# Fast and Private Submodular and $k$-Submodular Functions Maximization with Matroid Constraints

**Akbar Rafiey** [1] **Yuichi Yoshida** [2]

## Abstract

The problem of maximizing nonnegative monotone submodular functions under a certain constraint has been intensively studied in the last decade, and a wide range of efficient approximation algorithms have been developed for this problem. Many machine learning problems, including data summarization and influence maximization, can be naturally modeled as the problem of maximizing monotone submodular functions. However, when such applications involve sensitive data about individuals, their privacy concerns should be addressed. In this paper, we study the problem of maximizing monotone submodular functions subject to matroid constraints in the framework of differential privacy. We provide $(1-\frac{1}{\mathrm{e}})$-approximation algorithm which improves upon the previous results in terms of approximation guarantee. This is done with an almost cubic number of function evaluations in our algorithm.

Moreover, we study $k$-submodularity, a natural generalization of submodularity. We give the first $\frac{1}{2}$-approximation algorithm that preserves differential privacy for maximizing monotone $k$-submodular functions subject to matroid constraints. The approximation ratio is asymptotically tight and is obtained with an almost linear number of function evaluations.

## 1. Introduction

A set function $F\colon 2^E \to \mathbb{R}$ is *submodular* if for any $S \subseteq T \subseteq E$ and $e \in E \setminus T$ it holds that $F(S \cup \{e\}) - F(S) \geq F(T \cup \{e\}) - F(T)$. The theory of *submodular maximization* provides a general and unified framework for various combinatorial optimization problems including the Maximum Coverage, Maximum Cut, and Facility Location problems. Furthermore, it also appears in a wide variety of applications such as viral marketing (Kempe et al., 2003), information gathering (Krause & Guestrin, 2007), feature selection for classification (Krause & Guestrin, 2005), influence maximization in social networks (Kempe et al., 2003), document summarization (Lin & Bilmes, 2011), and speeding up satisfiability solvers (Streeter & Golovin, 2008). For a survey, see (Krause & Golovin, 2014). As a consequence of these applications and importance, a wide range of efficient approximation algorithms have been developed for maximizing submodular functions subject to different constraints (Călinescu et al., 2011; Nemhauser & Wolsey, 1978; Nemhauser et al., 1978; Vondrák, 2008).

The need for efficient optimization methods that guarantee the privacy of individuals is wide-spread across many applications concerning sensitive data about individuals, e.g., medical data, web search query data, salary data, social networks. Let us motivate privacy concerns by an example.

**Example 1.1** (Feature Selection (Krause & Guestrin, 2005; Mitrovic et al., 2017))**.** A sensitive dataset $D = \{(\mathbf{x}_i, C_i)\}_{i=1}^n$ consists of a feature vector $\mathbf{x}_i = (\mathbf{x}_i(1), \ldots, \mathbf{x}_i(m))$ associated to each individual $i$ together with a binary class label $C_i$. The objective is to select a small (e.g., size at most $k$) subset $S \subseteq [m]$ of features that can provide a good classifier for $C$. One particular example for this setting is determining collection of features such as height, weight, and age that are most relevant in predicting if an individual is likely to have a particular disease such as diabetes and HIV. One approach to address the feature selection problem, due to Krause & Guestrin (2005), is based on maximizing a submodular function which captures the mutual information between a subset of features and the class label of interest. Here, it is important that the selection of relevant features does not compromise the privacy of any individual who has contributed to the training dataset.

*Differential privacy* is a rigorous notion of privacy that allows statistical analysis of sensitive data while providing strong privacy guarantees. Basically, differential privacy requires that computations be insensitive to changes in any particular individual's record. A dataset is a collection of

[1]Department of Computing Science, Simon Fraser University, Burnaby, Canada [2]National Institute of Informatics, Tokyo, Japan. Correspondence to: Akbar Rafiey <arafiey@sfu.ca>, Yuichi Yoshida <yyoshida@nii.ac.jp>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Online, PMLR 119, 2020. 

records from some domain, and two datasets are *neighboring* if they differ in a single record. Simply put, the requirement for differential privacy is that the computation behaves nearly identically on two neighboring datasets; Formally, for $\epsilon, \delta \in \mathbb{R}_+$, we say that a randomized computation $M$ is $(\epsilon, \delta)$-*differentially private* if for any neighboring datasets $D \sim D'$, and for any set of outcomes $S \subseteq \text{range}(M)$,

$$\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.$$

When $\delta = 0$, we say $M$ is $\epsilon$-*differentially private*. Differentially private algorithms must be calibrated to the *sensitivity* of the function of interest with respect to small changes in the input dataset.

In this paper we consider designing a differentially private algorithm for maximizing nonnegative and *monotone* submodular functions in *low-sensitivity* regime. Whilst, a *cardinality* constraint (as in Example 1.1) is a natural one to place on a submodular maximization problem, many other problems, e.g., personalized data summarization (Mirzasoleiman et al., 2016), require the use of more general types of constraints, i.e., *matroid* constraints. The problem of maximizing a submodular function under a matroid constraint is a classical problem (Edmonds, 1971), with many important special cases, e.g., uniform matroid (the subset selection problem, see Example 1.1), partition matroid (submodular welfare/partition problem). We consider the following.

**Problem 1.1.** Given a sensitive dataset $D$ associated to a monotone submodular function $F_D \colon 2^E \to \mathbb{R}_+$ and a matroid $\mathcal{M} = (E, \mathcal{I})$. Find a subset $S \in \mathcal{I}$ that approximately maximizes $F_D$ in a manner that guarantees differential privacy with respect to the input dataset $D$.

Furthermore, we consider a natural generalization of submodular functions, namely, $k$-submodular functions. $k$-submodular function maximization allows for richer problem structure than submodular maximization. For instance, coupled feature selection (Singh et al., 2012), sensor placement with $k$ kinds of measures (Ohsaka & Yoshida, 2015), and influence maximization with $k$ topics can be expressed as $k$-submodular function maximization problems. To motivate the privacy concerns, consider the next example. More examples are given in Section 5.2.

**Example 1.2** (Influence Maximization with $k$ Topics)**.** For $k$ topics, a sensitive dataset is a directed graph $G = (V, E)$ with an edge probability $p^i_{u,v}$ for each edge $(u, v) \in E$, representing the strength of influence from $u$ to $v$ on the $i$-th topic. The goal is to distribute these topics to $N$ vertices of the graph so that we maximize *influence spread*. The problem of maximizing influence spread can be formulated as $k$-submodular function maximization problem (Ohsaka & Yoshida, 2015). An example for this setting is in viral marketing where dataset consists of a directed graph where each vertex represents a user and each edge represents the friendship between a pair of users. Given $k$ kinds of products, the objective is to promote products by giving (discounted) items to a selected group of influential people in the hope that large number of product adoptions will occur. Here, besides maximizing the influence spread, it is important to preserve the privacy of individuals in the dataset.

**Problem 1.2.** Given a sensitive dataset $D$ associated to a monotone $k$-submodular function $F_D \colon (k+1)^E \to \mathbb{R}_+$ and a matroid $\mathcal{M} = (E, \mathcal{I})$. Find $S = (S_1, \ldots, S_k)$ with $\bigcup_{i \in [k]} S_i \in \mathcal{I}$ that approximately maximizes $F_D$ in a manner that guarantees differential privacy with respect to the input dataset $D$.

### 1.1. Our Contributions

**Submodular Maximization:** For maximizing a nonnegative monotone submodular function subject to a matroid constraint, we show that a modification of the *continuous greedy* algorithm (Călinescu et al., 2011) yields a good approximation guarantee as well as a good privacy guarantee. Following the same idea, we maximize the so-called *multilinear extension* of the input submodular function in the corresponding *matroid polytope*, denoted by $\mathcal{P}(\mathcal{M})$. However, in order to greedily choose a direction, it requires to have a *discretization* of the matroid polytope. Fortunately, due to Yoshida (2019), an efficient discretization can be achieved. That is, we can *cover* a polytope with a small number of balls in polynomial time. Having these in hand, we prove the following.

**Theorem 1.1.** *Suppose $F_D$ is monotone with sensitivity $\Delta$ and $\mathcal{M} = (E, \mathcal{I})$ is a matroid. For every $\epsilon > 0$, there is an $(\epsilon r(\mathcal{M})^2)$-differentaily private algorithm that, with high probability, returns $S \in \mathcal{I}$ with quality at least $(1 - \frac{1}{\mathrm{e}})OPT - O\left(\sqrt{\epsilon} + \frac{\Delta r(\mathcal{M})|E| \ln |E|}{\epsilon^3}\right)$.*

For covering $C$ of $\mathcal{P}(\mathcal{M})$, the algorithm in Theorem 1.1 makes $O(r(\mathcal{M})|E||C|)$ queries to the *evaluation oracle*. We point out that $C$ has a size of roughly $|E|^{1/\epsilon^2}$. In Section 4, we present an algorithm that makes significantly fewer queries to the evaluation oracle.

