Fast and Private Submodular and $k$-Submodular Functions Maximization with Matroid Constraints (full version)

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 $(1 - \frac{1}{e})$-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 $F : 2^E \rightarrow \mathbb{R}$ 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 & Guestrin, 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 $D = \{(x_i, C_i)\}_{i=1}^n$ consists of a feature vector $x_i = (x_{i1}, \ldots, x_{in})$ 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 $S \subseteq [n]$ 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
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 \( F_D : 2^E \rightarrow \mathbb{R}_+ \) 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 \( p_{u,v}^k \) 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 \( F_D : (k + 1)^E \rightarrow \mathbb{R}_+ \) 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 \((\epsilon r)\)-differentially private algorithm that, with high probability, returns \( S \in \mathcal{I} \) with quality at least \( (1 - \frac{1}{\epsilon})OPT - O\left(\sqrt{\epsilon} + \frac{\Delta r |\mathcal{M}| |E| \ln |E|}{\epsilon^2}\right) \).

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 \( |E|^{1/2} \). 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 \((\epsilon r)\)-differentially private algorithm that, with high probability, returns \( S \in \mathcal{I} \) with quality at least \( (1 - \frac{1}{\epsilon})OPT - O\left(\sqrt{\epsilon} + \frac{\Delta r |\mathcal{M}| |E| \ln |E|}{\epsilon^2}\right) \). Moreover, this algorithm makes at most \( O(r|\mathcal{M}| |E|^2 \ln |E| / \epsilon) \) 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 \((\epsilon r)\)-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 $F_D : (k + 1)^E \rightarrow \mathbb{R}_+$ is monotone and has sensitivity $\Delta$. For any $\epsilon > 0$, there is an $O(\epsilon r(M))$-differentially private algorithm that, with high probability, returns a solution $X = (X_1, \ldots, X_n) \in (k + 1)^E$ with $\bigcup_{i \in [k]} X_i \in I$ and $F_D(X) \geq \frac{1}{2} \text{OPT} - O\left(\frac{\Delta r(M) \ln |E|}{\epsilon} \right)$ by evaluating $F_D$ at most $O(\epsilon r(M)|E|)$ 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 $O\left(\frac{k|E|\ln r(M)\ln \frac{r(M)}{\epsilon}}{\epsilon} \right)$. 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 $F_{x_i} : 2^E \rightarrow [0, 1]$. 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 $\frac{1}{n}$.

**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 $(1 - \frac{1}{4})\text{OPT} - O\left(\frac{k\ln(\epsilon/\delta)\ln|E|}{\epsilon} \right)$.

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 $(1 - \frac{1}{4}) \text{OPT} - O\left(\frac{2\Delta k\ln|E|}{\epsilon} \right)$.

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 $M = (E, I)$ be a matroid. Then for any $\epsilon > 0$, there is an $(\epsilon r(M))$-differentially private algorithm that, with high probability, returns a solution $S \in I$ with quality at least $\frac{1}{2} \text{OPT} - O\left(\frac{\Delta r(M) \ln |E|}{\epsilon} \right)$.

$k$-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., $|X_i| \leq N$ for $i = 1, \ldots, k$. Sakaue (2017) proved that $1/2$-approximation can be achieved for matroid constraint, i.e., $\bigcup_{i \in [k]} X_i \subseteq I$.

2. Preliminaries

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

2.1. Submodular Functions

Let $F : 2^E \rightarrow \mathbb{R}_+$ be a set function. We say that $F$ is monotonous 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 $f : [0, 1]^E \rightarrow \mathbb{R}$ of a set function $F : 2^E \rightarrow \mathbb{R}$ is $f(x) = \sum_{S \subseteq E} F(S) \prod_{e \in S} x(e) \prod_{e \in E \setminus S} (1 - x(e))$.

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

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

1. \( f \) is monotone, meaning \( \frac{\partial f}{\partial x_i(x)} \geq 0 \). Hence, \( \nabla f(x) = (\frac{\partial f}{\partial x_1(x)}, \ldots, \frac{\partial f}{\partial x_n(x)}) \) is a nonnegative vector.
2. \( f \) is concave along any direction \( d \geq 0 \).

2.2. \( k \)-submodular Functions

Given a natural number \( k \geq 1 \), a function \( F : (k+1)^E \to \mathbb{R}_+ \) 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 \( E \),

\[
F(S) + F(T) \geq F(S \cap T) + F(S \cup T),
\]

where we define

\[
S \cap T = (S_1 \cap T_1, \ldots, S_k \cap T_k),
\]

\[
S \cup T = \left( (S_1 \cup T_1), \ldots, \left( \bigcup_{i \neq 1} S_i \cup T_i \right) \right).
\]

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 \( r_M : 2^E \to \mathbb{Z}_+ \) of \( \mathcal{M} \) is

\[
r_M(S) = \max \{|I| : I \subseteq S, I \in \mathcal{I}\}.
\]

An independent set \( S \in \mathcal{I} \) is called a base if \( r_M(S) = r_M(E) \). We denote the set of all bases by \( \mathcal{B} \) and rank of \( \mathcal{M} \) by \( r(\mathcal{M}) \). The matroid polytope \( P(\mathcal{M}) \subseteq \mathbb{R}^E \) of \( \mathcal{M} \) is

\[
P(\mathcal{M}) = \text{conv}\{1_I : I \in \mathcal{I}\},
\]

where \( \text{conv} \) denotes the convex hull. Or equivalently (Edmonds, 2001),

\[
P(\mathcal{M}) = \{ x \geq 0 : x(S) \leq r_M(S) \forall S \subseteq E \}.
\]

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

**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 \( x \in K \), there exists \( y \in C \) such that \( \|x - y\| < \rho \).

**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 \( P(\mathcal{M}) \) 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 \( P(\mathcal{M}) \).

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 \( S \subseteq \text{range}(M) \),

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

When \( \delta = 0 \), we say \( M \) is \( \epsilon \)-differentially private.

