Differentially Private Online Submodular Minimization

Adrian Rivera Cardoso **Rachel Cummings**
Georgia Institute of Technology

Abstract

In this paper we develop the first algorithms for online submodular minimization that preserve differential privacy under full information feedback and bandit feedback. Our first result is in the full information setting, where the algorithm can observe the entire function after making its decision at each time step. We give an algorithm in this setting that is ϵ-differentially private and achieves expected regret $\tilde{O}\left(\frac{n\sqrt{T}}{\epsilon}\right)$ over T rounds for a collection of n elements. Our second result is in the bandit setting, where the algorithm can only observe the cost incurred by its chosen set, and does not have access to the entire function. This setting is significantly more challenging due to the limited information. Our algorithm using bandit feedback is ϵ-differentially private and achieves expected regret $\tilde{O}\left(\frac{n^{3/2}T^{2/3}}{\epsilon}\right)$.

1 Introduction

Online learning has received significant attention due to the growing amounts of information collected about individuals, and has been studied in the context of a wide variety of optimization problems, including portfolio optimization [8, 18, 21], shortest paths [27], combinatorial optimization [14], convex optimization [4, 15], and game theoretic optimization [6]. When these machine learning tools are applied to sensitive data from individuals, privacy concerns becoming increasingly important. In applications such as clinical trials, online ad placement, personalized pricing, and recommender systems, online learning algorithms are dealing with personal (and possibly highly sensitive) data.

In this paper, we develop the first algorithms for differentially private online submodular optimization. A function $f : 2^{[n]} \to [-M, M]$ mapping from discrete collections of elements to real values is *submodular* if it exhibits the following diminishing returns property: for all sets $S, S' \subseteq [n]$ such that $S' \subseteq S$ and for all elements $i \in [n] \setminus S$,

$$f_t(S' \cup \{i\}) - f_t(S') \geq f_t(S \cup \{i\}) - f_t(S).$$

Submodular functions have several applications in machine learning (see [22] for a survey) and are extensively used in economics because their diminishing returns property captures preferences for substitutable goods and satiation from multiple copies of the same good [2, 28].

In the *Online Submodular Minimization* problem, a sequence of T submodular functions $f_1, \ldots, f_T : 2^{[n]} \to [-M, M], M \geq 0$ arrive in an online fashion. At every timestep t, a decision maker chooses a set $S_t \subseteq [n]$ before observing the function f_t. The decision maker then incurs cost $f_t(S_t)$. The decision maker's goal is to minimize her total regret, which is defined as

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(S_t) - \min_{S \subseteq [n]} \sum_{t=1}^{T} f_t(S).$$

That is, her regret is the difference between her total cost across all rounds, and the cost of the best fixed set in hindsight after seeing all the functions. We say that an algorithm for the Online Submodular Minimization problem is *no regret* if the regret (or expected regret for randomized algorithms) is sublinear in T: $\text{Regret}(T) = o(T)$.

We consider two different settings based on the type of informational feedback the decision maker receives in each round. In the *full information setting*, the decision maker observes the entire function f_t

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

after making her choice of S_t. In the *bandit setting*, the decision maker only observes her cost $f_t(S_t)$ and does not receive any additional information about the function f_t. The bandit setting is a more challenging environment because the decision maker has severely restricted information when making decisions, but also captures the reality of many real-world online learning problems where counterfactual outcomes cannot be measured.

We formally incorporate the task of preserving privacy by using the framework of differential privacy. Differential privacy was first defined by [9] for algorithms operating on large static databases, and required that if a single entry in the database changed, then the algorithm would produce approximately the same output. In this work, we view our database as the sequence of submodular functions $f_1, \ldots, f_T$, and the algorithm's output is the sequence of sets $S_1, \ldots, S_T$. We require that if a single function f_t were changed to a different f'_t, then the entire sequence of chosen sets would be approximately the same. A formal definition is given in the preliminaries.

The main goal of this paper is to design differentially private no-regret algorithms for the Online Submodular Minimization problem. There are many applications of online learning problems using sensitive data that could benefit from formal privacy guarantees, such as clinical drug trials, online ad placement, and personalized pricing. For concreteness, we provide the following motivating example for the study of private online submodular optimization.