**Theorem 1.2.** *Suppose $F_D$ is monotone and has sensitivity $\Delta$ and $\mathcal{M} = (E, \mathcal{I})$ is a matroid. For every $\epsilon > 0$, there is an $(\epsilon r(\mathcal{M})^2)$-differentaily private algorithm that, with high probability, returns $S \in \mathcal{I}$ with quality at least $(1 - \frac{1}{\mathrm{e}})OPT - O\left(\sqrt{\epsilon} + \frac{\Delta r(\mathcal{M})|E| \ln(|E|/\epsilon)}{\epsilon^3}\right)$. Moreover, this algorithm makes at most $O(r(\mathcal{M})|E|^2 \ln \frac{|E|}{\epsilon})$ queries to the evaluation oracle.*

**$k$-submodular Maximization:** To the best of our knowledge, there is no algorithm for maximizing $k$-submodular functions concerning differential privacy. We study Problem 1.2 in Section 5. First, we discuss an $(\epsilon r(\mathcal{M}))$-differentially private algorithm that uses the evaluation oracle at most $O(kr(\mathcal{M})|E|)$ times and outputs a solution with

quality at least $1/2$ of the optimal one.

**Theorem 1.3.** *Suppose $F_D : (k+1)^E \to \mathbb{R}_+$ is monotone and has sensitivity $\Delta$. For any $\epsilon > 0$, there is an $O(\epsilon r(\mathcal{M}))$-differentially private algorithm that, with high probability, returns a solution $X = (X_1, \ldots, X_k) \in (k+1)^E$ with $\bigcup_{i\in[k]} X_i \in \mathcal{I}$ and $F_D(X) \geq \frac{1}{2}\mathrm{OPT} - O(\frac{\Delta r(\mathcal{M}) \ln |E|}{\epsilon})$ by evaluating $F_D$ at most $O(kr(\mathcal{M})|E|)$ times.*

This $1/2$ approximation ratio is asymptotically tight due to the hardness result in (Iwata et al., 2016). Applying a sampling technique (Mirzasoleiman et al., 2015; Mitrovic et al., 2017; Ohsaka & Yoshida, 2015), we propose an algorithm that preserves the same privacy guarantee and the same quality as before while evaluating $F_D$ almost linear number of times, namely $O\left(k|E| \ln r(\mathcal{M}) \ln \frac{r(\mathcal{M})}{\gamma}\right)$. Here, $\gamma$ is the failure probability of our algorithm.

### 1.2. Related Works

Gupta et al. (2010) considered an important case of Problem 1.1 called the *Combinatorial Public Projects* (CPP problem). The CPP problem was introduced by Papadimitriou et al. (2008) and is as follows. For a data set $D = (x_1, \ldots, x_n)$, each individual $x_i$ submits a *private* non-decreasing and submodular valuation function $F_{x_i} : 2^E \to [0, 1]$. Our goal is to select a subset $S \subseteq E$ of size $k$ to maximize function $F_D$ that takes the particular form $F_D(S) = \frac{1}{n}\sum_{i=1}^{n} F_{x_i}(S)$. Note that in this setting, the sensitivity can be always bounded from above by $\frac{1}{n}$. Gupta *et al.* showed the following.

**Theorem 1.4** (Gupta et al. (2010))**.** *For any $\delta \leq 1/2$, there is an $(\epsilon, \delta)$-differentially private algorithm for the CPP problem under cardinality constraint that, with high probability, returns a solution $S \subseteq E$ of size $k$ with quality at least $(1 - \frac{1}{\mathrm{e}})\mathrm{OPT} - O(\frac{k \ln(\mathrm{e}/\delta) \ln |E|}{\epsilon})$.*

There are many cases which do not fall into the CPP framework. For some problems, including feature selection via mutual information (Example 1.1), the submodular function $F_D$ of interest depends on the dataset $D$ in ways much more complicated than averaging functions associated to each individual. Unfortunately, the privacy analysis of Theorem 1.4 heavily relies on the assumption that the input function $F_D = \frac{1}{n}\sum_{i=1}^{n} F_{x_i}(S)$ is the average of $F_{x_i}$'s, and does not directly generalize to arbitrary submodular functions. Using a *composition theorem* for differentially private mechanisms, Mitrovic et al. (2017) proved the following

**Theorem 1.5** (Mitrovic et al. (2017))**.** *Suppose $F_D$ is monotone and has sensitivity $\Delta$. For any $\epsilon > 0$, there is a $(k\epsilon)$-differentially private algorithm that, with high probability, returns $S \subseteq E$ of size $k$ with quality at least $\left(1 - \frac{1}{\mathrm{e}}\right)\mathrm{OPT} - O\left(\frac{\Delta k \ln |E|}{\epsilon}\right)$.*

In the same work, Mitrovic et al. (2017) considered matroid constraints and more generally *p-extendable* constraints.

**Theorem 1.6** (Mitrovic et al. (2017))**.** *Suppose $F_D$ is monotone with sensitivity $\Delta$ and let $\mathcal{M} = (E, \mathcal{I})$ be a matroid. Then for any $\epsilon > 0$, there is an $(\epsilon r(\mathcal{M}))$-differentially private algorithm that, with high probability, returns a solution $S \in \mathcal{I}$ with quality at least $\frac{1}{2}\mathrm{OPT} - O\left(\frac{\Delta r(\mathcal{M}) \ln |E|}{\epsilon}\right)$.*

**$k$-submodular Maximization:** The terminology for $k$-submodular functions was first introduced in (Huber & Kolmogorov, 2012) while the concept has been studied previously in (Cohen et al., 2006). Note for $k = 1$ the notion of $k$-submodularity is the same as submodularity. For $k = 2$, this notion is known as *bisubmodularity*. Bisubmodularity arises in bicooperative games (Bilbao et al., 2008) as well as variants of sensor placement problems and coupled feature selection problems (Singh et al., 2012). For unconstrained nonnegative $k$-submodular maximization, Ward & Zivny (2014) proposed a $\max\{1/3, 1/(1+a)\}$-approximation algorithm where $a = \max\{1, \sqrt{(k-1)/4}\}$. The approximation ratio was improved to $1/2$ by Iwata et al. (2016). They also provided $k/(2k-1)$-approximation for maximization of monotone $k$-submodular functions. The problem of maximizing a monotone $k$-submodular function was considered by Ohsaka & Yoshida (2015) subject to different constraints. They gave a $1/2$-approximation algorithm for total size constraint, i.e., $|\bigcup_{i\in[k]} X_i| \leq N$, and $1/3$-approximation algorithm for individual size constraints, i.e., $|X_i| \leq N_i$ for $i = 1, \ldots, k$. Sakaue (2017) proved that $1/2$-approximation can be achieved for matroid constraint, i.e., $\bigcup_{i\in[k]} X_i \in \mathcal{I}$.

## 2. Preliminaries

For a set $S \subseteq E$, $\mathbf{1}_S \in \mathbb{R}^E$ denotes the characteristic vector of $S$. For a vector $\mathbf{x} \in \mathbb{R}^E$ and a set $S \subseteq E$, $\mathbf{x}(S)$ denotes the sum $\sum_{e\in S} \mathbf{x}(e)$.

### 2.1. Submodular Functions

Let $F\colon 2^E \to \mathbb{R}_+$ be a set function. We say that $F$ is *monotone* if $F(S) \leq F(T)$ holds for every $S \subseteq T \subseteq E$. We say that $F$ is *submodular* if $F(S \cup \{e\}) - F(S) \geq F(T \cup \{e\}) - F(T)$ holds for any $S \subseteq T \subseteq E$ and $e \in E \setminus T$.

The *multilinear extension* $f\colon [0,1]^E \to \mathbb{R}$ of a set function $F\colon 2^E \to \mathbb{R}$ is $f(\mathbf{x}) = \sum_{S\subseteq E} F(S) \prod_{e\in S} \mathbf{x}(e) \prod_{e\notin S} (1 - \mathbf{x}(e))$.

There is a probabilistic interpretation of the multilinear extension. Given $\mathbf{x} \in [0,1]^E$ we can define $X$ to be the random subset of $E$ in which each element $e \in E$ is included independently with probability $\mathbf{x}(e)$ and is not included with probability $1 - \mathbf{x}(e)$. We write $X \sim \mathbf{x}$ to denote that $X$ is a random subset sampled this way from $\mathbf{x}$. Then we can simply write $f$ as $f(\mathbf{x}) = \mathbb{E}_{X\sim\mathbf{x}}[F(X)]$.

