw $(x, q) \sim Q(D)$;
 - if $q \geq \tau$ then output (m, i) and halt;
 - flip a γ -biased coin: with probability γ , set $\tau \leftarrow \tau - \frac{1}{R}$.
-

Note that as soon as $\tau = 0$, the algorithm will output whichever samples of candidate that it gets, as $q \geq \tau$ is trivially true.

Here is an example which shows that trying many thresholds are not free for privacy: if we plan to try R thresholds, then we do have to pay a factor of R in the privacy cost.

Consider $Q(D) = \text{Bernoulli}(p)$, that is, $q \sim Q(D)$ will be 1 with probability p , and 0 otherwise. Therefore, in order for the algorithm to output a candidate with score 0, the threshold has to be decreased R times, until $\tau = 0$. And only then Algorithm 6 will output a score 0 candidate with probability $1 - p$. Let us compute the probability of this event. Let $a = (1 - p)(1 - \gamma)$, $b = (1 - p)\gamma$.

$$\begin{aligned}
\Pr[\text{output 1 before the } R\text{-th decrements on } D] &= p \sum_{j=0}^{\infty} \sum_{i=0}^{R-1} \binom{j}{i} a^{j-i} b^i \\
&= p \sum_{i=0}^{R-1} b^i \sum_{j=i}^{\infty} \binom{j}{i} a^{j-i} \\
&= p \sum_{i=0}^{R-1} b^i \frac{1}{(1-a)^{i+1}} \\
&= \frac{p}{1-a} \frac{1 - \left(\frac{b}{1-a}\right)^R}{1 - \frac{b}{1-a}} \\
&= 1 - \left(\frac{b}{1-a}\right)^R.
\end{aligned}$$

Therefore

$$\begin{aligned}
&\Pr[\text{output 0 on } D] \\
&= (1-p) \cdot (1 - \Pr[\text{output 1 before the } R\text{-th decrements on } D]) \\
&= (1-p) \left(\frac{b}{1-a}\right)^R = (1-p) \gamma^R \cdot \left(\frac{1-p}{p(1-\gamma) + \gamma}\right)^R.
\end{aligned}$$

Notice that $\frac{1-p}{p(1-\gamma) + \gamma}$ is monotone in p , if $(1-p)$ changes to $e^\varepsilon(1-p)$, then the e^ε factor will be amplified R times.

B.3. Outputting the p -th percentile. Without loss of generality, we consider $p = \frac{1}{2}$, that is, we output the median candidate. Also without loss of generality, let us say there are only two models, m_1 and m_2 , and $q(m_1) = 0$, $q(m_2) = 1$. Consider the following two distributions of $M_i(D)$ and $M_i(D')$.

$$\begin{aligned}
\forall i, M_i(D) &= \begin{cases} m_1, & \text{with probability } \frac{1-\varepsilon}{2} \\ m_2, & \text{with probability } \frac{1+\varepsilon}{2} \end{cases}, \\
\forall i, M_i(D') &= \begin{cases} m_1, & \text{with probability } \frac{1+\varepsilon}{2} \\ m_2, & \text{with probability } \frac{1-\varepsilon}{2} \end{cases}.
\end{aligned}$$

Clearly, the two distributions are $O(\varepsilon)$ -close, yet in one distribution, the median is m_2 , while in the other the median is m_1 . Therefore, the median of the distribution is not private. This is also the case if one takes the median of N samples, assuming N large enough (where we have concentration with high probability). This also applies if one pick an index k from $\{1, 2, \dots, \lceil N/2 \rceil\}$ uniformly at random, and then output the k -th highest.

APPENDIX C. IMPROVED ANALYSIS OF THE PRIVATE AMPLIFICATION ALGORITHM IN [19]

Let $\{Q_i(D)\}_{i=1}^N$ be a sequence of independent distributions, let $\{q_i(D)\}_{i=1}^N$ be the random variables where $q_i \sim Q_i$.