Motivating Example. As a concrete motivating example we consider the following online ad placement problem. Online retailers such as Amazon, Walmart, and Target design their websites such that the retailers can offer other products at check out which complement the item the customer is buying. Due to item complimentarities, the utility function of user t, g_t, defined over the possible subset of products the retailer can offer $[n]$, is supermodular. However, displaying too many items may hurt the chance of the user buying something else. At time t, the retailer is choosing S_t that maximizes $f_t(S) := g_t(S) - \sum_{i \in S} p_i$ for each user (where $p_i \in \mathbb{R}$ is the "cost" of displaying a product). The retailers must choose S_t without knowing g_t and they receive only bandit feedback (i.e., they can only observe $g_t(S_t)$, and not $g_t(\cdot)$). The retailer seeks to minimize regret: $\max_{S \in [n]} \sum_{t=1}^{T} f_t(S) - \sum_{t=1}^{T} f_t(S_t)$. Notice that since $\sum_{i \in S} p_i$ is modular, then the retailer has to solve an online submodular minimization problem with bandit feedback. Existing recommender systems have been shown to leak information about users [29], motivating the need for formal privacy guarantees in this settings. Therefore, the retailer will perform this optimization in a differentially private manner to ensure that no information about an individual is leaked to other users.

1.1 Our Results and Techniques

In this paper we develop the first algorithms for online submodular minimization that preserve differential privacy under full information feedback and bandit feedback that are almost as good as the best non-private algorithms.

We start with the full information setting, where the algorithm can observe the entire function f_t after making its decision at each time t. We give an algorithm in this setting that is both differentially private and satisfies no regret.

Theorem 1 (Informal)**.** *In the full information setting of Online Submodular Minimization, there is an ϵ-differentially private algorithm that achieves regret:*

$$\mathbb{E}[\text{Regret}(T)] = \tilde{O}\left(\frac{n\sqrt{T}}{\epsilon}\right).$$

This algorithm works by first relaxing each input submodular function to a convex function using the Lovasz extension (defined formally in Section 2.1). Our algorithm then simulates a variant of an algorithm for differentially private online convex optimization (due to Smith and Thakurta [26]) run on the sequence of Lovasz extensions. The differential privacy guarantee can be proved almost as it was done in [26]. To prove the regret bound, we show that the relaxation and optimization on convex functions does not increase the regret guarantee by too much. Our algorithm matches the regret bound of [26] for private online convex optimization, and loses only a factor of $\frac{1}{\epsilon}$ relative to the optimal non-private regret bound of [14] for online submodular minimization.

We next consider the bandit setting, which is significantly more challenging and requires a refined analysis. The private online convex optimization algorithm of Smith and Thakurta [26] requires use of the subgradient of the Lovasz extension. However in the bandit setting, the algorithm does not receive enough information to compute the exact Lovasz extension or its subgradients. Instead, we construct an unbiased estimate of the subgradient using the one-point estimation method of [14]. We then apply a variant of the algorithm from [26] to the unbiased estimate of the gradient of the Lovasz extension. This yields a differentially private no-regret algorithm for online submodular minimization in the bandit setting.

Theorem 2 (Informal)**.** *In the bandit setting of Online Submodular Minimization, there is an ϵ-differentially private algorithm that achieves regret:*

$$\mathbb{E}[\text{Regret}(T)] = \tilde{O}\left(\frac{n^{3/2}T^{2/3}}{\epsilon}\right).$$

The regret guarantee of our algorithm for the bandit setting only loses a factor of $\tilde{O}(\frac{n^{1/2}}{\epsilon})$ relative to the best know non-private regret bound of [14] for online submodular minimization. We actually improve upon the best known regret bound for private online convex optimization [26] which has $O(T^{3/4})$ dependence on T, compared to our $O(T^{2/3})$ guarantee.

1.2 Related Work

Our results rely on ideas from [26] and [14]. [26] provides a differentially private algorithm for online convex optimization that achieves a regret rate $\tilde{O}(\frac{\sqrt{nT}}{\epsilon})$ in the full information setting, which is worse than the non-private setting by only a factor of polylog$(T)\sqrt{n}$. Under bandit feedback, they give a modification of their full information algorithm that achieves cumulative regret $\tilde{O}(\frac{nT^{3/4}}{\epsilon})$. One of the key components in our algorithms are modifications of these tools for online convex optimization, which are applied once we have relaxed the submodular functions to their convex Lovasz extensions. [14] provides algorithms for non-private online submodular minimization in both the full information and bandit feedback settings. They design subgradient descent-type algorithms that achieve regret of $O(\sqrt{nT})$ and $O(nT^{2/3})$ in the full information and bandit settings respectively. Our algorithms make use of their one-point gradient estimation technique for the bandit setting. We remark that, to the best of our knowledge, there is no known way to modify subgradient descent-type algorithms, to achieve differential privacy in the online convex bandit problem without damaging the regret bounds by less than polylog(T) factors.

Although our algorithms use these tools, composition of these previous results is not straight-forward. The bound on the variance of the one-point gradient estimator for the Lovasz extension is not the same as that of the estimator used for online convex optimization with bandit feedback, which requires special care in the analysis. If one were to blindly compose the results of [26] and [14], it would yield regret $O(\frac{n^2T^{11/12}}{\epsilon})$ in the bandit setting, instead of the regret rate $O(\frac{n^{3/2}T^{2/3}}{\epsilon})$ that we achieve.