Observe that for all $S \subseteq E$ we have $f(\mathbf{1}_S) = F(S)$. The following is well known:

**Proposition 2.1** (Călinescu et al. (2011)). *Let $f\colon [0,1]^E \to \mathbb{R}$ be the multilinear extension of a monotone submodular function $F\colon 2^E \to \mathbb{R}$. Then*

1. *$f$ is monotone, meaning $\frac{\partial f}{\partial \mathbf{x}(e)} \geq 0$. Hence, $\nabla f(\mathbf{x}) = (\frac{\partial f}{\partial \mathbf{x}(1)}, \ldots, \frac{\partial f}{\partial \mathbf{x}(n)})$ is a nonnegative vector.*
2. *$f$ is concave along any direction $\mathbf{d} \geq \mathbf{0}$.*

### 2.2. $k$-submodular Functions

Given a natural number $k \geq 1$, a function $F\colon (k+1)^E \to \mathbb{R}_+$ defined on $k$-tuples of pairwise disjoint subsets of $E$ is called *$k$-submodular* if for all $k$-tuples $S = (S_1, \ldots, S_k)$ and $T = (T_1, \ldots, T_k)$ of pairwise disjoint subsets of $E$,

$$F(S) + F(T) \geq F(S \sqcap T) + F(S \sqcup T),$$

where we define

$$S \sqcap T = (S_1 \cap T_1, \ldots, S_k \cap T_k),$$

$$S \sqcup T = \left( (S_1 \cup T_1) \setminus \left( \bigcup_{i \neq 1} S_i \cup T_i \right), \ldots, (S_k \cup T_k) \setminus \left( \bigcup_{i \neq k} S_i \cup T_i \right) \right).$$

### 2.3. Matroids and Matroid Polytopes

A pair $\mathcal{M} = (E, \mathcal{I})$ of a set $E$ and $\mathcal{I} \subseteq 2^E$ is called a *matroid* if 1) $\emptyset \in \mathcal{I}$, 2) $A \in \mathcal{I}$ for any $A \subseteq B \in \mathcal{I}$, and 3) for any $A, B \in \mathcal{I}$ with $|A| < |B|$, there exists $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$. We call a set in $\mathcal{I}$ an *independent set*. The *rank function* $r_{\mathcal{M}}\colon 2^E \to \mathbb{Z}_+$ of $\mathcal{M}$ is

$$r_{\mathcal{M}}(S) = \max\{|I| : I \subseteq S, I \in \mathcal{I}\}.$$

An independent set $S \in \mathcal{I}$ is called a *base* if $r_{\mathcal{M}}(S) = r_{\mathcal{M}}(E)$. We denote the set of all bases by $\mathcal{B}$ and rank of $\mathcal{M}$ by $r(\mathcal{M})$. The *matroid polytope* $\mathcal{P}(\mathcal{M}) \subseteq \mathbb{R}^E$ of $\mathcal{M}$ is $\mathcal{P}(\mathcal{M}) = \operatorname{conv}\{\mathbf{1}_I : I \in \mathcal{I}\}$, where conv denotes the convex hull. Or equivalently (Edmonds, 2001),

$$\mathcal{P}(\mathcal{M}) = \{\mathbf{x} \geq \mathbf{0} : \mathbf{x}(S) \leq r_{\mathcal{M}}(S)\ \forall S \subseteq E\}.$$

Note that the matroid polytope is *down-monotone*, that is, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^E$ with $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y}$ and $\mathbf{y} \in \mathcal{P}(\mathcal{M})$ then $\mathbf{x} \in \mathcal{P}(\mathcal{M})$.

**Definition 2.2** ($\rho$-covering). *Let $K \subseteq \mathbb{R}^E$ be a set. For $\rho > 0$, a set $C \subseteq K$ of points is called a $\rho$-*covering *of $K$ if for any $\mathbf{x} \in K$, there exists $\mathbf{y} \in C$ such that $\|\mathbf{x} - \mathbf{y}\| \leq \rho$.*

**Theorem 2.3** (Theorem 5.5 of Yoshida (2019), paraphrased). *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid. For every $\epsilon > 0$, we can construct an $\epsilon B$-cover $C$ of $\mathcal{P}(\mathcal{M})$ of size $|E|^{O(1/\epsilon^2)}$ in $|E|^{O(1/\epsilon^2)}$ time, where $B$ is the maximum $\ell_2$-norm of a point in $\mathcal{P}(\mathcal{M})$.*

### 2.4. Differential Privacy

The definition of differential privacy relies on the notion of neighboring datasets. Recall that two datasets are *neighboring* if they differ in a single record. When two datasets $D, D'$ are neighboring, we write $D \sim D'$.

**Definition 2.4** (Dwork et al. (2006)). *For $\epsilon, \delta \in \mathbb{R}_+$, we say that a randomized computation $M$ is $(\epsilon, \delta)$-*differentially private *if for any neighboring datasets $D \sim D'$, and for any set of outcomes $S \subseteq \operatorname{range}(M)$,*

$$\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.$$

*When $\delta = 0$, we say $M$ is $\epsilon$-*differentially private.

In our case, a dataset $D$ consists of *private* submodular functions $F_1, \ldots, F_n\colon 2^E \to [0,1]$. Two datasets $D$ and $D'$ are neighboring if all but one submodular function in those datasets are equal. The submodular function $F_D$ depends on the dataset $D$ in different ways, for example $F_D(S) = \sum_{i=1}^{n} F_i(S)/n$ (CPP problem), or much more complicated ways than averaging functions associated to each individual.

Differentially private algorithms must be calibrated to the sensitivity of the function of interest with respect to small changes in the input dataset, defined formally as follows.

**Definition 2.5.** *The sensitivity of a function $F_D\colon X \to Y$, parameterized by a dataset $D$, is defined as $\max_{D' : D' \sim D} \max_{x \in X} |F_D(x) - F_{D'}(x)|$. A function with sensitivity $\Delta$ is called $\Delta$-sensitive.*

#### 2.4.1. Composition of Differential Privacy

Let $\{(\epsilon_i, \delta_i)\}_{i=1}^k$ be a sequence of privacy parameters and let $M^*$ be a mechanism that behaves as follows on an input $D$. In each of rounds $i = 1, \ldots, k$, the algorithm $M^*$ selects an $(\epsilon_i, \delta_i)$-differentially private algorithm $M_i$ possibly depending on the previous outcomes $M_1(D), \ldots, M_i(D)$ (but not directly on the sensitive dataset $D$ itself), and releases $M_i(D)$. The output of $M^*$ is informally referred as the *$k$-fold adaptive composition* of $(\epsilon_i, \delta_i)$-differentially private algorithms. For a formal treatment of adaptive composition, see Dwork & Roth (2014); Dwork et al. (2010). We have the following guarantee on the differential privacy of the composite algorithm.

**Theorem 2.6.** *(Bun & Steinke, 2016; Dwork & Lei, 2009; Dwork et al., 2010) The $k$-fold adaptive composition of $k$ $(\epsilon_i, \delta_i)$-differentially private algorithms, with $\epsilon_i \leq \epsilon_0$ and $\delta_i \leq \delta_0$ for every $1 \leq i \leq k$, satisfies $(\epsilon, \delta)$-differential privacy where*

- *$\epsilon = k\epsilon_0$ and $\delta = k\delta_0$ (the basic composition), or*
- *$\epsilon = \frac{1}{2}k\epsilon_0^2 + \sqrt{2 \ln 1/\delta'}\epsilon_0$ and $\delta = \delta' + k\delta$ for any $\delta' > 0$ (the advanced composition).*

**Algorithm 1** Differentially Private Continuous Greedy

**Input:** Submodular function $F_D\colon 2^E \to [0,1]$, dataset $D$, matroid $\mathcal{M} = (E, \mathcal{I})$, and $\epsilon > 0$ and $\rho \geq 0$.
Let $C_\rho$ be a $\rho$-covering of $\mathcal{P}(\mathcal{M})$, and $f_D$ be the multilinear extension of $F_D$.
$\mathbf{x}_0 \leftarrow \mathbf{0}$, $\epsilon' \leftarrow \frac{\epsilon}{2\Delta}$.
$\alpha \leftarrow \frac{1}{T}$, where $T = r(\mathcal{M})$.
**for** $t = 1$ to $T$ **do**
  Sample $\mathbf{y} \in C_\rho$ with probability proportional to $\exp\big(\epsilon' \langle \mathbf{y}, \nabla f_D(\mathbf{x}_{t-1})\rangle\big)$.
  Let $\mathbf{y}_{t-1}$ be the sampled vector.
  $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \alpha \mathbf{y}_{t-1}$.
**end for**
**Output:** $\mathbf{x}_T$

#### 2.4.2. Exponential Mechanism

One particularly general tool that we will use is the *exponential mechanism* of McSherry & Talwar (2007). The exponential mechanism is defined in terms of a *quality function* $q_D\colon \mathcal{R} \to \mathbb{R}$, which is parameterized by a dataset $D$ and maps a candidate result $R \in \mathcal{R}$ to a real-valued score.