Suppose that for some ε_1 , every q_i is ε_1 -DP. For a given $\tau \in \mathbb{R}$, let $\tilde{q}_i = \min\{\tau, q_i\}$. Denote $\mathcal{E}_\varepsilon(A_1, \dots, A_n)$ to be the exponential mechanism (with parameter ε) on the sequence $\{A_i\}$, which

is a random variable. For any given $\gamma > 0$, $n = N + 1 + \frac{1}{\gamma}$, and $A_i = \begin{cases} \tilde{q}_i(D), & \text{if } i \leq N \\ \tau, & \text{otherwise} \end{cases}$. Then, we consider the distribution of $\mathcal{E}_{\varepsilon_2}(A_1, \dots, A_n)$. Basically, we will add $1 + \frac{1}{\gamma}$ dummy classes with a score τ , scale back everything else and then apply exponential mechanism.

Theorem C.1. Fix $\varepsilon_2, \tau, \delta, \gamma \in (0, \frac{1}{4})$. Suppose that for every i , q_i is ε_1 -differentially private. Let $p = \frac{1}{N} \sum_{i=1}^N \mathbb{E} \exp(\varepsilon_2(\tilde{q}_i - \tau))$. Then the following holds.

- *Utility:* The mechanism outputs a dummy class with probability $\frac{\gamma+1}{Np\gamma+1}$, and

$$\Pr \left[\mathcal{E}_{\varepsilon_2}(A_1, \dots, A_n) \geq \tau - \frac{1}{\varepsilon_2} \ln \left(\frac{1}{\delta p} \right) \right] \geq 1 - \delta,$$

where the randomness is over both the internal randomness of exponential mechanism and the randomness of $\{q_i\}$.

- *Privacy:* $\mathcal{E}_{\varepsilon_2}(A_1, \dots, A_n)$ is $(2\varepsilon_1 + 8\gamma)$ -DP.

It is worth noting that the privacy on the training set does not depend on ε_2 . In other words, one can even set $\varepsilon_2 \rightarrow \infty$, which corresponds to sampling uniformly from the classes with a score exceeding some threshold and with one extra dummy class. The theorem says that doing so does not compromise the privacy of the training set at all.

Before we prove the theorem, we introduce a useful lemma similar to that of [25, Lemma C.1]. The key changes will be from a Binomial distribution to one that takes value from $[0, 1]$.

Lemma C.2. Let $\{X_i\}_{i=1}^N$ be a sequence of independent random variables over $[0, 1]$, then

$$\frac{1}{1 + \sum_{i=1}^N \mathbb{E} X_i} \leq \mathbb{E} \left[\frac{1}{1 + \sum_{i=1}^N X_i} \right] \leq \frac{1}{\sum_{i=1}^N \mathbb{E} X_i}.$$

Proof. The first inequality follows by Jensen's inequality, since the function $f(X) = \frac{1}{1+X}$ is convex for $X > -1$.

Next we use a formula for negative moments [6]. Note that for every $u, x > 0$,

$$\frac{u^{1+x}}{1+x} = \int_0^u t^x dt.$$

Setting $u = 1$ and taking expectations over $x = \sum X_i$,

$$\mathbb{E} \left[\frac{1}{1 + \sum X_i} \right] = \int_0^1 \mathbb{E} [t^{\sum_{i=1}^N X_i}] dt = \int_0^1 \prod_{i=1}^N \mathbb{E} [t^{X_i}] dt.$$

Next we show that $t^x \leq (t-1)x + 1$ for $x \in [0, 1]$.

For any given t , consider the following two points: $t^x = \begin{cases} 1, & \text{if } x = 0 \\ t, & \text{if } x = 1 \end{cases}$. Thus $(t-1)x + 1$ is the

line joining these two points. Since $t^x = e^{x \ln t}$ is convex as long as $t > 0$, and t^x meets $(t-1)x + 1$ at the two points, we get that $t^x \leq (t-1)x + 1$ for $x \in [0, 1]$.

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{1 + \sum X_i}\right] &= \int_0^1 \prod_{i=1}^N \mathbb{E}[t_i^X] \, dt \\
&\leq \int_0^1 \prod_{i=1}^N (1 + (t-1)\mathbb{E}X_i) \, dt \\
&\leq \int_0^1 \exp\left((t-1) \sum_{i=1}^N \mathbb{E}X_i\right) \, dt \\
&= \frac{1 - \exp\left(-\sum_{i=1}^N \mathbb{E}X_i\right)}{\sum_{i=1}^N \mathbb{E}X_i}.
\end{aligned}$$

This concludes the proof. \square

Now we are ready to prove Theorem C.1.