A previous (unpublished) version of the current paper [5] showed that a more careful combination of these tools, which takes into account the variance of the one-point gradient estimator for the Lovasz extension but uses the same analysis as in [26], can only achieve regret $\tilde{O}(\frac{n^{3/2}T^{3/4}}{\epsilon})$ in the bandit setting. This approach was unable to achieve the $\tilde{O}(T^{2/3})$ dependence on T that we achieve here because the analysis of [26] first gave differentially private regret guarantees for strongly convex cost function, and then extended these results to the setting with general convex costs via a regularization trick to ensure strong convexity (See Appendix E.3 from [26]). While this regularization trick allows for low regret, $\tilde{O}(T^{3/4})$, for the problem of private online convex optimization, there were dependencies in the regret bound which make it impossible to obtain the rate of $\tilde{O}(T^{2/3})$ for differentially private online submodular minimization. Our analysis requires additional techniques to achieve this lower regret bound.

Other relevant work includes [19], where the authors design differentially private algorithms for online convex optimization. However, these algorithms only achieve optimal regret rates in some special cases. In [1], the authors provide differentially private algorithms for the special case of online linear optimization with bandit feedback, and obtain regret $\tilde{O}(\frac{\sqrt{T}}{\epsilon})$ which is (almost) optimal. The problem of private online submodular maximization has been studied by [23] and [13]. However, our work cannot be compared to theirs since the problems of minimizing and maximizing a submodular functions are very different. Additionally, these works only consider the offline problem with full information feedback. Finally, [3] studies non-private online submodular maximization only under full information feedback.

2 Preliminaries

In this section we present background on submodular functions and differential privacy that will be useful for our results in later sections.

2.1 Submodular Functions

Submodular functions share many properties with both convex and concave functions. They can be thought of as convex functions when one is trying to minimize them, however they also exhibit a diminishing marginal returns property as some concave functions do (i.e., $f(x) = \log x$).

Definition 1 (Submodular function)**.** *A function* $f : 2^{[n]} \to [-M, M]$ *is* submodular *if for all sets* $S, S' \subseteq [n]$ *such that* $S' \subseteq S$ *and for all elements* $i \in [n] \setminus S$*,*

$$f(S' \cup i) - f(S') \geq f(S \cup i) - f(S).$$

The connection between convex and submodular functions is formalized through the *Lovasz extension* (Definition 3), which extends a submodular function f over $[n]$ to its corresponding convex function $\hat{f}$ over $[0,1]^n$. The Lovasz extension works by first describing each point in $[0,1]^n$ as a convex combination of points in $\{0,1\}^n$, which can be interpreted as subsets of $[n]$. It then defines $\hat{f}(x)$ as the convex combination of f evaluated on the sets associated with x. We first define the necessary notation.

Definition 2 (Maximal chain [14])**.** *A chain of subsets of* $[n]$ *is a collection of sets* $A_0, ..., A_p$ *such that* $A_0 \subset A_1 \subset \cdots \subset A_p$*. A chain is* maximal *if* $p = n$*. For a maximal chain,* $A_0 = \emptyset$*,* $A_n = [n]$*, and there is a unique associated permutation* $\pi : [n] \to [n]$ *such that* $A_{\pi(i)} = A_{\pi(i)-1} \cup \{i\}$ *for all* $i \in [n]$*. For this permutation, we can write* $A_{\pi(i)} = \{j \in [n] : \pi(j) \leq \pi(i)\}$ *for all* $i \in [n]$*.*

Define $\mathcal{K} = [0,1]^n$. For any set $S \subseteq [n]$, let $\mathcal{X}_S \in \{0,1\}^n$ denote the *characteristic vector* of S, defined as $\mathcal{X}_S(i) = 1$ if $i \in S$ and 0 otherwise. For any $x \in \mathcal{K}$, there is a unique chain $A_0 \subset \cdots \subset A_p$ such that x can be expressed as a convex combination of the characteristic vectors of the A_i. That is, $\exists \mu_1, \ldots, \mu_p > 0$ such that $x = \sum_{i=0}^{p} \mu_i \mathcal{X}_{A_i}$ and $\sum_{i=0}^{p} \mu_i = 1$. Note that if $p < n$ (i.e., the chain is not maximal), the chain can be extended to a maximal chain by setting $\mu_i = 0$ for all i's corresponding the the subsets of $[n]$ that were not present in the original chain. The chain and the weights can be found in $O(n \ln(n))$ time (see, e.g., Chap. 3 of Bach [2]).

We are now ready to define the Lovasz extension $\hat{f}$ of submodular function f.