**Definition 2.7** (McSherry & Talwar (2007))**.** *Let $\epsilon, \Delta > 0$ and let $q_D\colon \mathcal{R} \to \mathbb{R}$ be a quality score. Then, the* exponential mechanism $EM(\epsilon, \Delta, q_D)$ *outputs $R \in \mathcal{R}$ with probability proportional to* $\exp\left(\frac{\epsilon}{2\Delta} \cdot q_D(R)\right)$.

**Theorem 2.8** (McSherry & Talwar (2007))**.** *Suppose that the quality score $q_D\colon \mathcal{R} \to \mathbb{R}$ is $\Delta$-sensitive. Then, $EM(\epsilon, \Delta, q_D)$ is $\epsilon$-differentially private, and for every $\beta \in (0,1)$ outputs $R \in \mathcal{R}$ with*

$$\Pr\left[q_D(R) \geq \max_{R' \in \mathcal{R}} q_D(R') - \frac{2\Delta}{\epsilon} \ln\left(\frac{|\mathcal{R}|}{\beta}\right)\right] \geq 1 - \beta.$$

## 3. Differentially Private Continuous Greedy Algorithm

In this section we prove Theorem 1.1. Throughout this section, we fix (private) monotone submodular functions $F_1, \ldots, F_n\colon 2^E \to [0,1]$, $\epsilon, \delta > 0$, and a matroid $M = (E, \mathcal{I})$. Let $\mathbf{x}^* \in \mathcal{P}(\mathcal{M})$ be a maximizer of $f_D$. We drop the subscript $D$ when it is clear from the context. Our algorithm (Algorithm 1) is a modification of the continuous greedy algorithm (Călinescu et al., 2011).

### 3.1. Approximation Guarantee

**Lemma 3.1.** *For every $\mathbf{x}, \mathbf{v} \in [0,1]^E$ with $\|\mathbf{v}\|_2 \leq \rho$ and $\mathbf{x} + \mathbf{v} \in [0,1]^E$, we have $|f(\mathbf{x}) - f(\mathbf{x} + \mathbf{v})| \leq 4\sqrt[4]{|E|}\sqrt{\rho}$.*

**Lemma 3.2.** *Suppose $\mathbf{y} \in [0,1]^E$ satisfies $\|\mathbf{y} - \mathbf{x}^*\|_2 \leq \rho$. Then for any $\mathbf{x} \in [0,1]^E$, we have $\langle \mathbf{y}, \nabla f(\mathbf{x})\rangle \geq f(\mathbf{x}^*) - f(\mathbf{x}) - C_{3.2}\sqrt{\rho}$ for some constant $C_{3.2} > 0$.*

*Proof.* First, we show

$$\langle \mathbf{y}, \nabla f(\mathbf{x})\rangle \geq f(\mathbf{y}) - f(\mathbf{x}).$$

Let us consider a direction $\mathbf{d} \in [0,1]^E$ such that $\mathbf{d}(e) = \max\{\mathbf{y}(e) - \mathbf{x}(e), 0\}$ for every $e \in E$. Then, we have

$$\begin{aligned}\langle \mathbf{y}, \nabla f(\mathbf{x})\rangle &\geq \langle \mathbf{d}, \nabla f(\mathbf{x})\rangle \\ &\geq f(\mathbf{x} + \mathbf{d}) - f(\mathbf{x}) \\ &\geq f(\mathbf{y}) - f(\mathbf{x}),\end{aligned}$$

where the first inequality follows from $\mathbf{y} \geq \mathbf{d}$ and $\nabla f(\mathbf{x}) \geq 0$, the second inequality follows from the concavity of $f$ along $\mathbf{d}$, and the third inequality follows from $\mathbf{x} + \mathbf{d} \geq \mathbf{y}$ and the monotonicity of $f$. By Lemma 3.1, we have

$$f(\mathbf{y}) \geq f(\mathbf{x}^*) - 4\sqrt[4]{|E|}\sqrt{\rho},$$

which yields the desired result with $C_{3.2} = 4\sqrt[4]{|E|}$. □

**Theorem 3.3.** *Suppose $F_D$ is $\Delta$-sensitive and $C_\rho$ is a $\rho$-covering of $\mathcal{P}(\mathcal{M})$. Then Algorithm 1, with high probability, returns $\mathbf{x}_T \in \mathcal{P}(\mathcal{M})$ such that*

$$f_D(\mathbf{x}_T) \geq \left(1 - \frac{1}{\mathrm{e}}\right)\mathrm{OPT} - O\left(C_{3.2}\rho + \frac{\Delta r(\mathcal{M}) \ln |E|}{\epsilon\rho^2}\right)$$

*Moreover, the algorithm evaluates $f_D$ at most $O\left(r(\mathcal{M}) \cdot |C_\rho|\right)$ times.*

*Proof.* Clearly Algorithm 1 evaluates $f$ at most $O\left(r(\mathcal{M})|C_\rho|\right)$ times. Observe that the algorithm forms a convex combination of $T$ vertices of the polytope $\mathcal{P}(\mathcal{M})$, each with weight $\alpha$ hence $\mathbf{x}_T \in \mathcal{P}(\mathcal{M})$. In what follows, we focus on the quality of the output of the algorithm. Suppose $\mathbf{y}' \in C_\rho$ with $\|\mathbf{y}' - \mathbf{x}^*\|_2 \leq \rho$. By Theorem 2.8, with probability at least $1 - \frac{1}{|E|^2}$, we have

$$\begin{aligned}\langle \mathbf{y}_t, \nabla f(\mathbf{x}_t)\rangle &\geq \operatorname*{argmax}_{\mathbf{y} \in C_\rho} \langle \mathbf{y}, \nabla f(\mathbf{x}_t)\rangle - \frac{2\Delta}{\epsilon}\ln(|E|^2|C_\rho|) \\ &\geq \langle \mathbf{y}', \nabla f(\mathbf{x}_t)\rangle - \frac{2\Delta}{\epsilon}\ln(|E|^2|C_\rho|) \\ &\overset{\text{By Lemma 3.2}}{\geq} f(\mathbf{x}^*) - f(\mathbf{x}_t) - C_{3.2}\sqrt{\rho} - \frac{2\Delta}{\epsilon}\ln(|E|^2|C_\rho|)\end{aligned}$$

By a union bound, with probability at least $1 - \frac{1}{poly(|E|)}$, the above inequality holds for every $t$. In what follows, we assume this has happened. Further, let us assume that $t$ is a continuous variable in $[0, T]$. We remark that discretization of $t$ in our algorithm introduces error into the approximation guarantee. However, this can be handled by sufficiently large $T$, say, $r(\mathcal{M})$ as in Algorithm 1, and small step size $\alpha$ (Călinescu et al., 2011). In what follows $t$ is assumed to be continuous and we write $\frac{d\mathbf{x}_t}{dt} = \alpha \mathbf{y}_t$, hence

$$\frac{df(\mathbf{x}_t)}{dt} = \sum_e \frac{\partial f(\mathbf{x}_t(e))}{\partial \mathbf{x}_t(e)} \frac{d\mathbf{x}_t(e)}{dt}$$

$$= \nabla f(\mathbf{x}_t) \cdot \frac{d\mathbf{x}_t}{dt} = \alpha \langle \mathbf{y}_t, \nabla f(\mathbf{x}_t) \rangle$$
$$\geq \alpha \left( f(\mathbf{x}^*) - f(\mathbf{x}_t) - C_{3.2}\sqrt{\rho} - \frac{2\Delta}{\epsilon} \ln(|E|^2 |C_\rho|) \right),$$

where the first equality follows from the chain rule. Let $\beta = f(\mathbf{x}^*) - C_{3.2}\sqrt{\rho} - \frac{2\Delta}{\epsilon} \ln(|E|^2|C_\rho|)$. Solving the following differential equation $\frac{df(\mathbf{x}_t)}{dt} = \alpha(\beta - f(\mathbf{x}_t))$ with $f(\mathbf{x}_0) = 0$ gives us $f(\mathbf{x}_t) = \beta(1 - \mathrm{e}^{-\alpha t})$. For $\alpha = \frac{1}{T}, t = T$ we obtain

$$f(\mathbf{x}_T) = \beta(1 - \mathrm{e}^{-1})$$
$$= \left(1 - \frac{1}{\mathrm{e}}\right) f(\mathbf{x}^*) - O\left( C_{3.2}\sqrt{\rho} + \frac{2\Delta}{\epsilon} \ln(|E|^2|C_\rho|) \right)$$
$$\geq \left(1 - \frac{1}{\mathrm{e}}\right) f(\mathbf{x}^*) - O\left( C_{3.2}\sqrt{\rho} + \frac{\Delta r(\mathcal{M}) \ln |E|}{\epsilon \rho^2} \right) \quad \square$$