Proof of Theorem C.1. The probability of outputting a dummy is

$$\mathbb{E}\left[\frac{(1 + \frac{1}{\gamma}) \exp(\varepsilon_2 \tau)}{(1 + \frac{1}{\gamma}) \exp(\varepsilon_2 \tau) + \sum_{i=1}^N \exp(\varepsilon_2 \tilde{q}_i)}\right] \leq \frac{1 + \frac{1}{\gamma}}{Np + \frac{1}{\gamma}},$$

where the inequality follows from the definition of p and Lemma C.2. Then,

$$\begin{aligned}
&\Pr\left[\mathcal{E}_{\varepsilon_2}(A_1, \dots, A_n) < \tau - \frac{1}{\varepsilon_2} \ln \frac{1}{\delta p}\right] \\
&\leq \sum_{i \in [N]} \Pr\left[\tilde{q}_i < \tau - \frac{1}{\varepsilon_2} \ln \frac{1}{\delta p}\right] \exp(\varepsilon_2 \tilde{q}_i) \cdot \mathbb{E}\left[\frac{1}{(1 + \frac{1}{\gamma}) \exp(\varepsilon_2 \tau) + \sum_{j=1}^N \exp(\varepsilon_2 \tilde{q}_j)} \middle| \tilde{q}_i\right] \\
&< Np\delta \cdot \mathbb{E}\left[\frac{1}{\frac{1}{\gamma} + \sum_{j=1}^N \exp(\varepsilon_2(\tilde{q}_j - \tau))}\right] \\
&\leq Np\delta \cdot \frac{1}{\frac{1}{\gamma} - 1 + \sum_{i=1}^N \mathbb{E} \exp(\varepsilon_2(\tilde{q}_i - \tau))} \quad \text{by Lemma C.2} \\
&\leq \delta \quad \text{by definition of } p.
\end{aligned}$$

For the privacy part, consider any two neighboring datasets D_1 and D_2 . Let $A_i = \tilde{q}_i(D_1)$, and $B_i = \tilde{q}_i(D_2)$ for $i \leq N$, and $A_i = B_i = \tau$ for $i = N+1, \dots, N+1 + \frac{1}{\gamma}$. Let $M_1 := \mathcal{E}_{\varepsilon_2}(A_1, \dots, A_n)$, $M_2 := \mathcal{E}_{\varepsilon_2}(B_1, \dots, B_n)$ be the random variables of the two outcomes.

For any outcome S ,

$$\begin{aligned}
\Pr[M_1 = S] &= \sum_{i=1}^N \Pr[A_i = S] \exp(\varepsilon_2 S) \cdot \mathbb{E}\left[\frac{1}{\exp(\varepsilon_2 \tau) + \sum_{j=1}^N \exp(\varepsilon_2 A_j)} \middle| A_i = S\right] \\
&= \sum_{i=1}^N \Pr[A_i = S] \exp(\varepsilon_2(S - \tau)) \cdot \mathbb{E}\left[\frac{1}{1 + \frac{1}{\gamma} + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))}\right].
\end{aligned}$$

Note that by Lemma C.2,

$$\begin{aligned}
\mathbb{E} \left[\frac{1}{1 + \frac{1}{\gamma} + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \right] &\leq \mathbb{E} \left[\frac{1}{1 + \frac{1}{\gamma} + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \right] \\
&\leq \frac{1}{\frac{1}{\gamma} + \mathbb{E} \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \\
&\leq \frac{1}{\frac{1}{\gamma} + \mathbb{E} \sum_{j=1}^N \exp(\varepsilon_2(A_j - \tau)) - 1} \\
&\leq \frac{1}{Np + \frac{1}{\gamma} - 1}.
\end{aligned}$$

On the other hand, by Jensen's inequality,

$$\begin{aligned}
&\mathbb{E} \left[\frac{1}{1 + \frac{1}{\gamma} + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \right] \\
&\geq \frac{1}{1 + \frac{1}{\gamma} + \exp(\varepsilon_2(S - \tau)) + \mathbb{E} \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \\
&\geq \frac{1}{2 + \frac{1}{\gamma} + \mathbb{E} \sum_{j=1}^N \exp(\varepsilon_2(A_j - \tau))} \\
&\geq \frac{1}{2 + \frac{1}{\gamma} + Np}.
\end{aligned}$$