Definition 3 (Lovasz extension)**.** *Let* $f : 2^{[n]} \to [-M, M]$ *be submodular. The* Lovasz extension $\hat{f} : \mathcal{K} \to [-M, M]$ *of* f *is defined as follows. For each* $x \in \mathcal{K}$*, let* $A_0 \subset \cdots \subset A_p$ *be the chain associated with* x*, and let* $\mu_1, \ldots, \mu_p$ *be the corresponding weights in the convex combination* $x = \sum_{i=0}^{p} \mu_i \mathcal{X}_{A_i}$*. Then,*

$$\hat{f}(x) := \sum_{i=0}^{p} \mu_i f(A_i) \quad \forall x \in \mathcal{K}.$$

Equivalently, the Lovasz extension can also be defined by sampling τ *uniformly at random from the unit interval* $[0,1]$ *and considering level set* $S_\tau = \{i : x(i) \geq \tau\}$*. Then* $\hat{f}(x) = \mathbb{E}_\tau[f(S_\tau)]$ *for each* $x \in \mathcal{K}$*.*

We now provide some useful properties of the Lovasz extension.

Lemma 1 ([12, 14])**.** *The Lovasz extension* $\hat{f}$ *of submodular function* f *is convex. Additionally, for any* $x \in \mathcal{K}$*, let* $\emptyset = B_0 \subset B_1 \subset \cdots \subset B_n$ *be any maximal chain associated with* x *and let* $\pi : [n] \to [n]$ *be the corresponding permutation. Then a subgradient* g *of* $\hat{f}$ *at* x *is given by:* $g(i) = f(B_{\pi(i)}) - f(B_{\pi(i)-1})$ *for all* $i = 1, \ldots, n$*.*

Lemma 2 ([20])**.** *All subgradients* g *of the Lovasz extension* $\hat{f} : \mathcal{K} \to [-M, M]$ *of a submodular function are bounded by* $\|g\|_2 \leq \|g\|_1 \leq 4M$*.*

2.2 Tools from Differential Privacy

Let $\mathcal{F}$ be a class of functions. Let $F = \{f_1, ..., f_T\}$ and $F' = \{f'_1, \ldots, f'_T\}$ be sequences of functions where $f_t, f'_t \in \mathcal{F}$, and $f_t, f'_t : \mathcal{R} \to \mathbb{R}$ for all t. We say F and F' are neighboring sequences if $f_t = f'_t$ for all but at most one $t \in [T]$.

Definition 4 (Differential privacy [9])**.** *An algorithm* $\mathcal{A} : \mathcal{F}^T \to \mathcal{R}^T$ *is* (ϵ, δ)-differentially private *if for all neighboring sequences* $F, F' \in \mathcal{F}^T$ *and every subset of the output space* $\mathcal{S} \subseteq \mathcal{R}^T$*,*

$$\Pr[\mathcal{A}(F) \in \mathcal{S}] \leq e^\epsilon P[\mathcal{A}(F') \in \mathcal{S}] + \delta.$$

If $\delta = 0$*, we say that* $\mathcal{A}$ *is* ϵ*-differentially private.*

The following theorem states that differential privacy is robust to *post-processing*: computations performed on the output of a differentially private algorithm are still differentially private.

Theorem 3 (Post-processing [9])**.** *Let* $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ *be* (ϵ, δ)*-differentially private, and let* $f : \mathcal{R} \to \mathcal{R}'$ *be an arbitrary randomized function. Then* $f \circ \mathcal{A} : \mathcal{D} \to \mathcal{R}'$ *is* (ϵ, δ)*-differentially private.*

Our results require another differentially private algorithm: Tree-based Aggregation Protocol (TBAP). The Tree-Based Aggregation Protocol [7, 10, 26] is a tool for maintaining differentially private partial sums of vectors arriving in an online sequence. At each time t, TBAP outputs a noisy sum of the input vectors up to time t. A full presentation of the algorithm and its properties is given in Appendix A.2.

The following section (Section 2.2.1) discusses Regularized Follow The Leader, an algorithm from [16] for online convex optimization which is used for online learning. Prior work [26] privatized a variant of this algorithm, Follow The Approximate Leader, to give a differentially private algorithm for online convex optimization that uses TBAP as a subroutine. It takes in a sequence of strongly convex functions and outputs a sequence of points that minimizes regret.

2.2.1 The Cost of Privacy in Online Convex Optimization

Our algorithm uses the following Regularized Follow the Leader (RFTL) of [16] as a subroutine for online convex optimization. This algorithm is known to achieve low regret (Theorem 4).