**Remark 3.1.** As already pointed out in the proof of Theorem 3.3, the discretization of $t$ introduces error into the approximation guarantee yielding $(1 - 1/\mathrm{e} - 1/\mathrm{poly}(|E|))\mathrm{OPT}$. However, this can be shaved off to $(1 - 1/\mathrm{e})\mathrm{OPT}$ by sufficiently large $T$ (Călinescu et al., 2011). Moreover, evaluating $f$ (even approximately) is expensive. To achieve the nearly optimal approximation guarantees, the evaluation error needs to be very small and in a lot of cases, the error needs to be $O(1/|E|)$ times the function value. As a result, a single evaluation of the multilinear extension $f$ requires $\Omega(|E|)$ evaluations of $F$ (see Ene & Nguyen (2019) for recent improvement). Therefore, our algorithm requires $O(r(\mathcal{M})|E||C_\rho|)$ evaluation of $F$.

**Remark 3.2.** From a fractional solution $\mathbf{x}^*$, we can obtain an integral solution $\mathbf{s} \in \{0,1\}^E$ such that $f(\mathbf{s}) \geq f(\mathbf{x}^*)$. Such an integer solution corresponds to a vertex of $\mathcal{P}(\mathcal{M})$ and hence a discrete solution $S \in \mathcal{I}$. This can be done using the so-called *swap rounding* (Chekuri et al., 2010).

### 3.2. Privacy Analysis

**Theorem 3.4.** *Algorithm 1 preserves $O(\epsilon r(\mathcal{M})^2)$-differential privacy.*

*Proof.* Let $D$ and $D'$ be two neighboring datasets and $F_D, F_{D'}$ be their associated functions. For a fixed $\mathbf{y}_t \in C_\rho$, we consider the relative probability of Algorithm 1 (denoted by $M$) choosing $\mathbf{y}_t$ at time step $t$ given multilinear extensions of $F_D$ and $F_{D'}$. Let $M_t(f_D \mid \mathbf{x}_t)$ denote the output of $M$ at time step $t$ given dataset $D$ and point $\mathbf{x}_t$. Similarly, $M_t(f_{D'} \mid \mathbf{x}_t)$ denotes the output of $M$ at time step $t$ given dataset $D'$ and point $\mathbf{x}_t$. Further, write $d_{\mathbf{y}} = \langle \mathbf{y}, \nabla f_D(\mathbf{x}_t) \rangle$ and $d'_{\mathbf{y}} = \langle \mathbf{y}, \nabla f_{D'}(\mathbf{x}_t) \rangle$. We have

$$\frac{\Pr[M_t(f_D \mid \mathbf{x}_t) = \mathbf{y}_t]}{\Pr[M_t(f_{D'} \mid \mathbf{x}_t) = \mathbf{y}_t]} = \frac{\exp(\epsilon' \cdot d_{\mathbf{y}_t})}{\exp(\epsilon' \cdot d'_{\mathbf{y}_t})} \cdot \frac{\sum_{\mathbf{y} \in C_\rho} \exp(\epsilon' \cdot d'_{\mathbf{y}})}{\sum_{\mathbf{y} \in C_\rho} \exp(\epsilon' \cdot d_{\mathbf{y}})}.$$

For the first factor, we have

$$\frac{\exp(\epsilon' \cdot d_{\mathbf{y}_t})}{\exp(\epsilon' \cdot d'_{\mathbf{y}_t})} = \exp\big(\epsilon'(d_{\mathbf{y}_t} - d'_{\mathbf{y}_t})\big)$$
$$= \exp\big(\epsilon'(\langle \mathbf{y}_t, \nabla f_D(\mathbf{x}_t) - \nabla f_{D'}(\mathbf{x}_t) \rangle)\big)$$
$$\leq \exp\big(\epsilon' \|\mathbf{y}_t\|_1 \|\nabla f_D(\mathbf{x}_t) - \nabla f_{D'}(\mathbf{x}_t)\|_\infty\big)$$
$$= \exp\left( \epsilon' \sum_{e \in E} \mathbf{y}_t(e) \cdot \left( \max_{e \in E} \mathop{\mathbb{E}}_{R \sim \mathbf{x}_t} \Big[ F_D(R \cup \{e\}) - F_D(R) - F_{D'}(R \cup \{e\}) + F_{D'}(R) \Big] \right) \right)$$
$$\leq \exp(O(\epsilon' \cdot r(\mathcal{M}) \cdot 2\Delta)) = \exp(O(\epsilon \cdot r(\mathcal{M})))$$

Note that the last inequality holds since $\mathbf{y}_t$ is a member of the matroid polytope $\mathcal{P}(\mathcal{M})$ and by definition we have $\sum_{e \in E} \mathbf{y}_t(e) \leq r_{\mathcal{M}}(E) = r(\mathcal{M})$. Moreover, recall that $F_D$ is $\Delta$-sensitive.

For the second factor, let us write $\beta_{\mathbf{y}} = d'_{\mathbf{y}} - d_{\mathbf{y}}$ to be the *deficit* of the probabilities of choosing direction $\mathbf{y}$ in instances $f_{D'}$ and $f_D$. Then, we have

$$\frac{\sum_{\mathbf{y} \in C_\rho} \exp(\epsilon' \cdot d'_{\mathbf{y}})}{\sum_{\mathbf{y} \in C_\rho} \exp(\epsilon' \cdot d_{\mathbf{y}})} = \frac{\sum_{\mathbf{y} \in C_\rho} \exp(\epsilon' \cdot \beta_{\mathbf{y}}) \exp(\epsilon' \cdot d_{\mathbf{y}})}{\sum_{\mathbf{y} \in C_\rho} \exp(\epsilon' \cdot d_{\mathbf{y}})}$$
$$= \mathbb{E}_{\mathbf{y}}[\exp(\epsilon' \cdot \beta_{\mathbf{y}})] \leq \exp\big(O(\epsilon' \cdot r(\mathcal{M}) \cdot 2\Delta)\big)$$
$$= \exp\big(O(\epsilon \cdot r(\mathcal{M}))\big).$$

The expectation is taken over the probability distribution over $\mathbf{y}$ selected at time $t$ in instance with input $D$. Recall that we choose $\mathbf{y}$ with probability proportional to $\exp(\epsilon' d_{\mathbf{y}})$. By a union bound, Algorithm 1 preserves $O(\epsilon T r(\mathcal{M})) \leq O(\epsilon r(\mathcal{M})^2)$-differential privacy. To obtain an integral solution from a fractional solution, we use swap rounding technique (see Remark 3.2) which does not depend on the input function and hence preserves the privacy. $\square$

Note that the privacy factor in the work of Mitrovic et al. (2017) is $O(\epsilon r(\mathcal{M}))$. However, our privacy factor is $O(\epsilon r(\mathcal{M})^2)$, this is because we deal with the multilinear extension of a submodular function rather than the function itself (which is different from the previous works).

**Theorem 3.5** (Formal version of Theorem 1.1)**.** *Suppose $F_D$ is $\Delta$-sensitive and Algorithm 1 is instantiated with $\rho = \frac{\epsilon}{|E|^{1/2}}$. Then Algorithm 1 is $(\epsilon r(\mathcal{M})^2)$-differentially private and, with high probability, returns $S \in \mathcal{I}$ with quality at least*

$$F_D(S) \geq \left(1 - \frac{1}{\mathrm{e}}\right) \mathrm{OPT} - O\left( \sqrt{\epsilon} + \frac{\Delta r(\mathcal{M}) |E| \ln |E|}{\epsilon^3} \right)$$

**Example 3.1** (Maximum Coverage)**.** Let $G = (U, V, E)$ be a bipartite graph, and $B$ be a budget constraint. In Maximum Coverage problem, the goal is to find a set $S$ of $B$ vertices in

$U$ so that the number of vertices in $V$ incident to some vertex in $S$ is maximized. The edges incident to a vertex $v \in V$ are private information about $v$. If we instantiate Theorem 3.5 on this problem, the privacy factor is $\epsilon B^2$ and the additive error is $O(\Delta B|U|\ln(|U|)/\epsilon^3)$, where $\Delta$ is the maximum degree of a vertex in $V$. To have a meaningful privacy bound, we set $\epsilon \ll 1/B^2$, and the additive error becomes $\Delta B^7|U|\ln(|U|)$. However, OPT could be $\Omega(|V|)$, which is much larger than the additive error when $|V| \gg |U|$. Indeed, by optimizing $\rho$, we can improve the additive error to $O(\Delta B^3|U|\ln(|U|))$, which will be more practical.