By symmetry we have the same bound for M_2 and B_j . Therefore,

$$\begin{aligned}
\frac{\Pr[M_1 = S]}{\Pr[M_2 = S]} &\leq \max_{i \in [N]} \left\{ \frac{\Pr[A_i = S] \mathbb{E} \left[\frac{1}{1 + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \right]}{\Pr[B_i = S] \mathbb{E} \left[\frac{1}{1 + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(B_j - \tau))} \right]} \right\} \\
&\leq \exp(\varepsilon_1) \frac{Npe^{\varepsilon_1} + \frac{1}{\gamma} + 2}{Np + \frac{1}{\gamma} - 1} \\
&\leq \exp(2\varepsilon_1 + 8\gamma).
\end{aligned}$$

Similarly

$$\begin{aligned}
\frac{\Pr[M_1 = S]}{\Pr[M_2 = S]} &\geq \min_{i \in [N]} \left\{ \frac{\Pr[A_i = S] \mathbb{E} \left[\frac{1}{1 + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(A_j - \tau))} \right]}{\Pr[B_i = S] \mathbb{E} \left[\frac{1}{1 + \exp(\varepsilon_2(S - \tau)) + \sum_{j=1, j \neq i}^N \exp(\varepsilon_2(B_j - \tau))} \right]} \right\} \\
&\geq \exp(-\varepsilon_1) \frac{Npe^{-\varepsilon_1} + \frac{1}{\gamma} - 1}{Np + \frac{1}{\gamma} + 2} \\
&\geq \exp(-2\varepsilon_1 - 8\gamma).
\end{aligned}$$

This concludes the proof.

APPENDIX D. LOWER BOUNDS

In this section, we show that our algorithms loss in parameters are close to optimal. First note that since are competing against the best of K mechanisms, at least K oracle calls are needed, and our algorithm makes only $\tilde{O}(K)$ oracle calls.

When each of the input mechanisms \mathcal{M}_i is ε -DP, our final algorithm has privacy guarantee $2\varepsilon + \varepsilon'$ where ε' can be made arbitrarily small. Recall that in this factor of two loss occurs already in the case when \mathcal{M}_i is $q_i(\cdot) + \text{Lap}(S/\varepsilon)$ for some score functions with sensitivity S : in this case the NoisyMax mechanism has 2ε -DP, and the exponential mechanism with similar utility has the same factor of two loss. We next argue that this factor of two loss is necessary under weak utility assumptions. We start with a definition.

Definition D.1. Suppose that \mathcal{M} is an algorithm that takes as input a set of ε -DP mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_K$ and outputs an index i . We say that i^* is γ -dominant in $\mathcal{M}_1, \dots, \mathcal{M}_K$ on D if $\Pr_{m_i \sim \mathcal{M}_i(D)}[\arg \max_i m_i = i^*] \geq 1 - \gamma$. We say that \mathcal{M} is γ -weakly useful if $\Pr[\mathcal{M}(D) = i^*] \geq \gamma$ whenever i^* is γ -dominant in $\mathcal{M}_1, \dots, \mathcal{M}_K$ on D .

The next theorem says that a fairly mild weak usefulness condition already implies that this factor of 2 loss is unavoidable.

Theorem D.2. Suppose that \mathcal{M} is an algorithm that takes as input a set of ε -DP mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_K$, and outputs an index i . If \mathcal{M} is γ -weakly useful for $\gamma = K^{-\alpha}$ for a small enough $\alpha > 0$, then \mathcal{M} cannot be $\hat{\varepsilon}$ -DP for any $\hat{\varepsilon} < (2 - 6\alpha)\varepsilon$.