Algorithm 1 Regularized Follow The Leader: RFTL($\{f_i\}_{i=1}^T, H, X$)

```
Input: Online sequence of convex cost functions $\{f_1, ..., f_T\}$ strong convexity parameter $H$, convex compact decision set $X \subseteq \mathbb{R}^n$.
Output: Sequence of actions $x_1, \dots, x_T \in X$
Initialize $x_1 \leftarrow \arg\min_{x \in X} \frac{H}{2}||x||_2^2$
Output $x_1$, observe $f_1$
for t=1, ..., T-1 do
    $x_{t+1} \leftarrow \arg\min_{x \in X} \sum_{\tau=1}^t \nabla f_\tau(x_\tau)^\top x + \frac{H}{2}||x||^2$
    Output $x_{t+1}$ and observe $f_{t+1}$
end for
```

Theorem 4 ([17] Ch. 5)**.** *Let $\{f_t\}_{t=1}^T$ be any sequence of convex functions. Let $X \subseteq \mathbb{R}^n$ be a convex and compact set. RFTL guarantees that for any $x \in X$,*

$$\text{Regret}(T) \leq \frac{2}{H} \sum_{t=1}^T \|\nabla f_t(x_t)\|^2 + \frac{H}{2} \left[\|x\|^2 - \|x_1\|^2\right].$$

We give the following theorem, which quantifies the loss in regret due to adding a differential privacy constraint. A similar theorem was given in [26] for their analysis of a differentially private version of Follow The Approximate Leader, which is a variant of Regularized Follow the Leader. The main ideas in both proofs are similar, but we analyze a different algorithm (RFTL), so a new proof is needed for Theorem 5. The proof is given in the appendix.

Theorem 5. *Let $\{\hat{x}_t\}_{t=1}^T$ be the non private iterates of RFTL and let $\{x_t\}_{t=1}^T$ be the private iterates i.e. $x_{t+1} = \arg\min_{x \in X} v_t^\top x + \frac{H}{2}||x||^2$ where v_t is the private partial sum computed using $TBAP\{\nabla f_t(x_t), L, \epsilon\}$. It holds that*

$$\mathbb{E}[\sum_{t=1}^T f_t(x_t)] \leq \mathbb{E}[\sum_{t=1}^T f_t(\hat{x}_t)] + \frac{4nL^2 T \ln^{1.5}(T)}{\epsilon H}.$$

Where the expectation is taken with respect to the randomness of TBAP.

3 Full Information Setting

In this section we present Submodular Private Regularized Follow The Leader (SubmodPRFTL) which is an algorithm for Online Submodular Minimization that is both differentially private and achieves near optimal regret. In the full information setting, the result follows easily from using RFTL together with TBAP to compute a private version of the sum $\sum_{j=1}^t \nabla f_j(x_j)$.

The main difference between using a Regularized Follow The Leader type algorithm versus the subgradient descent type algorithm of [14] is the following. When using SubmodPRFTL to make the decision at time $t+1$, we use all the subgradients we have observed at times $1, \dots, t$. To contrast, if we used the algorithm of [14], we would only be using the subgradient obtained at t. This difference is crucial when trying to incorporate privacy into the setting.

Algorithm 2 Submodular Private Regularized Follow The Leader: SubmodPRFTL($\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon$)

```
Input: Online sequence of submodular cost functions $\{f_1, ..., f_T\}$, lower and upper bounds function values $[-M, M]$, strong convexity parameter $H$, Lipschitz parameter $L$, ground set $[n]$, privacy parameter $\epsilon$.
Output: Sequence of sets $S_1, \dots, S_T \subseteq [n]$
Initialize $S_1 \leftarrow \emptyset$
Set $x_1 \leftarrow 0 \in \mathcal{K}$
Output $S_1$
Compute and pass $\nabla \hat{f}_1(x_1)$ to TBAP($\{\nabla \hat{f}_i(x_i)\}, L, \epsilon$), and receive current partial sum $v_1$
for t=1, ..., T-1 do
    $x_{t+1} \leftarrow \arg\min_{x \in \mathcal{K}} v_t^\top x + \frac{H}{2}\|x\|_2^2$
    Sample $\tau_{t+1} \sim U[0,1]$
    Output $S_{t+1} = \{i : x_{t+1}(i) > \tau_t\}$ and observe $f_{t+1}$
    Compute $\nabla \hat{f}_t(x_{t+1})$ and pass $\nabla \hat{f}_{t+1}(x_{t+1})$ to TBAP($\{\nabla \hat{f}_i(x_i)\}, L, \epsilon$), and receive current partial sum $v_{t+1}$
end for
```

Algorithm 2 is differentially private (Theorem 6) and achieves $\tilde{O}(\sqrt{T})$ regret (Theorem 7).

Theorem 6 (Privacy guarantee)**.** SubmodPRFTL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon)$ *is* ϵ*-differentially private for any sequence of functions* $f_1, \ldots, f_T$ *with bounded range* $[-M, M]$ *and for any* $M, H, L, n, T > 0$.