## 4. Improving the Query Complexity

In this section, we improve the number of evaluations of $F$ from $O(r(\mathcal{M})|E|^{1+(\frac{r(\mathcal{M})}{\epsilon})^2})$ to $O(r(\mathcal{M})|E|^2\ln\frac{|E|}{\epsilon})$. In Algorithm 1, in order to choose a point with probability proportional to $\exp(\langle \mathbf{y}, \nabla f(\mathbf{x})\rangle)$, it requires to compute $Z = \sum_{\mathbf{z}\in C_\rho} \exp(\langle \mathbf{z}, \nabla f(\mathbf{x})\rangle)$. This summation needs evaluating $(\langle \mathbf{z}, \nabla f(\mathbf{x})\rangle)$ for all $\mathbf{z}$ in $C_\rho$. One way of improving the query complexity of this step is as follows. Partition $C_\rho$ into a number of layers such that points in each layer are almost the same in terms of the inner product $\langle \cdot, \nabla f(\mathbf{x})\rangle$. Now, instead of choosing a point in $C_\rho$, we carefully select a layer with some probability (i.e., proportional to its size and *quality* of points in it) and then choose a point from that layer uniformly at random. Of course, to estimate the size of each layer, we need to sample a sufficiently large number of points from $C_\rho$.

**Definition 4.1** (layer)**.** *For a point $\mathbf{x} \in C_\rho$ and $\mu > 0$, let the $i$-th layer to be $\mathcal{L}^{\mathbf{x}}_{\mu,i} = \{\mathbf{z} \in C_\rho \mid (1+\mu)^{i-1} \le \exp\big(\langle \mathbf{z}, \nabla f(\mathbf{x})\rangle\big) < (1+\mu)^i\}$, for $1 \le i \le k$, where*

$$k = \left\lceil \log_{1+\mu}\left(\frac{\max_{\mathbf{y}\in C_\rho}\exp\big(\langle \mathbf{y}, \nabla f(\mathbf{x})\rangle\big)}{\min_{\mathbf{y}\in C_\rho}\exp\big(\langle \mathbf{y}, \nabla f(\mathbf{x})\rangle\big)}\right)\right\rceil.$$

For a layer $\mathcal{L}^{\mathbf{x}}_{\mu,i}$ let $|\mathcal{L}^{\mathbf{x}}_{\mu,i}|$ denote the number of points in it, and define $\tilde{Z} \in \mathbb{R}$ and $\tilde{Z}_i \in \mathbb{R}$ for each $i \in [k]$ as follows:

$$\tilde{Z} = \sum_{i\in[k]} |\mathcal{L}^{\mathbf{x}}_{\mu,i}|(1+\mu)^{i-1}\,,\ \tilde{Z}_i = |\mathcal{L}^{\mathbf{x}}_{\mu,i}|(1+\mu)^{i-1}.$$

Then, a layer $\mathcal{L}^{\mathbf{x}}_{\mu,i}$ is chosen with probability $\frac{\tilde{Z}_i}{\tilde{Z}}$. Note that we do not want to spend time computing the exact value of $|\mathcal{L}^{\mathbf{x}}_{\mu,i}|$ for every layer, instead, we are interested in efficiently estimating these values. By Hoeffding's inequality (Hoeffding, 1963), to estimate $|\mathcal{L}^{\mathbf{x}}_{\mu,i}|/|C_\rho|$ with additive error of $\lambda$ with probability at least $1-\theta$, it suffices to sample $\Theta(\ln(1/\theta)/\lambda^2)$ points from $C_\rho$. Hence, by a union bound, if we want to estimate $|\mathcal{L}^{\mathbf{x}}_{\mu,i}|/|C_\rho|$ with additive error of $\lambda$ for all $i = 1,\dots,k$ with probability at least $1-\theta$, it suffices to sample $\Theta(\ln(k/\theta)/\lambda^2)$ points from $C_\rho$.

**Algorithm 2** Improved Differentially Private Continuous Greedy Algorithm

**Input:** Submodular function $F_D\colon 2^E \to [0,1]$, dataset $D$, a matroid $\mathcal{M} = (E, \mathcal{I})$, and $\epsilon, \mu, \rho, \lambda, \theta > 0$.
Let $C_\rho$ be a $\rho$-covering of $\mathcal{P}(\mathcal{M})$, and $f_D$ be the multilinear extension of $F_D$.
$\mathbf{x}(0) \leftarrow \mathbf{0}$, $\epsilon' \leftarrow \frac{\epsilon}{2\Delta}$.
**for** $t = 1$ to $T = r(\mathcal{M})$ **do**
  $C'_\rho \leftarrow$ Sample $\Theta(\ln(k/\theta)/\lambda^2)$ points from $C_\rho$ uniformly at random.
  Define $\mathcal{L}^{\mathbf{x}_{t-1}}_{\mu,i}$ as in Definition 4.1, and estimate each $|\mathcal{L}^{\mathbf{x}_{t-1}}_{\mu,i}|$ using $C'_\rho$.
  Let $\tilde{L}^{\mathbf{x}_{t-1}}_{\mu,i}$ denote the estimated value.
  Set $\tilde{Z}_i \leftarrow \tilde{L}^{\mathbf{x}_{t-1}}_{\mu,i}(1+\mu)^{\epsilon'(i-1)}$ and $\tilde{Z} \leftarrow \sum_{i\in[k]} \tilde{L}^{\mathbf{x}_{t-1}}_{\mu,i}(1+\mu)^{\epsilon'(i-1)}$
  Let $\mathcal{L}$ be the chosen layer $\mathcal{L}^{\mathbf{x}_{t-1}}_{\mu,i}$ with probability proportional to $\frac{\tilde{Z}_i}{\tilde{Z}}$.
  Let $\mathbf{y}_{t-1}$ be a point sampled uniformly at random from $\mathcal{L}$.
  $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \alpha\mathbf{y}_{t-1}$.
**end for**
**Outout:** $\mathbf{x}_T$

**Corollary 4.2.** *Let $C_\rho$ be a $\rho$-covering of $\mathcal{P}(\mathcal{M})$ and $\mathbf{x}_t$ be a point in $\mathcal{P}(\mathcal{M})$. Algorithm 2 estimates $|\mathcal{L}^{\mathbf{x}_t}_{\mu,i}|/|C_\rho|$ with an additive error $\lambda_{4.2}$ with probability at least $1-\theta_{4.2}$.*

**Lemma 4.3** (Analogous to Theorem 2.8)**.** *At each time step $t$, Algorithm 2 returns $\mathbf{y}_{t-1}$ such that for every $\beta \in (0,1)$ and $\xi = \ln\left(\frac{|C_\rho|(1+k\lambda|C_\rho|)(1+\mu)^{\epsilon'}}{\beta}\right)$ we have*

$$\Pr\left[\langle \mathbf{y}_{t-1}, \nabla f(\mathbf{x}_{t-1})\rangle \ge \max_{\mathbf{z}\in C_\rho}\langle \mathbf{z}, \nabla f(\mathbf{x}_{t-1})\rangle - \frac{2\Delta}{\epsilon}\xi\right] \ge 1-\beta.$$

**Theorem 4.4.** *Suppose $F_D$ is $\Delta$-sensitive and $C_\rho$ is a $\rho$-covering of $\mathcal{P}(\mathcal{M})$. Then Algorithm 2, with high probability (depending on $\theta_{4.2}$), returns $\mathbf{x}_T \in \mathcal{P}(\mathcal{M})$ such that*

$$f(\mathbf{x}_T) \ge \left(1-\frac{1}{e}\right)\mathrm{OPT} - O\Bigg(C_{3.2}\sqrt{\rho} + \ln(1+\mu) + \left(\frac{\Delta r(\mathcal{M})}{\epsilon\rho^2}\right)(\ln|E| + \ln(k\lambda_{4.2}))\Bigg)$$