In our case, a dataset \( D \) consists of private submodular functions \( F_1, \ldots, F_n : 2^E \to [0, 1] \). 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 \( F_D : X \to Y \), parameterized by a dataset \( D \), is defined as \( \max_{D' : D \sim D'} |F_D(x) - F_D(x')| \). 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 \( M_i(D) \). 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
- \( \epsilon = \frac{1}{k} k\epsilon_0^2 + \sqrt{2 \ln 1/k} \epsilon_0 \) and \( \delta = \delta' + k\delta \) for any \( \delta' > 0 \) (the advanced composition).
Algorithm 1 Differentially Private Continuous Greedy

1: \textbf{Input:} Submodular function $F_D : 2^E \rightarrow [0, 1]$, dataset $D$, matroid $\mathcal{M} = (E, \mathcal{I})$, and $\epsilon > 0$ and $\rho \geq 0$.
2: Let $C_\rho$ be a $\rho$-covering of $\mathcal{P}(\mathcal{M})$, and $f_D$ be the multilinear extension of $F_D$.
3: $x_0 \leftarrow 0$, $e' \leftarrow \frac{\epsilon}{2\rho}$.
4: $\alpha \leftarrow \frac{1}{\rho}$, where $T = r(\mathcal{M})$.
5: for $t = 1$ to $T$ do
6: Sample $y \in C_\rho$ with probability proportional to $\exp(e'(y, \nabla f_D(x_{t-1})))$.
7: Let $y_{t-1}$ be the sampled vector.
8: $x_t \leftarrow x_{t-1} + \alpha y_{t-1}$.
9: end for
10: \textbf{Output:} $x_T$

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 $q_D : \mathcal{R} \rightarrow \mathbb{R}$, 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 $q_D : \mathcal{R} \rightarrow \mathbb{R}$ be a quality score. Then, the exponential mechanism $EM(\epsilon, \Delta, q_D)$ outputs $R \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\epsilon}{\Delta} q_D(R)\right)$.

Theorem 2.8 (McSherry & Talwar (2007)). Suppose that the quality score $q_D : \mathcal{R} \rightarrow \mathbb{R}$ 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 $F_1, \ldots, F_n : 2^E \rightarrow [0, 1]$, $\epsilon, \delta > 0$, and a matroid $\mathcal{M} = (E, \mathcal{I})$. Let $x^* \in \mathcal{P}(\mathcal{M})$ 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 $x, y \in [0, 1]^E$ with $\|y\|_2 \leq \rho$ and $x + v \in [0, 1]^E$, we have $|f(x) - f(x + v)| \leq 4\sqrt{|E|}\sqrt{\rho}$.

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

Proof. First, we show $\langle y, \nabla f(x) \rangle \geq f(y) - f(x)$.

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

$$\langle y, \nabla f(x) \rangle \geq \langle d, \nabla f(x) \rangle \geq f(x + d) - f(x) \geq f(y) - f(x),$$

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

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

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

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 $x_T \in \mathcal{P}(\mathcal{M})$ such that

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

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 $x_T \in \mathcal{P}(\mathcal{M})$. In what follows, we focus on the quality of the output of the algorithm. Suppose $y' \in C_\rho$ with $\|y' - x^*\|_2 \leq \rho$. By Theorem 2.8, with probability at least $1 - \frac{1}{|E|}$, we have

$$\langle y_t, \nabla f(x_t) \rangle \geq \arg\max_{y \in C_\rho} \langle y, \nabla f(x_t) \rangle - \frac{2\Delta}{\epsilon} \ln(|E|^2 |C_\rho|) \geq \langle y', \nabla f(x_t) \rangle - \frac{2\Delta}{\epsilon} \ln(|E|^2 |C_\rho|)$$

By Lemma 3.2

$$\langle y_t, \nabla f(x_t) \rangle \geq f(x^*) - f(x_t) - C_{3.2} \sqrt{\rho} - \frac{2\Delta}{\epsilon} \ln(|E|^2 |C_\rho|)$$

By a union bound, with probability at least $1 - \frac{1}{\text{poly}(|E|)}$, 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 $\frac{df}{dt} = \alpha y_t$, hence

$$\frac{df(x_t)}{dt} = \sum_e \frac{\partial f(x_t(e))}{\partial x_t(e)} \frac{dx_t(e)}{dt}$$
\[
= \nabla f(x_t) \cdot \frac{dx_t}{dt} = \alpha \langle y_t, \nabla f(x_t) \rangle \\
\geq \alpha \left( f(x^*) - f(x_0) - C_{3.2} \sqrt{\frac{\beta}{\epsilon}} \ln(|E|^2|C_{\rho}|) \right),
\]
where the first equality follows from the chain rule. Let \( \beta = f(x^*) - C_{3.2} \sqrt{\frac{\beta}{\epsilon}} \ln(|E|^2|C_{\rho}|) \). Solving the following differential equation \( \frac{df(x_t)}{dt} = \alpha(\beta - f(x_t)) \) with \( f(x_0) = 0 \) gives us \( f(x_t) = \beta(1 - e^{-\alpha t}) \).

For the first factor, we have
\[
\frac{\exp(\epsilon' \cdot d_{y'})}{\exp(\epsilon' \cdot d_{y_0})} = \frac{\exp(\epsilon'(d_{y_1} - d_{y_0}))}{\exp(\epsilon'(d_{y_1} - d_{y_0}))} = \frac{\exp(\epsilon'(y_t, \nabla f_D(x_t) - \nabla f_{D'}(x_t)))}{\exp(\epsilon'(y_t, \nabla f_D(x_t) - \nabla f_{D'}(x_t)))} \\
\leq \frac{\exp(\epsilon' \|y_t\|_1 \|\nabla f_D(x_t) - \nabla f_{D'}(x_t)\|_\infty)}{\exp(\epsilon' \|y_t\|_1 \|\nabla f_D(x_t) - \nabla f_{D'}(x_t)\|_\infty)} \\
= \exp \left( \epsilon' \sum_{e \in E} y_t(e) \cdot \left( \max_{e \in E \setminus x_t} \left[ F_D(R \cup \{e\}) - F_D(R) \right] \right) \right) \\
\leq \exp(O(\epsilon' \cdot r(M) \cdot 2\Delta)) \leq \exp(O(\epsilon' \cdot r(M)))
\]

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

For the second factor, let us write \( \beta_y = d_{y'} - d_y \) to be the \textit{deficit} of the probabilities of choosing direction \( y \) in instances \( f_{D'} \) and \( f_D \). Then, we have
\[
\sum_{y \in C_{\rho}} \exp(\epsilon' \cdot d_y) = \sum_{y \in C_{\rho}} \exp(\epsilon' \cdot \beta_y) \\
\leq \mathbb{E}_y[\exp(\epsilon' \cdot \beta_y)] \leq \exp(O(\epsilon' \cdot r(M) \cdot 2\Delta)) \\
= \exp(O(\epsilon' \cdot r(M)))
\]

The expectation is taken over the probability distribution over \( y \) selected at time \( t \) in instance with input \( D \). Recall that we choose \( y \) with probability proportional to \( \exp(\epsilon' \cdot d_y) \). By a union bound, Algorithm 1 preserves \( O(r_M(\mathcal{M})) \leq O(\epsilon(\mathcal{M})^2) \)-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.