Theorem 7 (Regret guarantee)**.** SubmodPRFTL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon)$ *run with* $H = M\sqrt{T}$ *and* $||\nabla \hat{f}_t|| \leq L = 4M$ *for any sequence of submodular functions* $f_1, \ldots, f_T : 2^{[n]} \to [-M, M]$ *for any* $M, n, T > 0$ *guarantees,*

$$\mathbb{E}[\text{Regret}(T)] \leq O\left(\frac{nM^2 \ln^{1.5}(T)\sqrt{T}}{\epsilon}\right),$$

where the expectation is taken over the randomness of TBAP *and the sampling procedure used to choose* S_t.

4 Bandit Setting

In this section we present Submodular Private Follow The Approximate Leader with Bandit Feedback (BanditSubmodPRFTL). This algorithm is differentially private and achieves a no regret guarantee for online submodular minimization with bandit feedback. The regret bound only loses a factor of $O(n^{1/2} \log^{1.5}(T))$ relative to the best known algorithm in the non-private setting.

The bandit setting makes the problem much more challenging because we do not have access to the whole function f_t nor to its subgradients. Instead we only observe the function evaluated at a single point, $f_t(S_t)$ for our chosen set S_t. This means that we can no longer compute subgradients of the Lovasz extension $\nabla \hat{f}_t$ and run RFTL on functions $\hat{f}_t$ as in the full information setting.

The key to obtaining sublinear regret is to balance exploration and exploitation. In this setting, exploitation is achieved by sampling S_t exactly from the distribution μ defined (through the Lovasz extension) by iterate x_t of BanditSubmodPRFTL.

However, if we sample according to the distribution over sets μ, we do not learn anything about the function's subgradients so, it is unclear what to do in future steps. To fix this, we should sample from some distribution that is close to μ, that allows us to explore (i.e., obtain an unbiased estimate of the Lovasz extension at x_t). We use the sampling procedure from Hazan and Kale [14] to achieve this. With these modifications, BanditSubmodPRFTL now works similarly to SubmodPRFTL for the full information setting. The algorithm works by computing an unbiased estimator $\hat{g}_t$ of the gradient of the Lovasz extension $\nabla \hat{f}_t$, updating a private iterate $x_t \in \mathcal{K}$ using TBAP to obtain a private partial sum of $\sum_{j=1}^t \hat{g}_t$, and outputting a random set S_t that depends on x_t. We now present the full algorithm BanditSubmodPRFTL in Algorithm 3.

Algorithm 3 Submodular Private Regularized Follow The Leader with Bandit Feedback: BanditSubmodPRFTL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$

```
**Input:** Online sequence of submodular cost functions $\{f_1, ..., f_T\}$, lower and upper bounds function values $[-M, M]$, strong convexity parameter $H$, Lipschitz parameter $L$, ground set $[n]$, privacy parameter $\epsilon$, parameter $\gamma \in (0, 1)$.
**Output:** Sequence of sets $S_1, \ldots, S_T \subseteq [n]$
Initialize $x_i \leftarrow \arg\min_{x \in \mathcal{K}} ||x||^2$
**for** t=1, . . . , T **do**
    Find maximal chain associated with $x_t$, $\emptyset = B_0 \subset B_1 \subset B_2 \subset \cdots B_n = [n]$, let $\pi$ be the associated permutation
    Write $x_t$ as $x_t = \sum_{i=0}^n \mu_i \mathcal{X}_{B_i}$, where $\mu_i = 0$ for the extra sets $B_i$ that where added to complete the maximal chain for $x_t$.
    Sample $S_t$ according to distribution: $S_t = B_i$ with probability $\rho_i = (1-\gamma)\mu_i + \frac{\gamma}{n+1}$
    Output $S_t$ and observe $f_t(S_t)$
    **if** $S_t = B_0$ **then**
        Set $\hat{g}_t = -\frac{1}{\rho_0} f_t(B_0) e_{\pi^{-1}(1)}$
    **else if** $S_t = B_n$ **then**
        Set $\hat{g}_t = \frac{1}{\rho_n} f_t(B_n) e_{\pi^{-1}(n)}$
    **else**
        Choose $\xi \in \{+1, -1\}$ uniformly at random
        **if** $\xi = +1$ **then**
            Set $\hat{g}_t = \frac{2}{\rho_i} f_t(B_i) e_{\pi^{-1}(i)}$
        **else**
            Set $\hat{g}_t = -\frac{2}{\rho_i} f_t(B_i) e_{\pi^{-1}(i+1)}$
        **end if**
    **end if**
    Pass $\hat{g}_t$ to TBAP$(\{\hat{g}_i\}, L, \epsilon)$, and receive current partial sum $\hat{v}_t$
    Update $x_{t+1} = \arg\min_{x \in \mathcal{K}} \hat{v}_t^\top x + \frac{H}{2}||x||^2$
**end for**
```

In the algorithm e_i refers to the vector with all entries equal to 0 except for the i-th entry which is equal to 1. The analysis of BanditSubmodPRFTL relies on the following key properties of the estimate $\hat{g}$.[1] Proofs are deferred to the appendix.