**Theorem 4.5.** *Algorithm 2 preserves $O\left(\epsilon r(\mathcal{M})^2\right)$-differential privacy.*

**Theorem 4.6** (Formal version of Theorem 1.2)**.** *Suppose $F_D$ is $\Delta$-sensitive and Algorithm 2 is instantiated with $\rho = \frac{\epsilon}{|E|^{1/2}}, \mu = \mathrm{e}^\epsilon, \lambda_{4.2} = 1/\sqrt{|E|}, \theta_{4.2} = 1/|E|^2$. Then*

*Algorithm 2 is $(\epsilon r(\mathcal{M})^2)$-differentially private and, with high probability, returns $S \in \mathcal{I}$ with quality at least*

$$F_D(S) \geq \left(1 - \frac{1}{e}\right) \mathrm{OPT} - O\left(\sqrt{\epsilon} + \frac{\Delta r(\mathcal{M})|E| \ln(\frac{|E|}{\epsilon})}{\epsilon^3}\right).$$

*Moreover, it evaluates $F_D$ at most $O(r(\mathcal{M})|E|^2 \ln(\frac{|E|}{\epsilon}))$ times.*

## 5. $k$-Submodular Function Maximization

In this section, we study a natural generalization of submodular functions, namely $k$-submodular functions. Associate $(S_1, \ldots, S_k) \in (k+1)^E$ with $\mathbf{s} \in \{0, 1, \ldots, k\}^E$ by $S_i = \{e \in E \mid \mathbf{s}(e) = i\}$ for $i \in [k]$ and define the *support* of $\mathbf{s}$ as $\mathrm{supp}(\mathbf{s}) = \{e \in E \mid \mathbf{s}(e) \neq 0\}$. Let $\preceq$ be a partial ordering on $(k+1)^E$ such that, for $\mathbf{s} = (S_1, \ldots, S_k)$ and $\mathbf{t} = (T_1, \ldots, T_k)$ in $(k+1)^E$, $\mathbf{s} \preceq \mathbf{t}$ if $S_i \subseteq T_i$ for every $i \in [k]$. We say that a function $F \colon (k+1)^E \to \mathbb{R}_+$ is *monotone* if $F(\mathbf{s}) \leq F(\mathbf{t})$ holds for every $\mathbf{s} \preceq \mathbf{t}$. Define the *marginal gain* of adding $e \notin \bigcup_{\ell \in [k]} S_\ell$ to the $i$-th set of $\mathbf{s} \in (k+1)^E$ to be

$$\Delta_{e,i} F(\mathbf{s}) = F(S_1, \ldots, S_{i-1}, S_i \cup \{e\}, S_{i+1}, \ldots, S_k) - F(S_1, \ldots, S_k).$$

The monotonicity of $F$ is equivalent to $\Delta_{e,i} F(\mathbf{s}) \geq 0$ for any $\mathbf{s} = (S_1, \ldots, S_k)$ and $e \notin \bigcup_{\ell \in [k]} S_\ell$ and $i \in [k]$.

Our goal is maximizing a monotone $k$-submodular function under matroid constraints. That is, given a monotone $k$-submodular function $F_D \colon (k+1)^E \to \mathbb{R}_+$ and a matroid $\mathcal{M} = (E, \mathcal{I})$, we want to solve the following problem.

$$\max_{\mathbf{x} \in (k+1)^E} F_D(\mathbf{x}) \quad \text{subject to} \quad \bigcup_{i \in [k]} X_i \in \mathcal{I}$$

The following are known due to Sakaue (2017). They may have appeared in other literature that we are not aware of.

**Lemma 5.1** (Sakaue (2017)). *For any maximal optimal solution $\mathbf{o}$ we have $|\mathrm{supp}(\mathbf{o})| = r(\mathcal{M})$.*

**Lemma 5.2** (Sakaue (2017)). *Suppose $A \in \mathcal{I}$ and $B \in \mathcal{B}$ (recall $\mathcal{B}$ denotes the set of bases) satisfy $A \subseteq B$. Then, for any $e \notin A$ satisfying $A \cup \{e\} \in \mathcal{I}$, there exists $e' \in B \setminus A$ such that $B \setminus \{e'\} \cup \{e\} \in \mathcal{B}$.*

Having Lemma 5.1, our algorithm runs in $r(\mathcal{M})$ iterations and at each iteration chooses an element $e$ with probability proportional to $\exp(\epsilon' \Delta_{e,i} F_D(\mathbf{x}))$ and adds $e$ to $\mathrm{supp}(\mathbf{x})$. The analysis for the approximation guarantee is similar to the ones in Iwata et al. (2016); Ohsaka & Yoshida (2015); Sakaue (2017); Ward & Zivny (2014) and relies on Theorem 2.8.

**Algorithm 3** Differentially private $k$-submodular maximization with a matroid constraint

**Input:** monotone $k$-submodular functions $F_D \colon (k+1)^E \to [0,1]$, a matroid $\mathcal{M} = (E, \mathcal{I})$, and $\epsilon > 0$.
$\mathbf{x} \leftarrow \mathbf{0}$, $\epsilon' \leftarrow \frac{\epsilon}{2\Delta}$
**for** $t = 1$ to $r(\mathcal{M})$ **do**
  Let $\Lambda(\mathbf{x}) = \{e \in E \setminus \mathrm{supp}(\mathbf{x}) \mid \mathrm{supp}(\mathbf{x}) \cup \{e\} \in \mathcal{I}\}$
  Choose $e \in \Lambda(\mathbf{x})$ and $i \in [k]$ with probability proportional to $\exp(\epsilon' \Delta_{e,i} F_D(\mathbf{x}))$.
  $\mathbf{x}(e) \leftarrow i$.
**end for**
**Output:** $\mathbf{x}$

**Theorem 5.3.** *Suppose $F_D$ has sensitivity $\Delta$. Then Algorithm 3, with high probability, returns $\mathbf{x} \in (k+1)^E$ such that $\mathrm{supp}(\mathbf{x}) \in \mathcal{B}$ and $F_D(\mathbf{x}) \geq \frac{1}{2}\mathrm{OPT} - O(\frac{\Delta r(\mathcal{M}) \ln |E|}{\epsilon})$.*

The privacy guarantee follows immediately from the $\epsilon$-differential privacy of the exponential mechanism, together with Theorem 2.6.

**Theorem 5.4.** *Algorithm 3 preserves $O(\epsilon r(\mathcal{M}))$-differential privacy. It also provides $(\frac{1}{2} r(\mathcal{M}) \epsilon^2 + \sqrt{2 \ln 1/\delta'}\epsilon, \delta')$-differential privacy for every $\delta' > 0$.*

Clearly, Algorithm 3 evaluates $F_D$ at most $O(k|E|r(\mathcal{M}))$ times. Next theorem summarizes the results of this section.

**Theorem 5.5.** *Suppose $F_D$ has sensitivity $\Delta$. Then Algorithm 3, with high probability, outputs a solution $\mathbf{x} \in (k+1)^E$ such that $\mathrm{supp}(\mathbf{x})$ is a base of $\mathcal{M}$ and $F_D(\mathbf{x}) \geq \frac{1}{2}\mathrm{OPT} - O(\frac{\Delta r(\mathcal{M}) \ln |E|}{\epsilon})$ by evaluating $F_D$ at most $O(k|E|r(\mathcal{M}))$ times. Moreover, this algorithm preserves $O(r(\mathcal{M})\epsilon)$-differential privacy.*

### 5.1. Improving the Query Complexity

By applying a sampling technique (Mirzasoleiman et al., 2015; Ohsaka & Yoshida, 2015), we improve the number of evaluations of $F$ from $O(k|E|r(\mathcal{M}))$ to $O(k|E| \ln r(\mathcal{M}) \ln \frac{r(\mathcal{M})}{\gamma})$, where $\gamma > 0$ is a failure probability. Hence, even when $r(\mathcal{M})$ is as large as $|E|$, the number of function evaluations is almost linear in $|E|$. The main difference from Algorithm 3 is that we sample a sufficiently large subset $R$ of $E$, and then greedily assign a value only looking at elements in $R$.