Note that the privacy factor in the work of Mitrovic et al. (2017) is \( O(\epsilon(\mathcal{M})) \). However, our privacy factor is \( O(\epsilon(\mathcal{M})^2) \), 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 \( \rho = \frac{\epsilon r_M(\mathcal{M})}{|E|^2} \). Then Algorithm 1 is \( (\epsilon(\mathcal{M})^2) \)-differentially private and, with high probability, returns \( S \in I \) with quality at least
\[
F_D(S) \geq \left( 1 - \frac{1}{e} \right) \OPT - O \left( \sqrt{\epsilon} + \frac{\Delta r(M)|E| \ln |E|}{\epsilon^3} \right)
\]

**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
4. Improving the Query Complexity

In this section, we improve the number of evaluations of $F$ from $O(r(M)|E|^1 + (\frac{|E|}{\epsilon})^3)$ to $O(r(M)|E|^2 \ln \frac{|E|}{\epsilon})$. In Algorithm 1, in order to choose a point with probability proportional to $\exp(\langle y, \nabla f(x) \rangle)$, it requires to compute $Z = \sum_{x \in C_\rho} \exp(\langle z, \nabla f(x) \rangle)$. This summation needs evaluating $\langle z, \nabla f(x) \rangle$ for all $z$ 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 $\langle z, \nabla f(x) \rangle$. 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 $x$ in $C_\rho$ and $\mu > 0$, let the $i$-th layer be $L^x_{\mu,i} = \{ z \in C_\rho \mid (1 + \mu)^{i-1} \leq \exp(\langle z, \nabla f(x) \rangle) < (1 + \mu)^i \}$, for $1 \leq i \leq k$, where

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

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

$$ \tilde{Z} = \sum_{i \in [k]} |L^x_{\mu,i}|(1 + \mu)^{i-1}, \quad Z_i = |L^x_{\mu,i}|(1 + \mu)^{i-1}. $$

Then, a layer $L^x_{\mu,i}$ is chosen with probability $\tilde{Z}_i$. Note that we do not want to spend time computing the exact value of $|L^x_{\mu,i}|$ for every layer, instead, we are interested in efficiently estimating these values. By Hoeffding’s inequality (Hoeffding, 1963), to estimate $|L^x_{\mu,i}|/|C_\rho|$ 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 $|L^x_{\mu,i}|/|C_\rho|$ with additive error of $\lambda$ for all $i = 1, \ldots, k$ 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

1. **Input:** Submodular function $F_D : 2^E \rightarrow [0, 1]$, dataset $D$, a matroid $M = (E, \mathcal{I})$, and $\epsilon$, $\mu$, $\rho$, $\lambda$, $\theta > 0$.
2. Let $C_\rho$ be a $\rho$-covering of $\mathcal{P}(\mathcal{M})$, and $f_D$ be the multilinear extension of $F_D$.
3. $x(0) \leftarrow 0$, $\epsilon' \leftarrow \frac{\epsilon}{\lambda^2}$.
4. for $t = 1$ to $T = r(M)$ do
5. $C_\rho' \leftarrow$ Sample $\Theta(\ln(k/\theta)/\lambda^2)$ points from $C_\rho$ uniformly at random.
6. Define $L^x_{\mu,i}$ as in Definition 4.1, and estimate each $|L^x_{\mu,i}|$ using $C_\rho'$.
7. Let $L^x_{\mu,i}$ denote the estimated value.
8. Set $Z_t \leftarrow \sum_{i \in [k]} L^x_{\mu,i} (1 + \mu)^{i-1}$ and $\tilde{Z} \leftarrow \sum_{i \in [k]} \tilde{Z}_i (1 + \mu)^{i-1}$.
9. Let $L$ be the chosen layer $L^x_{\mu,i}$ with probability proportional to $\frac{Z_t}{\tilde{Z}}$.
10. Let $y_{t-1}$ be a point sampled uniformly at random from $L$.
11. $x_t \leftarrow x_{t-1} + \omega y_{t-1}$.
12. end for
13. **Out: $x_T$**

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

**Lemma 4.3** (Analogous to Theorem 2.8). At each time step $t$, Algorithm 2 returns $y_{t-1}$ such that for every $\beta \in (0, 1)$ and $\xi = \ln \left( |C_\rho|/(1 + k\lambda_1\lambda_2)(1 + \mu)^{t-1} \right)$ we have

$$ \Pr \left[ |y_{t-1}, \nabla f(x_{t-1})| \geq \max_{\mathbf{z} \in C_\rho} |\mathbf{z}, \nabla f(x_{t-1})| - \frac{2\Delta}{\epsilon} \frac{\xi}{1 - \beta} \right] \geq 1 - \theta_{1,2}. $$

**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 $\theta_{1,2}$), returns $x_T \in \mathcal{P}(\mathcal{M})$ such that

$$ f(x_T) \geq \left( 1 - \frac{1}{\epsilon} \right) \text{OPT} - O \left( C_{3,2} \sqrt{\rho} + \ln(1 + \mu) + \frac{\Delta r(M)}{\epsilon \rho^2} (|E| + \ln(k \lambda_{1,2})) \right). $$

**Theorem 4.5.** Algorithm 2 preserves $\Theta(\epsilon r(M)^2)$-differential privacy.

**Theorem 4.6** (Formal version of Theorem 1.2). Suppose $F_D$ is $\Delta$-sensitive and Algorithm 2 is instantiated with $\rho = \frac{1}{|E|^{1/2}}$, $\mu = \epsilon^2$, $\lambda_{1,2} = 1/\sqrt{|E|}$, $\theta_{1,2} = 1/|E|^2$. Then
Algorithm 2 is \((er(M))^2\)-differentially private and, with high probability, returns \(S \in I\) with quality at least
\[
F_D(S) \geq \left(1 - \frac{1}{e}\right)OPT - O\left(\frac{\Delta r(M) |E| \ln(\frac{|E|}{\epsilon})}{\epsilon^3}\right).
\]
Moreover, it evaluates \(F_D\) at most \(O(r(M)|E|^2 \ln(\frac{|E|}{\epsilon})\) 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 \(s \in \{0, 1, \ldots, k\}^E\) by
\[
S_i = \{e \in E \mid s(e) = i\} \quad \text{for} \quad i \in [k]
\]
and define the support of \(s\) as \(\text{supp}(s) = \{e \in E \mid s(e) \neq 0\}\). Let \(\leq\) be a partial ordering on \((k+1)^E\) such that, for \((S_1, \ldots, S_k)\) and \(t = (T_1, \ldots, T_k)\) in \((k+1)^E\), \(s \preceq t\) if \(S_i \subseteq T_i\) for every \(i \in [k]\). We say that a function \(F: (k+1)^E \to \mathbb{R}_+\) is monotone if \(F(s) \leq F(t)\) holds for every \(s \preceq t\). Define the marginal gain of adding \(e \not\in \bigcup_{i \in [k]} S_i\) to the \(i\)-th set of \(s \in (k+1)^E\) to be
\[
\Delta_{e,i}F(s) = F(S_1, \ldots, S_{i-1}, S_i \cup \{e\}, S_{i+1}, \ldots, S_k) - F(S_1, \ldots, S_k).
\]
The monotonicity of \(F\) is equivalent to \(\Delta_{e,i}F(s) \geq 0\) for any \(s = (S_1, \ldots, S_k)\) and \(e \not\in \bigcup_{i \in [k]} S_i\) and \(i \in [k]\).