Lemma 3. *Let $\gamma \in (0,1)$. The random vector $\hat{g}_t$ computed in* BanditSubmodPRFTL *is an unbiased estimate of a subgradient of the Lovasz extension $\hat{f}_t$ of submodular f_t, evaluated at point x_t. That is,*

$$\mathbb{E}\left[\hat{g}_t \mid x_t\right] = \nabla \hat{f}_t(x_t).$$

Lemma 4. *The random vector $\hat{g}_t$ computed in* BanditSubmodPRFTL *satisfies the following bound on its expected L_2-norm,*

$$\mathbb{E}\left[\|\hat{g}_t\|^2\right] \leq \frac{16M^2n^2}{\gamma},$$

where the expectation is taken over the algorithm's internal randomness up to time t.

The exploration-exploitation dilemma can be better understood through the parameter γ. This parameter trades off between variance of the estimate $\hat{g}_t$ and the approximation of the Lovasz extension $\hat{f}_t$ to the true submodular function f_t. When γ is large, the variance of $\hat{g}_t$ is diminished, as can be seen in the statement of Lemma 4. When γ is small, the performance of $f_t(S_t)$ is close to that of $\hat{f}_t(x_t)$ (see Lemma 5 in Section 4.1). In the statement of our main result (Theorem 9), we optimally tune γ to balance exploration and exploitation and minimize overall regret of BanditSubmodPRFTL.

Our two main results of this section show that BanditSubmodPRFTL is differentially private and achieves low regret.

Theorem 8 (Privacy guarantee)**.** BanditSubmodPRFTL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$ *is ϵ-differentially private for any sequence of functions $f_1, \ldots, f_T$ with bounded range $[-M, M]$ and for any $M, H, L, n, T, \gamma > 0$.*

Theorem 9 (Regret guarantee)**.** BanditSubmodPRFTL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$ *run with $H = MT^{2/3}$, $L = \frac{4Mn}{\sqrt{\gamma}}$, and $\gamma = \frac{n^{3/2}}{T^{1/3}}$ for any sequence of submodular functions $f_1, \ldots, f_T : 2^{[n]} \to [-M, M]$ for any $M, n, T > 0$ guarantees,*

$$\mathbb{E}[\text{Regret}(T)] \leq \tilde{O}\left(\frac{MnT^{2/3}}{\epsilon}\right),$$

where the expectation is taken with respect to all the internal randomness of the algorithm.

The proof of Theorem 9 relies on several key lemmas, presented in Section 4.1.

4.1 Regret Analysis of BanditSubmodPRFTL

There are several sources of potential sub-optimality in the output of BanditSubmodPRFTL that must be bounded. Firstly, the algorithm optimizes using continuous iterates x_t instead of discrete (Lemma 5). The algorithm incurs additional loss from the noise added in TBAP to preserve privacy (Lemma 8). Lastly, due to the bandit feedback, we cannot compute an exact subgradient of the regularized Lovasz extension, and must instead use a (random) unbiased estimator (Lemmas 6 and 7).

The following lemmas bound the regret from these sources of error, and are used in the proof of Theorem 9. All omitted proofs are presented in the appendix.

We start with a lemma from Hazan and Kale [14], showing that the additional loss from choosing a subset of the ground set S_t instead of the point in $x_t \in \mathcal{K}$ is not too large.

Lemma 5 ([14])**.** *For any submodular function $f_t : [n] \to [-M, M]$, let x_t and S_t be the corresponding iterates and sets as defined in* BanditSubmodPRFTL*, then $\mathbb{E}[f_t(S_t)] \leq \mathbb{E}[\hat{f}_t(x_t)] + 2\gamma M$. Where the expectation is taken with respect to all the randomness of the algorithm.*

The following lemma bounds the regret loss due to the fact that we only have bandit feedback. The main idea of the proof comes from [11], the first paper that provided an algorithm for online convex optimization with bandit feedback, however we must modify it accordingly to account for the fact that our one-point gradient estimator is for the Lovasz extension of a submodular function and not just any convex function. This modification will exploit the bound on the variance of $\hat{g}_t$ from Lemma 4 and will allow us to prove a regret rate of $\tilde{O}(T^{2/3})$ instead of $\tilde{O}(T^{3/4})$ which is obtained for general convex functions while trying to preserve privacy (see [26]).