**Theorem 5.6.** *Suppose $F_D$ has sensitivity $\Delta$. Then Algorithm 4, with probability at least $1 - \gamma$, outputs a solution with quality at least $\frac{1}{2}\mathrm{OPT} - O\left(\frac{\Delta r(\mathcal{M}) \ln \frac{|E|}{\gamma}}{\epsilon}\right)$ by evaluating $F_D$ at most $O\left(k|E| \ln r(\mathcal{M}) \ln \frac{r(\mathcal{M})}{\gamma}\right)$ times.*

Similar to Theorem 5.4 and using the composition Theorem 2.6, Algorithm 4 preserves $O(\epsilon r(\mathcal{M}))$-differential

**Algorithm 4** Improved differentially private $k$-submodular maximization with a matroid constraint

**Input:** monotone $k$-submodular functions $F_D\colon (k+1)^E \to [0,1]$, a matroid $\mathcal{M} = (E, \mathcal{I})$, $\epsilon > 0$, and a failure probability $\gamma > 0$.
$\mathbf{x} \leftarrow \mathbf{0}$, $\epsilon' \leftarrow \frac{\epsilon}{2\Delta}$
**for** $t = 1$ to $r(\mathcal{M})$ **do**
  $R \leftarrow$ a random subset of size $\min\{\frac{|E|-t+1}{r(\mathcal{M})-t+1} \log \frac{r(\mathcal{M})}{\gamma}, |E|\}$ uniformly sampled from $E \setminus \mathrm{supp}(\mathbf{x})$.
  Choose $e \in R$ with $\mathrm{supp}(\mathbf{x}) \cup \{e\} \in \mathcal{I}$ and $i \in [k]$ with probability proportional to $\exp(\epsilon' \Delta_{e,i} F_D(\mathbf{x}))$.
  $\mathbf{x}(e) \leftarrow i$.
**end for**
**Output:** $\mathbf{x}$

privacy. It also provides $O\left(\frac{1}{2} r(\mathcal{M})\epsilon^2 + \sqrt{2 \ln 1/\delta'}\epsilon, \delta'\right)$-differential privacy for every $\delta' > 0$. In summary, we have

**Theorem 5.7.** *Suppose $F_D$ has sensitivity $\Delta$. Then, with probability at least $1-\gamma$, Algorithm 4 returns a solution $\mathbf{x} \in (k+1)^E$ such that $\mathrm{supp}(\mathbf{x}) \in \mathcal{B}$ and $F_D(\mathbf{x}) \geq \frac{1}{2}\mathrm{OPT} - O\left(\frac{\Delta r(\mathcal{M}) \ln \frac{|E|}{\gamma}}{\epsilon}\right)$ by evaluating $F_D$ at most $O\left(k|E| \ln r(\mathcal{M}) \ln \frac{r(\mathcal{M})}{\gamma}\right)$ times. Moreover, this algorithm preserves $O(\epsilon r(\mathcal{M}))$-differential privacy.*

### 5.2. Motivating Examples

**Example 5.1.** Suppose that we have $m$ ad slots and $k$ ad agencies, and we want to allocate at most $B(\leq m)$ slots to the ad agencies. Each ad agency $i$ has a influence graph $G_i$, which is a bipartite graph $(U, V, E_i)$, where $U$ and $V$ correspond to ad slots and users, respectively, and an edge $uv \in E_i$ indicates that if the ad agency $i$ takes the ad slot $u$ (and put an ad there), the user $v$ will be influenced by the ad. The goal is to maximize the number of influenced people (each person will be counted multiple times if he/she is influenced by multiple ad agencies), based on which we get revenue from the ad agencies. This problem can be modeled as $k$-submodular function maximization under a cardinality constraint (a special case of matroid constraints), and edges incident to a user $v$ in $G_1, \ldots, G_k$ are sensitive data about $v$.

**Example 5.2.** Another example comes from (a variant of) facility location. Suppose that we have a set $E$ of $n$ lands, and we want to provide $k$ resources (e.g., gas and electricity) to all the lands by opening up facilities at some of the lands. For each resource type $i$ and lands $e, e' \in E$, we have a cost $c_i(e, e')$ of sending the resource of type $i$ from $e$ to $e'$. For a set $S \subseteq E$, let $c_i(e, S) = \min_{e' \in S} c_i(e, e')$, which is the cost of sending a resource of type $i$ to e when we open up facilities of type $i$ at lands in $S$. Assume we cannot open two or more facilities in the same land. Then, the goal is to find disjoint sets $S_1, \ldots, S_k$ with $\sum_i |S_i| <= B$ for some fixed B that maximize $\sum_e \sum_i (C - c_i(e, S_i))$, where $C$ is a large number so that the objective function is always non-negative. This problem can be modeled as $k$-submodular function maximization under a cardinality constraint, and the costs $c_i(e, \cdot)$ are sensitive data about $e$.

## 6. Conclusion

We proposed a differentially private algorithm for maximizing monotone submodular functions under matroid constraint. Our algorithm provides the best possible approximation guarantee that matches the approximation guarantee in non-private setting. It also has a competitive number of function evaluations that is significantly faster than the non-private one. We also presented a differentially private algorithm for $k$-submodular maximization under matroid constraint that uses almost liner number of function evaluations and has an asymptotically tight approximation ratio.

## Acknowledgments

A.R. is thankful to Igor Shinkar and Nazanin Mehrasa for useful discussions. We also thank anonymous referees for useful suggestions. A.R. is supported by NSERC. Y.Y. is supported by JSPS KAKENHI Grant Number 18H05291.

## References

Bilbao, J. M., Fernández, J. R., Jiménez, N., and López, J. J. A survey of bicooperative games. In *Pareto Optimality, Game Theory And Equilibria*, pp. 187–216. Springer, 2008.

Bun, M. and Steinke, T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th International Conference on Theory of Cryptography (TCC)*, pp. 635–658, 2016.

Călinescu, G., Chekuri, C., Pál, M., and Vondrák, J. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

Chekuri, C., Vondrák, J., and Zenklusen, R. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 575–584, 2010.

Cohen, D. A., Cooper, M. C., Jeavons, P., and Krokhin, A. A. The complexity of soft constraint satisfaction. *Artif. Intell.*, 170(11):983–1016, 2006.

Dwork, C. and Lei, J. Differential privacy and robust statis-

tics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 371–380, 2009.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Technique (EUROCRYPT)*, pp. 486–503, 2006.

Dwork, C., Rothblum, G. N., and Vadhan, S. P. Boosting and differential privacy. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 51–60, 2010.

Edmonds, J. Matroids and the greedy algorithm. *Math. Program.*, 1(1):127–136, 1971.

Edmonds, J. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization - Eureka, You Shrink!*, pp. 11–26, 2001.

Ene, A. and Nguyen, H. L. Towards nearly-linear time algorithms for submodular maximization with a matroid constraint. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 54:1–54:14, 2019.

Gupta, A., Ligett, K., McSherry, F., Roth, A., and Talwar, K. Differentially private combinatorial optimization. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1106–1125, 2010.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Huber, A. and Kolmogorov, V. Towards minimizing $k$-submodular functions. In *Proceedings of the 2nd International Symposium on Combinatorial Optimization (ISCO)*, pp. 451–462, 2012.

Iwata, S., Tanigawa, S., and Yoshida, Y. Improved approximation algorithms for $k$-submodular function maximization. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 404–413, 2016.

Kempe, D., Kleinberg, J. M., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 137–146, 2003.

Krause, A. and Golovin, D. Submodular function maximization., 2014.

Krause, A. and Guestrin, C. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI)*, pp. 324–331, 2005.

Krause, A. and Guestrin, C. Near-optimal observation selection using submodular functions. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1650–1654, 2007.

Lin, H. and Bilmes, J. A. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, pp. 510–520, 2011.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 94–103, 2007.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier than lazy greedy. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1812–1818, 2015.

Mirzasoleiman, B., Zadimoghaddam, M., and Karbasi, A. Fast distributed submodular cover: Public-private data summarization. In *Advances in Neural Information Processing Systems*, pp. 3594–3602, 2016.

Mitrovic, M., Bun, M., Krause, A., and Karbasi, A. Differentially private submodular maximization: Data summarization in disguise. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 2478–2487, 2017.

Nemhauser, G. L. and Wolsey, L. A. Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.*, 3(3):177–188, 1978.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functionsi. *Math. Program.*, 14(1):265–294, 1978.

Ohsaka, N. and Yoshida, Y. Monotone $k$-submodular function maximization with size constraints. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 694–702, 2015.

Papadimitriou, C. H., Schapira, M., and Singer, Y. On the hardness of being truthful. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 250–259, 2008.

Sakaue, S. On maximizing a monotone $k$-submodular function subject to a matroid constraint. *Discrete Optimization*, 23:105–113, 2017.

Singh, A. P., Guillory, A., and Bilmes, J. A. On bisubmodular maximization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1055–1063, 2012.

Streeter, M. J. and Golovin, D. An online algorithm for maximizing submodular functions. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1577–1584, 2008.

Vondrák, J. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 67–74, 2008.

Ward, J. and Zivny, S. Maximizing bisubmodular and $k$-submodular functions. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1468–1481, 2014.

Yoshida, Y. Cheeger inequalities for submodular transformations. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2582–2601, 2019.