Our goal is maximizing a monotone \(k\)-submodular function under matroid constraints. That is, given a monotone \(k\)-submodular function \(F_D: (k+1)^E \to \mathbb{R}_+\) and a matroid \(\mathcal{M} = (E, I)\), we want to solve the following problem.
\[
\max_{x \in (k+1)^E} F_D(x) \quad \text{subject to} \quad \bigcup_{i \in [k]} X_i \subseteq I
\]
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 \(x\) we have \(|\text{supp}(x)| = r(M)\).

Lemma 5.2 (Sakaue (2017)). Suppose \(A \subseteq I\) and \(B \subseteq B\) (recall \(B\) denotes the set of bases) satisfy \(A \subseteq B\). Then, for any \(e \not\in A\) satisfying \(A \cup \{e\} \subseteq I\), there exists \(e' \in B \setminus A\) such that \(B \setminus \{e'\} \cup \{e\} \subseteq B\).

Having Lemma 5.1, our algorithm runs in \(r(M)\) iterations and at each iteration chooses an element \(e\) with probability proportional to \(\exp(\epsilon \Delta_{e,i}F_D(x))\) and adds \(e\) to \(\text{supp}(x)\). 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

1. Input: monotone \(k\)-submodular functions \(F_D: (k+1)^E \to [0, 1]\), a matroid \(\mathcal{M} = (E, I)\), and \(\epsilon > 0\).
2. \(x \leftarrow 0, e' \leftarrow \frac{r}{2\Delta}
3. for \(t = 1\) to \(r(M)\) do
4. \(\Lambda(x) = \{e \in E \mid \text{supp}(x) \cup \{e\} \subseteq I\}\)
5. \(\epsilon\)
6. \(x(\epsilon) \leftarrow i\)
7. end for
8. Output: \(x\)

Theorem 5.3. Suppose \(F_D\) has sensitivity \(\Delta\). Then Algorithm 3, with high probability, returns \(x \in (k+1)^E\) such that \(\text{supp}(x) \subseteq B\) and \(F_D(x) \geq \frac{1}{2}\epsilon\text{OPT} - O\left(\frac{r(M) \ln |E|}{\epsilon}\right)\). 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 \(O(\epsilon r(M))\)-differential privacy. It also provides \(\frac{1}{2}\epsilon\text{OPT} - O\left(\frac{r(M) \ln |E|}{\epsilon}\right)\)-differential privacy for every \(\epsilon > 0\).

Clearly, Algorithm 3 evaluates \(F_D\) at most \(O(k|E|r(M))\) 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 \(x \in (k+1)^E\) such that \(\text{supp}(x)\) is a base of \(M\) and \(F_D(x) \geq \frac{1}{2}\epsilon\text{OPT} - O\left(\frac{r(M) \ln |E|}{\epsilon}\right)\) by evaluating \(F_D\) at most \(O(k|E|r(M))\) times. Moreover, this algorithm preserves \(O(r(M)\epsilon)\)-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 \(O(k|E|r(M))\) to \(O(k|E| \ln r(M) \ln \frac{r(M)}{\gamma})\), where \(\gamma > 0\) is a failure probability. Hence, even when \(r(M)\) 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 \(\frac{1}{2}\epsilon\text{OPT} - O\left(\frac{\Delta r(M) \ln |E|}{\epsilon}\right)\) by evaluating \(F_D\) at most \(O\left(k|E| \ln r(M) \ln \frac{r(M)}{\gamma}\right)\) times.

Similar to Theorem 5.4 and using the composition Theorem 2.6, Algorithm 4 preserves \(O(\epsilon r(M))\)-differential
Algorithm 4 Improved differentially private k-submodular maximization with a matroid constraint

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

It also provides $O\left(\frac{1}{2} r(M)^2 + \sqrt{2\ln 1/\delta'}\epsilon, \delta'ight)$-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 $x \in (k+1)^E$ such that $\text{supp}(x) \in B$ and $F_D(x) \geq \frac{1}{2} \text{OPT} - O\left(\frac{\Delta r(M) \ln \frac{|E|}{\gamma}}{\epsilon}\right)$ by evaluating $F_D$ at most $O\left(k |E| \ln r(M) \ln \frac{r(M)}{\gamma}\right)$ times. Moreover, this algorithm preserves $O(\epsilon r(M))$-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 an 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 $\sum_i |S_i| \leq B$ 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 linear 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


Dwork, C. and Lei, J. Differential privacy and robust statis-
Fast and Private Submodular and $k$-Submodular Functions Maximization (full version)


A. Probability Distributions

Let $P$ be a probability distribution over a finite set $E$. For an element $e \in E$, we write $P(e)$ to denote the probability that $e$ is sampled from $P$.

Let $P$ and $Q$ be two distributions over the same set $E$. The total variation distance and the Hellinger distance between $P$ and $Q$ are

\[
d_{TV}(P, Q) = \frac{1}{2} \sum_{e \in E} |P(e) - Q(e)| \quad \text{and} \quad h(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{e \in E} \left( \sqrt{P(e)} - \sqrt{Q(e)} \right)^2},
\]

respectively. It is well known that $d_{TV}(P, Q) \leq \sqrt{2} h(P, Q)$ holds.

For two distributions $P$ and $Q$, we denote by $P \otimes Q$ their product distribution. The following is well known:

**Lemma A.1.** Let $P_1, \ldots, P_n$ and $Q_1, \ldots, Q_n$ be probability distributions over $E$. Then, we have

\[
h(P_1 \otimes P_2 \otimes \cdots \otimes P_n, Q_1 \otimes Q_2 \otimes \cdots \otimes Q_n)^2 \leq \sum_{i=1}^n h(P_i, Q_i)^2.
\]

Finally, we use the following result due to Hoeffding in order to bound the error of our sampling step in Section 4.