The next lemma bounds the loss our algorithm incurs due to bandit feedback against an adaptive adversary. The key to prove such a result is to bound with probability one the absolute difference between $\sum_{t=1}^T \nabla \hat{f}_t(x_t)$ and $\sum_{t=1}^T \nabla \hat{g}_t$, then use the fact that $\hat{g}_t$ is an unbiased estimator of $\nabla \hat{f}_t$.

Lemma 6. *Let $\{\hat{g}_t\}_{t=1}^T$ be the sequence of one point gradient estimates generated by*

[1] Our Lemmas 3 and 4 were asserted without proof in [14]. Due to minor errors in the construction of $\hat{g}_t$ in [14], these claims are easily seen to be false under their construction. Here, we build the correct estimator and prove its correctness.

BanditSubmodPRFTL*($\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma$). Then,*

$$\mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{g}_t^\top x\right] \leq \mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \nabla \hat{f}_t^\top x\right] + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}},$$

where the expectation is taken with respect to all the randomness of the algorithm.

Lemma 7. *Let $\{\hat{g}_t\}_{t=1}^T$ and $\{x_t\}_{t=1}^T$ be the sequences generated by* BanditSubmodPRFTL*($\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma$). Then,*

$$\mathbb{E}[\sum_{t=1}^{T} \hat{g}_t^\top x_t] = \mathbb{E}[\sum_{t=1}^{T} \nabla \hat{f}_t^\top x_t],$$

where the expectation is taken with respect to all the randomness of the algorithm.

The following lemma quantifies the loss in the regret due to privacy.

Lemma 8. *Let $\{x_t\}_{t=1}^T$ be the sequence generated by* BanditSubmodPRFTL*($\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma$). Let $\hat{x}_t$ be the non private iterate of the algorithm, that is $\hat{x}_{t+1} = \sum_{\tau=1}^{t} \hat{g}_\tau^\top x + \frac{H}{2}||x||^2$. Then,*

$$\mathbb{E}[\sum_{t=1}^{T} \hat{g}_t^\top x_t] \leq \mathbb{E}[\sum_{t=1}^{T} \hat{g}_t^\top \hat{x}_t] + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H},$$

where the expectation is taken with respect to the randomness of the algorithm.

We are now ready to prove the regret guarantee of BanditSubmodPRFTL. A complete proof is given in the appendix, and we sketch the proof outline here.

To prove Theorem 9 we combine Lemmas 4, 5, 6, 7, 8 and the no regret guarantee of RTFL to upper bound the expected regret by:

$$\frac{32M^2n^2T}{H\gamma} + nH + 2\gamma MT +$$
$$\frac{8Mn\sqrt{T}}{\sqrt{\gamma}} + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H}$$

Choosing $\gamma = \frac{n^{3/2}}{T^{1/3}}$, $H = MT^{2/3}$ yields the result.

5 Acknowledgements

Adrian Rivera Cardoso was supported in part by a TRIAD-NSF grant (award 1740776). Rachel Cummings was supported in part by a Mozilla Research Grant.

References

[1] N. Agarwal and K. Singh. The price of differential privacy for online learning. *arXiv preprint arXiv:1701.07953*, 2017.

[2] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3): 145–373, 2013.

[3] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 671–680, 2014.

[4] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

[5] A. R. Cardoso and R. Cummings. Differentially private online submodular optimization. *arXiv preprint arXiv:1807.02290*, 2018.

[6] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[7] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 14 (3):1–24, 2011.

[8] T. M. Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, 2006.

[10] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, 2010.

[11] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[12] S. Fujishige. *Direct Submodular Functions and Optimization.* Annals of Discrete Mathematics. Elsevier, 2005.

[13] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1106–1125, 2010.

[14] E. Hazan and S. Kale. Online submodular minimization. *Journal of Machine Learning Research*, 13:2903–2922, 2012.

[15] E. Hazan and S. Kale. An optimal algorithm for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15:2489–2512, 2014.

[16] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

[17] E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[18] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[19] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *Proceedings of the 25th Annual Conference on Learning Theory*, COLT '12, pages 1–34, 2012.

[20] S. Jegelka and J. Blimes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1897–1904, 2011.

[21] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3:423–440, 2002.

[22] A. Krause and C. Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):1–20, 2011.

[23] M. Mitrovic, M. Bun, A. Krause, and A. Karbasi. Differentially private submodular maximization: Data summarization in disguise. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2478–2487, 2017.

[24] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[25] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[26] A. Smith and A. Thakurta. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS '13, pages 2733–2741, 2013.

[27] E. Takimoto and M. K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.

[28] D. M. Topkis. *Supermodularity and complementarity.* Princeton University Press, 2011.

[29] B. Zhang, N. Wang, and H. Jin. Privacy concerns in online recommender systems: Influences of control and user data input. In *Proceedings of 10th Symposium On Usable Privacy and Security*, SOUPS '14, pages 159–173, 2014.