Proof. The proof is a simple packing argument. Our mechanisms \mathcal{M}_i all have range $\{0, 1\}$ and output 1 with probability $p_i(D)$ on dataset D . We define a set of $K + 1$ datasets D_0, D_1, \dots, D_K such that:

$$p_i(D_0) = \frac{1}{2K^{0.5}}$$

$$p_i(D_j) = \begin{cases} 1 - \frac{1}{2K^\alpha} & \text{if } i = j, \\ \frac{1}{2K^{1+\alpha}} & \text{otherwise.} \end{cases}$$

It is easy to check that if D_0 and D_i are distance $\Delta = \lceil \frac{(0.5+\alpha)\ln K}{\varepsilon} \rceil$, then the \mathcal{M}_i 's can be extended to satisfy ε -DP. Moreover, any $\frac{1}{K^\alpha}$ -weakly useful algorithm on dataset D_i should output i with probability at least $\frac{1}{K^\alpha}$. Suppose that \mathcal{M} is $\hat{\varepsilon}$ -DP. Then,

$$\Pr[\mathcal{M}(D_0) = i] \geq \exp(-\Delta\hat{\varepsilon}) \cdot \frac{1}{K^\alpha}.$$

Since $\sum_i \Pr[\mathcal{M}(D_0) = i] \leq 1$, it follows that for some i , this probability $\Pr[\mathcal{M}(D_0) = i] \leq \frac{1}{K}$. It follows that

$$-\ln K \geq -\Delta\hat{\varepsilon} - \alpha \ln K$$

$$\Leftrightarrow \hat{\varepsilon} \geq \frac{(1 - \alpha) \ln K}{\lceil ((0.5 + \alpha) \ln K) / \varepsilon \rceil}.$$

For large enough K , this implies that $\hat{\varepsilon} \geq 2(1 - 3\alpha)\varepsilon$. □

APPENDIX E. USEFUL PROPERTIES OF DIFFERENTIAL PRIVACY

In this section, we prove some folklore properties of the distance implicit in the definition of differential privacy that are useful. We start with a definition of closeness.

Definition E.1. For distributions P and Q , we say that P is (ε, δ) -far from Q , if for all events S ,

$$\Pr_{x \sim P}[x \in S] \leq \exp(\varepsilon) \Pr_{x \sim Q}[x \in S] + \delta$$

We say that $P \equiv_{\varepsilon, \delta} Q$ if P is (ε, δ) -far from Q and Q is (ε, δ) -far from P .

Lemma E.2. Suppose that $P \equiv_{\varepsilon, \delta} Q$ for $\delta < \frac{1}{10}$. Then for any $\varepsilon' > \varepsilon$, there is an event B such that (a) $\Pr_{x \sim P}[x \in B] \leq \delta/(1 - \exp(\varepsilon - \varepsilon'))$, and (b) $P \mid B^c \equiv_{\varepsilon', 0} Q \mid B^c$. In particular, setting $\varepsilon' = \varepsilon + \sqrt{2\delta}$, we get $\Pr_P[B] \leq \sqrt{\delta}$.

Proof. Without loss of generality², the distributions have a density function. Let $B = \{x : \frac{\Pr_P[x]}{\Pr_Q[x]} \geq \exp(\varepsilon')\}$. Now note that

$$\Pr_P[B] \leq \exp(\varepsilon) \Pr_Q[B] + \delta \leq \exp(\varepsilon) \exp(-\varepsilon') \Pr_P[B] + \delta,$$

so that $\Pr_P[B] \leq \delta/(1 - \exp(\varepsilon - \varepsilon'))$. Setting $\varepsilon' = \varepsilon + \sqrt{2\delta}$, and noting that $\exp(-\sqrt{2\delta}) \leq 1 - \sqrt{\delta}$ for $\delta < \frac{1}{10}$, the claim follows. \square

Lemma E.3. Suppose that there is an event B such that $P \mid B^c \equiv_{\varepsilon, 0} Q \mid B^c$, and that $\Pr_{x \sim P}[B] \leq \delta$. Then $P \equiv_{\varepsilon, \delta} Q$.

Proof. Let S be any event. Then

$$\begin{aligned} \Pr_P[S] &\leq \Pr_{P \mid B^c}[S] + \Pr[B] \\ &\leq \exp(\varepsilon) \Pr_{Q \mid B^c}[S] + \Pr[B] \\ &\leq \exp(\varepsilon) \Pr_Q[S] + \delta. \end{aligned}$$

\square

²This can be ensured by having the mechanism outputting a uniform $[0, 1]$ r.v. in addition to its original output.