**Theorem A.2** (Hoeffding’s inequality (Hoeffding, 1963)). Let $X_1, \ldots, X_n$ be independent random variables bounded by the interval $[0, 1]: 0 \leq X_i \leq 1$. We define the empirical mean of these variables by $\bar{X} = \frac{1}{n}(X_1 + \cdots + X_n)$. Then

\[
\Pr[|\bar{X} - E[\bar{X}]| \geq t] \leq \exp(-2nt^2).
\]

B. Missing Proofs from Section 3

Proof of Lemma 3.1. We have

\[
|f(x) - f(x + v)| = \left| \sum_{S \subseteq E} F(S) \left( \prod_{e \in S} x(e) \prod_{e \notin S} (1 - x(e)) - \prod_{e \in S} (x(e) + v(e)) \prod_{e \notin S} (1 - x(e) - v(e)) \right) \right|
\]

\[
\leq \sum_{S \subseteq E} \left| \prod_{e \in S} x(e) \prod_{e \notin S} (1 - x(e)) - \prod_{e \in S} (x(e) + v(e)) \prod_{e \notin S} (1 - x(e) - v(e)) \right|.
\]  

(1)

Now, we define probability distributions $\{P_e\}_{e \in E}$ and $\{Q_e\}_{e \in E}$ over $\{0, 1\}$ so that $P_e(1) = x(e)$ and $Q_e(1) = x(e) + v(e)$, respectively, for every $e \in E$. Note that

\[
g(x(e)) = h(P_e, Q_e)^2 = \left( \sqrt{x(e)} - \sqrt{x(e) + v(e)} \right)^2 + \left( \sqrt{1 - x(e)} - \sqrt{1 - x(e) - v(e)} \right)^2
\]

is a convex function with domain $x(e) \in [0, 1 - v(e)]$. The maximum value for this function happens at $x(e) = 0$ and $x(e) = 1 - v(e)$. Further its minimum is at $x(e) = [1 - v(e)]/2$.

\[
h(P_e, Q_e)^2 \leq g(x(e)) \leq g(0) = g(1 - v(e)) = 2 - 2\sqrt{1 - v(e)}
\]

\[
\leq v(e)^2 + v(e)
\]

\[
\leq 2v(e) \quad \text{(for } v(e) \in [0, 1])
\]

Letting $P = \bigotimes_{e \in E} P_e$ and $Q = \bigotimes_{e \in E} Q_e$, we have

\[
(1) \leq 2 \cdot d_{TV}(P, Q) = 2\sqrt{2} \cdot h(P, Q)
\]
\[ \leq 2\sqrt{2} \sqrt{\sum_{e \in E} h(P_e, Q_e)^2} \]  
(By Lemma A.1)

\[ = 2\sqrt{2} \sqrt{\sum_{e \in E} 2v(e)} \]

\[ = 4\sqrt{|v|_1} \]

\[ \leq 4\sqrt{|E||v|_2} \]

\[ \leq 4\sqrt{|E|\sqrt{\rho}} \]

C. Missing Proofs from Section 4

C.1. Proof of Lemma 4.3

Proof of Lemma 4.3. Let \( OPT = \max_{x \in C_0} \langle z, \nabla f(x_{t-1}) \rangle \) and \( q_t(z) = \langle z, \nabla f(x_{t-1}) \rangle \) for every \( z \in \mathcal{P}(M) \). Further, let \( y_t \) be the output of the algorithm and \( \bar{x}_{t-1}^{\mu,j} \) denote the estimated size of the \( i \)-th layer.

\[ \Pr \left[ q(y_t) \leq OPT - \frac{2\Delta}{\epsilon} \xi \right] \leq \frac{Pr[q(y_t) \geq OPT - \frac{2\Delta}{\epsilon} \xi]}{Pr[q(y_t) = OPT]} \]

\[ \leq \exp \left[ \epsilon' (OPT - \frac{2\Delta}{\epsilon} \xi + \ln(1 + \mu)) \right] \frac{\sum_{j=1}^{k} |L_{\mu,j}^{x_t-1} |(1 + \mu)^{\epsilon'(j-1)} \exp(\epsilon'OPT)}{\sum_{j=1}^{k} \bar{x}_{\mu,j}^{x_t-1}(1 + \mu)^{\epsilon'(j-1)}} \]

Consider the first term,

\[ \frac{\exp \left[ \epsilon' (OPT - \frac{2\Delta}{\epsilon} \xi + \ln(1 + \mu)) \right]}{\exp(\epsilon'OPT)} = \exp \left[ \epsilon' \left( -\frac{2\Delta}{\epsilon} \xi + \ln(1 + \mu) \right) \right] \]

\[ = \exp(-\xi) \exp (\epsilon' \ln(1 + \mu)) \]

\[ = \exp(-\xi)(1 + \mu)^{\epsilon'} \]

Consider the second term. By Corollary 4.2, the algorithm estimates \( |L_{\mu,j}^{x_t-1}|/|C_{\rho}| \) within additive error \( \lambda_{4,2} \) with probability at least \( 1 - \theta_{4,2} = 1 - \beta \). Therefore,

\[ \sum_{j=1}^{k} |L_{\mu,j}^{x_t-1} |(1 + \mu)^{\epsilon'(j-1)} \leq \sum_{j=1}^{k} (L_{\mu,j}^{x_t-1} + \lambda_{4,2}|C_{\rho}|)(1 + \mu)^{\epsilon'(j-1)} \]

\[ \leq 1 + \sum_{j=1}^{k} \lambda_{4,2}|C_{\rho}| \]

\[ \leq 1 + k\lambda_{4,2}|C_{\rho}| \]
Therefore, putting both upper bounds together yields

\[
\Pr \left[ q(y_t) \leq \text{OPT} - \frac{2\Delta}{\epsilon} \right] \leq \exp(-\xi)(1 + \mu)^{t'} (1 + k\lambda_4)(C_\rho)
\]

As there are at most \(|C_\rho|\) outputs with quality \(\text{OPT} - \frac{2\Delta}{\epsilon} \xi\) their cumulative probability is at most

\[
|C_\rho|(1 + k\lambda_4)(C_\rho)(1 + \mu)^{t'} \exp(-\xi) = \frac{|C_\rho|(1 + k\lambda_4)(C_\rho)(1 + \mu)^{t'} \beta}{|C_\rho|(1 + k\lambda_4)(C_\rho)(1 + \mu)^{t'}} = \beta.
\]

\[\square\]

### C.2. Proof of Theorem 4.4

**Proof of Theorem 4.4.** Suppose \(y' \in C_\rho\) with \(\|y' - x^*\|_2 \leq \rho\). Let \(\beta = \frac{1}{|E|^2}\). By Lemma 4.3, with probability at least \(1 - \frac{1}{\text{poly}(|E|)}\), we have

\[
\langle y_t, \nabla f(x_t) \rangle = \operatorname{argmax}_{y \in C_\rho} \langle y, \nabla f(x_t) \rangle - \frac{2\Delta}{\epsilon} \xi
\]

\[
\geq \langle y', \nabla f(x_t) \rangle - \frac{2\Delta}{\epsilon} \xi
\]

\[
\geq f(x^*) - f(x_t) - C_{3.2} \sqrt{\rho} - \frac{2\Delta}{\epsilon} \xi \quad \text{(by Lemma 3.2)}
\]

By a union bound, with probability at least \(1 - \frac{1}{\text{poly}(|E|)}\), the above inequality holds for every \(t\). In what follows, we assume this has happened. As in the proof of Theorem 3.3, suppose \(t\) is a continuous variable and define \(\frac{dx_t}{dt} = \alpha y_t\).

\[
\frac{df(x_t)}{dt} = \sum_e \frac{\partial f(x_t(e))}{\partial x_t(e)} \frac{dx_t(e)}{dt} = \nabla f(x_t) \cdot \frac{dx_t}{dt} = \alpha \langle y_t, \nabla f(x_t) \rangle
\]

\[
\geq \alpha \left( f(x^*) - f(x_t) - C_{3.2} \sqrt{\rho} - \frac{2\Delta}{\epsilon} \xi \right)
\]

Solving the differential equation with \(f(x_0) = 0\) gives us

\[
f(x_t) = (1 - e^{-\alpha t}) \left( f(x^*) - C_{3.2} \sqrt{\rho} - \frac{2\Delta}{\epsilon} \xi \right).
\]

For \(\alpha = \frac{1}{T}\) and \(t = T\) we obtain

\[
f(x_T) = (1 - e^{-1}) \left( f(x^*) - C_{3.2} \sqrt{\rho} - \frac{2\Delta}{\epsilon} \xi \right)
\]

\[
= f(x^*)(1 - e^{-1}) - O \left( C_{3.2} \sqrt{\rho} + \frac{2\Delta}{\epsilon} \xi \right).
\]

Recall that \(\xi = \ln \left( \left| |C_\rho|(1 + k\lambda_4)(C_\rho)(1 + \mu)^{t'} \right| / \beta \right)\) and \(\beta = 1/|E|^2\). Next we give an upper bound for the error term.

\[
O \left( C_{3.2} \sqrt{\rho} + \frac{2\Delta}{\epsilon} \xi \right) = O \left( C_{3.2} \sqrt{\rho} + \frac{2\Delta}{\epsilon} \ln(|E|^2|C_\rho|) + \frac{2\Delta}{\epsilon} \ln(1 + \mu)^{t'} + \frac{2\Delta}{\epsilon} \ln(k\lambda_4|C_\rho|) \right)
\]

\[
= O \left( C_{3.2} \sqrt{\rho} + \ln(1 + \mu) + \frac{2\Delta}{\epsilon} \left[ \ln(|E|^2|C_\rho|) + \ln(k\lambda_4|C_\rho|) \right] \right)
\]

\[
= O \left( C_{3.2} \sqrt{\rho} + \ln(1 + \mu) + \frac{\Delta}{\rho} \left( B |E|^2 + \ln(k\lambda_4|C_\rho|) \right) \right)
\]

Note that by letting \(\mu = e^\epsilon - 1\) we get \(\ln(1 + \mu) = \epsilon\). Moreover, we get \(k \leq \frac{r(M)}{\epsilon^2}\). \(\square\)
C.3. Proof of Theorem 4.5

Proof of Theorem 4.5. Let $M$ denote Algorithm 2. Let $D$ and $D'$ be two neighboring datasets and $F_D$ and $F_{D'}$ be their associated functions. Suppose $C'_\rho(D, t)$ denotes the set of sampled points at time step $t$ given dataset $D$. Similarly, $C'_\rho(D', t)$ denotes set of sampled points at time step $t$ given dataset $D'$. Samples are drawn uniformly at random and independent from the input function. Hence, Line 2 of $M$ is 0-differentially private. Therefore, we assume $C'_\rho(D, t) = C'_\rho(D', t) = S_t$ for every time step $t$. Define $k, k'$ as follow:

\[
\begin{align*}
    k &= \log_{1+\mu} \left( \frac{\max_{y \in C'_\rho} \exp(\langle y, \nabla f_D(x) \rangle)}{\min_{y \in C'_\rho} \exp(\langle y, \nabla f_D(x) \rangle)} \right), \\
    k' &= \log_{1+\mu} \left( \frac{\max_{y \in C'_\rho} \exp(\langle y, \nabla f_{D'}(x) \rangle)}{\min_{y \in C'_\rho} \exp(\langle y, \nabla f_{D'}(x) \rangle)} \right).
\end{align*}
\]

Note that the layers might be different. Let us use $L_i(D)$ and $L_i(D')$ for the $i$-th layer given dataset $D$ and $D'$, respectively. Further, $\tilde{L}_i(D)$ and $\tilde{L}_i(D')$ denote the estimated size of the $i$-th layer.

For a fixed $y \in C'_\rho$, we consider the relative probability of $M$ choosing $y$ at time step $t$ given multilinear extensions of $F_D$ and $F_{D'}$. Let $M_t(f_D | x_t)$ denote the output of $M$ at time step $t$ given dataset $D$ and point $x_t$. Similarly, $M_t(f_{D'} | x_t)$ denote the output of $M$ at time step $t$ given dataset $D'$ and point $x_t$. Further, write $d_y = \langle y, \nabla f_D(x_t) \rangle$ and $d'_y = \langle y, \nabla f_{D'}(x_t) \rangle$.

Suppose $y \in L_i(D)$ given dataset $D$, and $y \in L_i(D')$ given dataset $D'$. Then, we have

\[
\begin{align*}
    \Pr[M_t(f_D | x_t) = y] &= \frac{\Pr[y \in S_t | D]}{\Pr[y \in S_t | D']} \times \frac{|L_i(D)|[(1+\mu)^{r'(i-1)}]}{|L_i(D')|(1+\mu)^{r'(i-1)}} \times \frac{\sum_{j=1}^{k'} \tilde{L}_j(D') \exp(e' \cdot (1 + \mu)^{j-1})}{\sum_{j=1}^{k} \tilde{L}_j(D) \exp(e' \cdot (1 + \mu)^{j-1})} \\
    &= (1 + \mu)^{r'(i-1)} \times \frac{\sum_{j=1}^{k'} \tilde{L}_j(D') \exp(e' \cdot (1 + \mu)^{j-1})}{\sum_{j=1}^{k} \tilde{L}_j(D) \exp(e' \cdot (1 + \mu)^{j-1})} \times \frac{\sum_{j=1}^{k'} \tilde{L}_j(D') \exp(e' \cdot (1 + \mu)^{j-1})}{\sum_{j=1}^{k} \tilde{L}_j(D) \exp(e' \cdot (1 + \mu)^{j-1})} \\
    &= \Pr[M_t(f_{D'} | x_t) = y].
\end{align*}
\]

The second equality holds since points are sampled uniformly at random from $C'_\rho$ in Line 2.

Lemma C.1. Let $D, D'$ be neighboring datasets and $F$ be $\Delta$-sensitive. Suppose $z \in C'_\rho$ is a point in $L_j(D)$. Then

\[
(1 + \mu)^{r'(j-1)} \exp(-\frac{\epsilon r(M)}{2}) \leq \exp(e' \langle z, \nabla f_{D'}(x_t) \rangle) \leq (1 + \mu)^{r'(j)} \exp(\frac{\epsilon r(M)}{2}).
\]

Proof. Since $z \in C'_\rho$ is a point in $L_j(D)$, then $(1 + \mu)^{j-1} \leq \exp(\langle z, \nabla f_D(x) \rangle) < (1 + \mu)^j$. Since $F_D$ is $\Delta$-sensitive hence $f_D$ is $\Delta r(M)$-sensitive (recall the proof of Theorem 3.4). Therefore,

\[
\begin{align*}
    \exp(\langle z, \nabla f_{D'}(x_t) \rangle) &\leq \exp(\langle z, \nabla f_D(x_t) \rangle + \Delta r(M)) \leq (1 + \mu)^{j} \exp(\Delta r(M)) \quad (3) \\
    (1 + \mu)^{-1} \exp(-\Delta r(M)) &\leq \exp(\langle z, \nabla f_D(x_t) \rangle - \Delta r(M)) \leq \exp(\langle z, \nabla f_{D'}(x_t) \rangle) \quad (4) \\
    \Rightarrow (3), (4) &\Rightarrow (1 + \mu)^{j-1} \exp(-\Delta r(M)) \leq \exp(\langle z, \nabla f_{D'}(x_t) \rangle) \leq (1 + \mu)^j \exp(\Delta r(M)) \quad (5) \\
    \Rightarrow (5) &\Rightarrow (1 + \mu)^{r'(j-1)} \exp(-\frac{\epsilon r(M)}{2}) \leq \exp(e' \langle z, \nabla f_{D'}(x_t) \rangle) \leq (1 + \mu)^{r'(j)} \exp(\frac{\epsilon r(M)}{2}) \quad (6)
\end{align*}
\]

\(\square\)
The interpretation of (5) is that if a point \( z \in S_t \) appears in layer \( L_t(D) \) then it can be in any of the layers \( L_{j'}(D') \) for \((j - 1) + \log_{1+\mu}[(\exp(-\Delta r(M))] \leq p < j + \log_{1+\mu}(\exp(\Delta r(M)))]^1\). In a sense, the same argument in Claim C.2 shows that \( \lfloor \frac{k'}{k} \rfloor = 1 \). Now, we are ready to provide an upper bound for (2).

Consider the first term \( \frac{(1+\mu)^{r'(i-1)}}{(1+\mu)^{r'(j-1)}} \). Recall that \( y \in L_t(D) \) given dataset \( D \), and \( y \in L_{j'}(D') \) given dataset \( D' \). By Lemma C.1, we have \((1+\mu)^{r'(i-1)} \exp(-\frac{\epsilon r(M)}{2}) \leq \exp(\epsilon'(z, \nabla f_D(x_i)))\). Therefore,

\[
\frac{(1+\mu)^{r'(i-1)}}{(1+\mu)^{r'(j-1)}} \leq \frac{(1+\mu)^{r'(i-1)}}{(1+\mu)^{r'(j-1)} \exp(-\frac{\epsilon r(M)}{2})}
= \exp(\frac{\epsilon r(M)}{2})
\]

Now, we provide an upper bound for the second term of (2):

\[
\sum_{j=1}^{k'} \tilde{L}_j(D') \exp(\epsilon' \cdot (1+\mu)^{j-1}) \leq \sum_{j=1}^{k} \tilde{L}_j(D) \exp(\epsilon r(M)) \exp(\epsilon' \cdot (1+\mu)^{j-1})
\leq \sum_{j=1}^{k} \tilde{L}_j(D) \exp(\epsilon' \cdot (1+\mu)^{j-1})
= \frac{[\exp(\epsilon r(M))] \sum_{j=1}^{k} \tilde{L}_j(D) \exp(\epsilon' \cdot (1+\mu)^{j-1})}{\sum_{j=1}^{k} \tilde{L}_j(D) \exp(\epsilon' \cdot (1+\mu)^{j-1})}
\leq \exp(\epsilon r(M))
\]

By a union bound and composition Theorem 2.6, Algorithm 2 preserves \( O(\epsilon Tr(M)) \leq O(\epsilon r(M)^2) \)-differential privacy. The heart of the above inequality is that, given the set of sample points, the layers defined for both instances are almost identical. \( \square \)

**Claim C.2.** \( \frac{k'}{k} \leq 1 + \frac{2\Delta r(M)}{k \ln(1+\mu)} \).

**Proof.**

\[
k' = \log_{1+\mu} \left( \max_{y \in C_x} \frac{\exp(\langle y, \nabla f_D(x) \rangle)}{\min_{y \in C_x} \exp(\langle y, \nabla f_D(x) \rangle)} \right) \\
\leq \log_{1+\mu} \left( \max_{y \in C_x} \frac{\exp(\langle y, \nabla f_D(x) \rangle + \Delta r(M))}{\min_{y \in C_x} \exp(\langle y, \nabla f_D(x) \rangle - \Delta r(M))} \right) \\
\leq \log_{1+\mu} \left( \max_{y \in C_x} \frac{\exp(\langle y, \nabla f_D(x) \rangle)}{\min_{y \in C_x} \exp(\langle y, \nabla f_D(x) \rangle)} \right) \\
+ \log_{1+\mu} \exp(2\Delta r(M)) \\
= k + \frac{2\Delta r(M)}{\ln(1+\mu)} \\
= k \left( 1 + \frac{2\Delta r(M)}{k \ln(1+\mu)} \right).
\]

\( ^1 \)Note that in low-sensitivity regime, where \( \Delta \ll r(M) \), we have \( j - 1 \leq p < j \).
D. Missing Proofs from Section 5

D.1. Proof of Theorem 5.3

Proof of Theorem 5.3. Consider the $j$-th iteration of the algorithm. Let $(e^{(j)}, i^{(j)})$ be the pair chosen in this iteration. Further, let $o$ be the optimal solution and $x^{(j)}$ be the solution after the $j$-th iteration. Note that $|\text{supp}(x^{(j)})| = j$ for $j \in [r(M)]$. We define a sequence of vectors $o^{(0)} = o, o^{(1)}, \ldots, o^{r(M)}$, as in (Iwata et al., 2016; Ohsaka & Yoshida, 2015; Sakaue, 2017; Ward & Zivny, 2014), such that

1. $x^{(j)} \prec o^{(j)}$ for all $0 \leq j \leq r(M) - 1$,
2. $x^{(r(M))} = o^{(r(M))}$,
3. $O^{(j)} := \text{supp}(o^{(j)}) \in B$ for all $0 \leq j \leq r(M)$.

For the sake of completeness, let us describe how to obtain $o^{(j)}$ from $o^{(j-1)}$ assuming $x^{(j-1)} \prec o^{(j-1)}$ and $O^{(j-1)} \in B$. Let $X^{(j)} = \text{supp}(x^{(j)})$, $x^{(j-1)} \prec o^{(j-1)}$ implies that $X^{(j-1)} \subseteq O^{(j-1)}$ and $e^{(j)}$ is chosen to satisfy $X^{(j-1)} \cup \{e^{(j)}\} \in I$. By Lemma 5.2, there exists $e' \in O^{(j-1)} \setminus X^{(j-1)}$ such that $O^{(j-1)} \setminus \{e'\} \cup \{e^{(j)}\} \in B$.

Now let $o^{(j)} = e'$ and define $o^{(j-1/2)}$ as the vector obtained by assigning 0 to the $o^{(j)}$-th element of $o^{(j-1)}$. We then define $o^{(j)}$ as the vector obtained from $o^{(j-1/2)}$ by assigning $i^{(j)}$ to the $e^{(j)}$-th element. Therefore, for vector $o^{(j)}$ we have $O^{(j)} \in B$ and $x^{(j)} \prec o^{(j)}$.

By Theorem 2 in (Sakaue, 2017), if we always selected $(e^{(j)}, i^{(j)})$ with $e^{(j)} \in \Lambda(x), i \in [k]$ and maximum $\Delta_{e,i} f(x)$, we would have

$$ F(x^{(j)}) - F(x^{(j-1)}) \geq F(o^{(j-1)}) - F(o^{(j)}) - \frac{2\Delta}{\epsilon} \ln \frac{\Lambda(x^{(j)})}{\beta} $$

with probability at least $1 - \beta$. Given this, one can derive the following:

$$ F(o) - F(x^{(r(M))}) = \sum_{j=1}^{r(M)} F(o^{(j-1)}) - F(o^{(j)}) $$

$$ \leq \sum_{j=1}^{r(M)} \left( F(x^{(j-1)}) - F(x^{(j)}) + \frac{2\Delta}{\epsilon} \ln \frac{\Lambda(x^{(j)})}{\beta} \right) $$

$$ = F(x^{(r(M))}) - F(0) + r(M) \left( \frac{2\Delta}{\epsilon} \ln \frac{\Lambda(x^{(j)})}{\beta} \right) $$

$$ = F(x^{(r(M))}) + r(M) \left( \frac{2\Delta}{\epsilon} \ln \frac{\Lambda(x^{(j)})}{\beta} \right), $$

which means Algorithm 3 returns $x = x^{(r(M))}$ with quality at least $\frac{1}{2} \text{OPT} - r(M) \left( \frac{2\Delta}{\epsilon} \ln \frac{\Lambda(x^{(j)})}{\beta} \right)$ with probability at least $1 - r(M) \beta$. Having $\beta = \frac{1}{|E|}$, $|\Lambda(x^{(j)})| \leq |E|$ gives us

$$ F(x) \geq \frac{1}{2} \text{OPT} - O \left( \frac{\Delta r(M) \ln |E|}{\epsilon} \right). $$

D.2. Proof of Theorem 5.6

Proof of Theorem 5.6. Let $R^{(j)}$ be $R$ in the $j$-th iteration, $o$ be the optimal solution and $x^{(j)}$ be the solution after the $j$-th iteration. Further, let $X^{(j)} = \text{supp}(x^{(j)}), O^{(j)} = \text{supp}(o^{(j)})$, and

$$ A(x^{(j)}) = \{e \in E \setminus \text{supp}(x^{(j)}) \mid \text{supp}(x^{(j)}) \cup \{e\} \in I\} $$
We iteratively define $o^{(0)} = o, o^{(1)}, \ldots, o^{(r(M))]$ as follows. If $R(j) \cap \Lambda(x)^{(j)} = \emptyset$, then we regard that the algorithm failed. Else we proceed as follows. By Lemma 5.2, for any $e^{(j)} \in R(j) \cap \Lambda(x)^{(j)}$, there exists $e'$ such that $e' \in O^{(j)} \setminus X^{(j)}$ and $O^{(j-1)} \setminus \{e'\} \cup \{e^{(j)}\} \in B$. Now let $o^{(j)} = e'$ and define $o^{(j-1/2)}$ as the vector obtained by assigning 0 to the $o^{(j)}$-th element of $o^{(j-1)}$. We then define $o^{(j)}$ as the vector obtained from $o^{(j-1/2)}$ by assigning $i^{(j)}$ to the $e^{(j)}$-th element. Therefore, for vector $o^{(j)}$ we have $O^{(j)} \in B$ and $x^{(j)} \prec o^{(j)}$.

If the algorithm does not fail and $o^{(0)} = o, o^{(1)}, \ldots, o^{(r(M)]}$ are well defined, or in other words, if $R(j) \cap \Lambda(x)^{(j)}$ is not empty for every $j \in [r(M)]$, then the rest of the analysis is completely the same as in Theorem 5.3, and we achieve an approximation ratio of (roughly) $1/2$. Hence, it suffices to show that $R(j) \cap \Lambda(x)^{(j)}$ is not empty with a high probability.

**Lemma D.1.** With probability at least $1 - \frac{2r(M)}{r(M)^2}$, we have $R(j) \cap \Lambda(x)^{(j)} \neq \emptyset$ for every $j \in [r(M)]$.

Analogous to the analysis in Theorem 5.3, for every time step $0 \leq j \leq r(M)$, with probability at least $1 - \frac{2r(M)}{r(M)^2}$ we have

$$F(x^{(j)}) - F(x^{(j-1)}) \geq F(o^{(j-1)}) - F(o^{(j)}) - \frac{2\Delta}{\epsilon} \ln \left(\frac{r(M) |\Lambda(x)^{(j)}|}{\gamma}\right).$$

By a union bound over $j \in [r(M)]$, with probability at least $1 - \gamma$, it follows that

$$F(x) \geq \frac{1}{2} \text{OPT} - O \left( \frac{\Delta r(M) \ln(|E|/\gamma)}{\epsilon} \right).$$

Applying a similar argument as in (Ohsaka & Yoshida, 2015), the number of evaluations of $f$ is at most

$$k \sum_{i=1}^{r(M)} \frac{|E| - t + 1}{r(M) - t + 1} \ln \frac{r(M)}{\gamma} = k \sum_{i=1}^{r(M)} \frac{|E| - r(M) + t}{t} \ln \frac{r(M)}{\gamma} = O \left( k|E| \ln r(M) \ln \frac{r(M)}{\gamma} \right) \quad \Box$$

**Proof of Lemma D.1.**

$$\Pr[R(j) \cap \Lambda(x)^{(j)} = \emptyset] = \left(1 - \frac{r(M) - \text{supp}(x^{(j)})}{|E - \text{supp}(x^{(j)})|}\right)^{|R(j)|} \leq \exp \left( - \frac{r(M) - j + 1}{|E| - j + 1} \ln \frac{r(M)}{\gamma} \right) = \exp \left( - \ln \frac{r(M)}{\gamma} \right) = \frac{\gamma}{r(M)} \quad \Box$$